

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Гаджибутаева Султанага Рамазановна
Должность: Директор
Дата подписания: 09.06.2024 12:40:33
Уникальный программный ключ:
2b71376f78d52b66ab183b5be5a3b5fe443c04a8

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РЕСПУБЛИКИ ДАГЕСТАН

Частное профессиональное образовательное учреждение
«РЕСПУБЛИКАНСКИЙ ПОЛИПРОФЕССИОНАЛЬНЫЙ КОЛЛЕДЖ»
(ЧПОУ «Республиканский полипрофессиональный колледж»)



УТВЕРЖДАЮ

Зам. директора по учебно-методической работе

Кадыршева Ж.А.

« 03 » июля 2023 г.

Комплект контрольно-оценочных средств
по профессиональному модулю

**ПМ.01 РАЗРАБОТКА МОДУЛЕЙ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ
ДЛЯ КОМПЬЮТЕРНЫХ СИСТЕМ**

программы подготовки специалистов среднего звена по
специальности: 09.02.07 Информационные системы и
программирование

Год набора: 2023

Кизляр
2023г.

ОДОБРЕН
на заседании цикловой методической
комиссии общепрофессиональных
дисциплин и профессиональных
модулей по специальности 09.02.07
Информационные системы и
программирование
Протокол № 10 от «28» июня 2023 г.

Председатель ЦМК
Кадрьшева Ж.А.



Составлен в соответствии с требованиями
федерального государственного
образовательного стандарта по
специальности 09.02.07 Информационные
системы и программирование и рабочей
программы по профессиональному
модулю ПМ.01 Разработка модулей
программного обеспечения для
компьютерных систем

Организация-разработчик: Частное профессиональное образовательное учреждение
«Республиканский полипрофессиональный колледж».

Разработчик(и):

Магомедова Мадина Нурмагомедовна, преподаватель

Ф.И.О., ученая степень, звание, должность

Потапов Игорь Алексеевич, преподаватель

Ф.И.О., ученая степень, звание, должность

Содержание

1. ПАСПОРТ КОМПЛЕКТА КОНТРОЛЬНО-ОЦЕНОЧНЫХ СРЕДСТВ ПО ПРОФЕССИОНАЛЬНОМУ МОДУЛЮ.....	4
1.1 Общие положения	4
1.2 Формы контроля и оценивания элементов профессионального модуля.....	4
1.3 Требования к результатам освоения программы профессионального модуля.....	5
1.4. Распределение основных показателей оценки результатов по видам аттестации	10
2. РЕЗУЛЬТАТЫ ОСВОЕНИЯ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ.....	18
2.1. Перечень общих компетенций	18
2.2 Перечень профессиональных компетенций	18
3. ОЦЕНКИ ОСВОЕНИЯ ТЕОРЕТИЧЕСКОГО КУРСА ПРОФЕССИОНАЛЬНОГО МОДУЛЯ.....	19
3.1. Задания для оценки освоения МДК 01.01 Разработка программных модулей	19
3.2. Задания для оценки освоения МДК 01.02 Поддержка и тестирование программных модулей	88
3.3. Задания для оценки освоения МДК 01.03 Разработка мобильных приложений.....	102
3.4 Задания для оценки освоения МДК 01.04 Системное программирование	123
3.5 Задания для оценки освоения МДК 01.05 Разработка веб-приложений	130
3.6. Рекомендуемая литература для разработки оценочных средств и подготовки обучающихся к аттестации.	175
3.7 перечень материалов, оборудования и информационных источников	176
5. ПЕРЕЧЕНЬ ВИДОВ РАБОТ ДЛЯ ПРОВЕРКИ РЕЗУЛЬТАТОВ ОСВОЕНИЯ ПРОГРАММЫ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ НА ПРАКТИКЕ	180
6. КОНТРОЛЬНО-ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ЭКЗАМЕНА (КВАЛИФИКАЦИОННОГО).....	181

1. ПАСПОРТ КОМПЛЕКТА КОНТРОЛЬНО-ОЦЕНОЧНЫХ СРЕДСТВ ПО ПРОФЕССИОНАЛЬНОМУ МОДУЛЮ

ПМ.01 РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ КОМПЬЮТЕРНЫХ СИСТЕМ

1.1 Общие положения

Контрольно-оценочные средства (КОС) разработаны в соответствии с требованиями образовательной программы и Федерального государственного образовательного стандарта СПО по специальности 09.02.07 Информационные системы и программирование, программы профессионального модуля ПМ.01 Разработка программных модулей программного обеспечения для компьютерных систем.

Контрольно-оценочные средства предназначены для контроля и оценки образовательных достижений обучающихся, освоивших программу профессионального модуля ПМ.01 Разработка программных модулей программного обеспечения для компьютерных систем для специальности СПО 09.02.07 Информационные системы и программирование в части освоения вида профессиональной деятельности (ВПД): Разработка программных модулей программного обеспечения для компьютерных систем.

Формой аттестации по профессиональному модулю является экзамен (квалификационный).

Итогом экзамена является однозначное решение: «вид профессиональной деятельности «освоен/не освоен»».

1.2 Формы контроля и оценивания элементов профессионального модуля

Элемент модуля	Форма контроля и оценивания	
	Формы промежуточной аттестации	Текущий контроль
МДК 01.01 Разработка программных модулей	Контрольная работа Экзамен 6 семестр	Экспертная оценка в процессе защиты лабораторных работ Оценка результатов внеаудиторной самостоятельной работы. Тестирование
МДК 01.02 Поддержка и тестирование программных модулей	Дифференцированный зачет	Наблюдение и оценка выполнения практических работ Тестирование Контроль выполнения самостоятельной работы
МДК 01.03 Разработка мобильных приложений	Контрольная работа Экзамен	Наблюдение и оценка выполнения практических работ Тестирование Контроль выполнения самостоятельной работы
МДК 01.04 Системное программирование	Контрольная работа Экзамен	Наблюдение и оценка выполнения практических работ Тестирование Контроль выполнения самостоятельной работы
МДК 01.05	Дифференцированный зачет	Наблюдение и оценка

Разработка веб-приложений	Экзамен	выполнения практических работ Тестирование Контроль выполнения самостоятельной работы
Учебная практика УП.01	Дифференцированный зачет 1 Дифференцированный зачет 2 Дифференцированный зачет 3	Оценка выполненных работ на учебной практике
Производственная практика (по профилю специальности) ПП.01	Дифференцированный зачет 1 Дифференцированный зачет 2	Оценка выполненных работ на производственной практике
ПМ.01 Разработка программных модулей программного обеспечения для компьютерных систем	Экзамен квалификационный ПМ.01.ЭК	Результаты МДК 01.01, МДК 01.02, МДК 01.03, МДК 01.04, МДК 01.05, УП.01, ПП.01

1.3 Требования к результатам освоения программы профессионального модуля

Код и наименование ПК (ОК)	Код и наименование основных показателей оценки результатов (ОПОР)	Код и наименование практического опыта	Код и наименование умений	Код и наименование знаний
ОК 01 Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам	- обоснование сущности и социальной значимости своей будущей профессии; - добросовестное выполнение учебных обязанностей при освоении профессиональной деятельности.	П1 в разработке кода программного продукта на основе готовой спецификации на уровне модуля;	У1 осуществляют разработку кода программного модуля на языках низкого и высокого уровней;	З1 осуществлять разработку кода программного модуля на языках низкого и высокого уровней;
ОК 02 Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения	- обоснованный выбор и применение методов и способов решения профессиональных задач в области вычислительной техники и компьютерной	П1 в разработке кода программного продукта на основе готовой спецификации на уровне модуля;	У1 осуществляют разработку кода программного модуля на языках низкого и высокого уровней;	З1 осуществлять разработку кода программного модуля на языках низкого и высокого

задач профессиональной деятельности	оргтехники; - правильная последовательность выполнения действий на лабораторных работах, во время производственной практики в соответствии с инструкциями, указаниями и т.п.			уровней;
ОК 03 Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по финансовой грамотности в различных жизненных ситуациях	- безошибочность решения стандартных и нестандартных профессиональных задач.	П1 в разработке кода программного продукта на основе готовой спецификации на уровне модуля;	У2 создавать программу по разработанному алгоритму как отдельный модуль;	32 создавать программу по разработанному алгоритму как отдельный модуль;
ОК 04 Эффективно взаимодействовать и работать в коллективе и команде	- демонстрация приемов и способов работы с различными информационными источниками (учебной, справочной, технической литературой) для эффективного выполнения профессиональных задач.	П1 в разработке кода программного продукта на основе готовой спецификации на уровне модуля;	У2 создавать программу по разработанному алгоритму как отдельный модуль;	32 создавать программу по разработанному алгоритму как отдельный модуль;
ОК 05 Осуществлять устную и письменную коммуникацию на государственном	- использование методов и средств информационных и телекоммуникационных технологий; - владение методами анализа	П1 в разработке кода программного продукта на основе готовой спецификации на уровне	У6 оформлять документацию на программные средства	36 оформлять документацию на программные средства

языке Российской Федерации с учетом особенностей социального и культурного контекста	информационных ресурсов.	модуля;		
ОК 06 Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения	- взаимодействие с обучающимися и преподавателями в ходе обучения; - полнота понимания того, что успешность и результативность работы зависит от согласованности действий всех участников команды работающих.	П1 в разработке кода программного продукта на основе готовой спецификации на уровне модуля;	У6 оформлять документацию на программные средства	З6 оформлять документацию на программные средства
ОК 07 Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях	- самоанализ и коррекция результатов собственной работы.	П1 в разработке кода программного продукта на основе готовой спецификации на уровне модуля;	У4 осуществляют разработку кода программного модуля на современных языках программирования;	З4 осуществлять разработку кода программного модуля на современных языках программирования;
ОК 08 Использовать средства физической	- организация самостоятельных занятий при изучении	П1 в разработке кода программного продукта на		

культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности	профессионального модуля.	основе готовой спецификации на уровне модуля;		
ОК 09 Пользоваться профессиональной документацией на государственном и иностранном языках	- соблюдение техники безопасности	П1 в разработке кода программного продукта на основе готовой спецификации на уровне модуля;	У6 оформлять документацию на программные средства	З6 оформлять документацию на программные средства
ПК 1.1 Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием	– обслуживание модулей; оценке качества кода предложенного программного модуля, поиску некачественного программного кода, его анализу, оптимизации методами рефакторинга.	П1 в разработке кода программного продукта на основе готовой спецификации на уровне модуля;	У1 осуществлять разработку кода программного модуля на языках низкого и высокого уровней;	З1 осуществлять разработку кода программного модуля на языках низкого и высокого уровней;
ПК 1.2 Разрабатывать программные модули в соответствии с техническим заданием	практическое задание по разработке программного модуля в соответствии с техническим заданием	П1 в разработке кода программного продукта на основе готовой спецификации на уровне модуля;	У2 создавать программу по разработанному алгоритму как отдельный модуль;	З2 создавать программу по разработанному алгоритму как отдельный модуль;
ПК 1.3 Выполнять отладку программных модулей с использованием специализирован	практическое задание по выполнению отладки предложенного программного модуля	П2 использовании инструментальных средств на этапе отладки программного продукта;	У3 выполнять отладку и тестирование программы на уровне модуля;	З3 выполнять отладку и тестирование программы на уровне

ных программных средств				модуля;
ПК 1.4 Выполнять тестирование программных модулей	практическое задание по выполнению заданных видов тестирования программного модуля.	П3 проведении тестирования программного модуля по определенному сценарию;	У5 уметь выполнять оптимизацию и рефакторинг программного кода;	З5 уметь выполнять оптимизацию и рефакторинг программного кода;
ПК 1.5 Осуществлять рефакторинг и оптимизацию программного кода	оценке качества кода предложенного программного модуля, поиску некачественного программного кода, его анализу, оптимизации методами рефакторинга	П4 использовании инструментальных средств на этапе отладки программного продукта;	У6 оформлять документацию на программные средства	З6 оформлять документацию на программные средства
ПК 1.6 Разрабатывать модули программного обеспечения для мобильных платформ	практическое задание по разработке программного модуля в соответствии с техническим заданием	П5 разработке мобильных приложений	У4 осуществляют разработку кода программного модуля на современных языках программирования;	З4 осуществлять разработку кода программного модуля на современных языках программирования;
ДПК 1.7 Разрабатывать веб-приложение в соответствии с техническим заданием	практическое задание по разработке программного модуля в соответствии с техническим заданием	П1 в разработке кода программного продукта на основе готовой спецификации на уровне модуля;	У4 осуществляют разработку кода программного модуля на современных языках программирования;	З4 осуществлять разработку кода программного модуля на современных языках программирования;

1.4. Распределение основных показателей оценки результатов по видам аттестации

Код и наименование компетенции	Основные показатели оценки результатов	Текущий контроль	Формы промежуточной аттестации					УП.0 1.01	ПП.0 1.01
			МДК 01.01	МДК 01.02	МДК 01.03	МДК 01.04	МДК 01.05		
			КР-3, Э-4	ДЗ-4	КР-4, Э-5	КР-4, Э-5	ДЗ-5, Э-6		
ОК 01 Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам	обоснованность постановки цели, выбора и применения методов и способов решения профессиональных задач; - адекватная оценка и самооценка эффективности и качества выполнения профессиональных задач	Экспертная оценка в форме дифференцированного зачета Защита отчетов по практике						+	+
ОК 02 Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности	использование различных источников, включая электронные ресурсы, медиаресурсы, Интернетресурсы, периодические издания по специальности для решения профессиональных задач	Экспертная оценка в форме дифференцированного зачета Защита отчетов по практике						+	+
ОК 03 Планировать и реализовывать собственное профессиональное и личностное развитие,	демонстрация ответственности за принятые решения - обоснованность самоанализа и коррекция результатов собственной работы;	Экспертная оценка в форме дифференцированного зачета Защита отчетов по практике						+	+

предпринимательскую деятельность в профессиональной сфере, использовать знания по финансовой грамотности в различных жизненных ситуациях									
ОК 04 Эффективно взаимодействовать и работать в коллективе и команде	- взаимодействие с обучающимися, преподавателями и мастерами в ходе обучения, с руководителями учебной и производственной практик; - обоснованность анализа работы членов команды (подчиненных)	Экспертная оценка в форме дифференцированного зачета Защита отчетов по практике						+	+
ОК 05 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста	демонстрация грамотности устной и письменной речи, - ясность формулирования и изложения мыслей	Экспертная оценка в форме дифференцированного зачета Защита отчетов по практике						+	+
ОК 06 Проявлять гражданско-патриотическую позицию,	демонстрация грамотности устной и письменной речи, - ясность формулирования и изложения мыслей	Экспертная оценка в форме дифференцированного зачета						+	+

демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения		Защита отчетов по практике							
ОК 07 Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях	- эффективность выполнения правил ТБ во время учебных занятий, при прохождении учебной и производственной практик; - демонстрация знаний и использование ресурсосберегающих технологий в профессиональной деятельности	Экспертная оценка в форме дифференцированного зачета Защита отчетов по практике						+	+
ОК 08 Использовать	- эффективность использования средств физической культуры для	Экспертная оценка в форме						+	+

средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности	сохранения и укрепления здоровья при выполнении профессиональной деятельности.	дифференцированно о зачет Защита отчетов по практике							
ОК 09 Пользоваться профессиональной документацией на государственном и иностранном языках	- эффективность использования в профессиональной деятельности необходимой технической документации, в том числе на английском языке	Экспертная оценка в форме дифференцированно о зачет Защита отчетов по практике						+	+
ПК 1.1 Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием	Соответствие выполненных работ предъявляемым требованиям: - знает: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования; актуальную нормативно-правовую базу в области документирования алгоритмов. - применяет алгоритмы решения поставленной задачи и реализовывать его средствами	Экспертная оценка в форме экзамена: практическое задание по построению алгоритма в соответствии с техническим заданием Защита отчетов по практическим и лабораторным работам	+					+	+

	автоматизированного проектирования. - выполняет оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства - использует приобретенный опыт в разработке алгоритма решений поставленной задачи и реализовать его средствами автоматизированного проектирования.									
ПК 1.2 Разрабатывать программные модули в соответствии с техническим заданием	Соответствие выполненным работ предъявляемым требованиям: - знает: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования; знание API современных мобильных операционных систем. - применяет разработанный код программного продукта на основе готовой спецификации на уровне модуля. - выполняет оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства - использует разработку кода программного модуля на языках низкого уровня и высокого уровней в том числе для мобильных платформ.	Экспертная оценка в форме экзамена: практическое задание по построению алгоритма в соответствии с техническим заданием Защита отчетов по практическим и лабораторным работам	+		+		+		+	+
ПК 1.3 Выполнять	Соответствие выполненным работ	Экспертная оценка в		+		+		+		+

отладку программных модулей с использованием специализированных программных средств	предъявляемым требованиям: - знает: основные принципы отладки и тестирования программных продуктов; инструментарий отладки программных продуктов. - применяет инструментальные средства на этапе отладки программного продукта; - выполняет оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства - использует тестирование программного модуля по определенному сценарию.	форме экзамена: практическое задание по построению алгоритма в соответствии с техническим заданием Защита отчетов по практическим и лабораторным работам							
ПК 1.4 Выполнять тестирование программных модулей	Соответствие выполненных работ предъявляемым требованиям: - знает: основные виды и принципы тестирования программных продуктов. - применяет тестирование программного модуля по определенному сценарию. - выполняет оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства - использует: инструментальные средства на этапе тестирования программного продукта	Экспертная оценка в форме экзамена: практическое задание по построению алгоритма в соответствии с техническим заданием Защита отчетов по практическим и лабораторным работам		+				+	+
ПК 1.5 Осуществлять рефакторинг и оптимизацию программного кода	Соответствие выполненных работ предъявляемым требованиям: - знает: способы оптимизации и приемы рефакторинга; инструментальные средства анализа	Экспертная оценка в форме экзамена: практическое задание по построению алгоритма в		+				+	+

	алгоритма; методы организации рефакторинга и оптимизации кода; принципы работы с системой контроля версий. - применяет алгоритмы, в том числе с применением инструментальных средств. - выполняет оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства - использует работу с системой контроля версий.	соответствии с техническим заданием Защита отчетов по практическим и лабораторным работам							
ПК 1.6 Разрабатывать модули программного обеспечения для мобильных платформ	Соответствие выполненным работ предъявляемым требованиям: - знает: основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования. - применяет в разработке мобильных приложений; разработку кода программного модуля на современных языках программирования - выполняет оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства - использует разработанное мобильное приложение.	Экспертная оценка в форме экзамена: практическое задание по построению алгоритма в соответствии с техническим заданием Защита отчетов по практическим и лабораторным работам			+			+	+
ДПК 1.7 Разрабатывать веб-приложение в соответствии с техническим	Соответствие выполненным работ предъявляемым требованиям: - знает: основные этапы разработки программного обеспечения; основные принципы технологии	Экспертная оценка в форме экзамена: практическое задание по построению алгоритма в					+	+	+

заданием	структурного и объектно-ориентированного программирования; знание API современных мобильных операционных систем. - применяет разработанный код программного продукта на основе готовой спецификации на уровне модуля. - выполняет оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства - использует разработку кода программного модуля на языках низкого уровня и высокого уровней в том числе для мобильных платформ.	соответствии с техническим заданием Защита отчетов по практическим и лабораторным работам							
----------	--	---	--	--	--	--	--	--	--

2. РЕЗУЛЬТАТЫ ОСВОЕНИЯ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

Результатом освоения профессионального модуля является овладение обучающимися видом профессиональной деятельности разработка программных модулей программного обеспечения для компьютерных систем, в том числе профессиональными (ПК) и общими (ОК) компетенциями:

2.1. Перечень общих компетенций

Код	Наименование общих компетенций
ОК 01	Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам
ОК 02	Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности
ОК 03	Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по финансовой грамотности в различных жизненных ситуациях
ОК 04	Эффективно взаимодействовать и работать в коллективе и команде
ОК 05	Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста.
ОК 06	Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения
ОК 07	Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях
ОК 08	Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности
ОК 09	Пользоваться профессиональной документацией на государственном и иностранном языках

2.2 Перечень профессиональных компетенций

Код	Наименование видов деятельности и профессиональных компетенций
ПК 1.1	Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием
ПК 1.2	Разрабатывать программные модули в соответствии с техническим заданием
ПК 1.3	Выполнять отладку программных модулей с использованием специализированных программных средств
ПК 1.4	Выполнять тестирование программных модулей
ПК 1.5	Осуществлять рефакторинг и оптимизацию программного кода
ПК 1.6	Разрабатывать модули программного обеспечения для мобильных платформ
ДПК 1.7	Разрабатывать веб-приложение в соответствии с техническим заданием

3. ОЦЕНКИ ОСВОЕНИЯ ТЕОРЕТИЧЕСКОГО КУРСА ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

Основной целью оценки теоретического курса профессионального модуля является оценка умений и знаний.

Оценка теоретического курса профессионального модуля осуществляется с использованием следующих оценочных средств:

3.1. Задания для оценки освоения МДК 01.01 Разработка программных модулей

Практические и лабораторные работы Тема 1.2 Структурное программирование
Практическая работа №1. Оценка сложности алгоритмов сортировки

Цель практической работы: получение навыков определения сложности алгоритмов сортировки.

Задание:

Оценить эмпирически вычислительную сложность алгоритма простой сортировки на массиве, заполненном случайными числами (средний случай).

Эталон ответа:

Сортировка простого обмена (Exchange Sort) — это простой алгоритм сортировки, состоящий в повторяющихся проходах по сортируемому массиву. На каждой итерации последовательно сравниваются соседние элементы, и, если порядок в паре неверный, то элементы меняют местами.

Псевдокод:

```
N = length(A)
for i = 0 to N - 1 do
    for j = 0 to N - 2 do
        if (A[j] > A[j + 1])
            swap(A[j], A[j + 1])
        end
    end
end
```

end

Инвариант цикла: после выполнения i -й операции все элементы массива, находящиеся в «верхней» части массива будут отсортированы. Размерность массива конечна, следовательно конечен и цикл.

Я реализовал алгоритм на языке C++, результат представлен на рис. 1.



```
void ExchangeSort(int *arr, int N)
{
    for (int i = 0; i < N; i++)
    {
        for (int j = 0; j < N-1; j++)
        {
            if (arr[j] > arr[j + 1])
            {
                int b = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = b;
            }
        }
    }
}
```

Рисунок 1 – Реализация алгоритма «пузырьком» на C++

Анализ эффективности алгоритма

Лучший случай: массив изначально отсортирован по возрастанию. Функция роста: $T(n) = n - 1$.

Средний случай: массив заполнен случайными числами. Функция роста: $T(n) = 0,5n(n-1)$.

Худший случай: массив отсортирован по убыванию. Функция роста: $T(n) = n(n-1)$.

В таблице 1 показаны теоретические и практические результаты вычисления количества совершённых операций и времени работы. На рис. 2 показана графическая визуализация

зависимости количества операций от длины массива.

Таблица 2 – Сводная таблица результатов для алгоритма Exchange Sort

n	T(n), мс	$T_n = C_n + M_n$
100	0,009	7132
1000	0,437	746144
10000	49,820	749214123
100000	10394,5	74964165183
1000000	Невозможно измерить	Невозможно измерить

Ёмкостная сложность: $O(n)$, так как программе необходимо хранить лишь сам массив из N элементов.

Вывод об эмпирической вычислительной сложности алгоритма на основе скорости роста функции роста

Сортировка простым обменом наиболее эффективна когда массив уже частично отсортирован и когда элементов массива не много. Судя по полученным результатам и графику функции роста, зависимость количества операций от длины массива – квадратичная.

Варианты:

```

for j = 0 to A.length do
  min = j
  for i = j + 1 to A.length do
    if A[min] > A[i] then
      min = i
    end if
  end for
  swap A[min], A[j]
end for
    
```

```

for j = 0 to A.length do
  swapped = false
  for i = A.length - 1 downto j do
    if A[i - 1] > A[i] then
      swap A[i - 1], A[i]
      swapped = true
    end if
  end for
  break if not swapped
end for
    
```

```

for j = 1 to A.length do
  value = A[j]
  i = j - 1
  while i >= 0 and A[i] > value do
    A[i + 1] = A[i]
    A[i] = value
    i = i - 1
  end while
end for
    
```

Практическая работа №2. Оценка сложности алгоритмов поиска

Цель практической работы: Формирование навыков и умений составления эвристических алгоритмов, исследование и оценка сложности.

Задание:

Составить алгоритм нахождения кратчайшего пути на графе между тремя парами вершин методом динамического программирования

Эталон ответа:

```
class GFG {
    static int V = 9;
    int minDistance(int[] dist,
                   bool[] sptSet)
    {
        // Initialize min value
        int min = int.MaxValue, min_index = -1;

        for (int v = 0; v < V; v++)
            if (sptSet[v] == false && dist[v] <= min) {
                min = dist[v];
                min_index = v;
            }
        return min_index;
    }
    void printSolution(int[] dist, int n)
    {
        Console.WriteLine("Vertex   Distance "
                          + "from Source\n");
        for (int i = 0; i < V; i++)
            Console.WriteLine(i + " \t\t " + dist[i] + "\n");
    }
    void dijkstra(int[, ] graph, int src)
    {
        int[] dist = new int[V]; // The output array. dist[i]
        bool[] sptSet = new bool[V];
        for (int i = 0; i < V; i++) {
            dist[i] = int.MaxValue;
            sptSet[i] = false;
        }
        dist[src] = 0;
        for (int count = 0; count < V - 1; count++) {
            int u = minDistance(dist, sptSet);
            sptSet[u] = true;
            for (int v = 0; v < V; v++)
                if (!sptSet[v] && graph[u, v] != 0 &&
                    dist[u] != int.MaxValue && dist[u] + graph[u, v] < dist[v])
                    dist[v] = dist[u] + graph[u, v];
            printSolution(dist, V);
        }
    }
    // Driver Code
    public static void Main()
    {
        /* Let us create the example
        graph discussed above */
        int[, ] graph = new int[, ] { { 0, 4, 0, 0, 0, 0, 0, 8, 0 },
```

```

        { 4, 0, 8, 0, 0, 0, 0, 11, 0 },
        { 0, 8, 0, 7, 0, 4, 0, 0, 2 },
        { 0, 0, 7, 0, 9, 14, 0, 0, 0 },
        { 0, 0, 0, 9, 0, 10, 0, 0, 0 },
        { 0, 0, 4, 14, 10, 0, 2, 0, 0 },
        { 0, 0, 0, 0, 0, 2, 0, 1, 6 },
        { 8, 11, 0, 0, 0, 0, 1, 0, 7 },
        { 0, 0, 2, 0, 0, 0, 6, 7, 0 } };
    GFG t = new GFG();
    t.dijkstra(graph, 0);
}
}

```

Варианты:

<p>Составьте алгоритм на языке C#, который находит кратчайший путь между всеми парами вершин в графе с использованием динамического программирования. Нахождение кратчайшего пути с учетом весов ребер</p>
<p>Напишите алгоритм на языке C#, который находит кратчайший путь между двумя заданными вершинами взвешенного графа, используя метод динамического программирования. Нахождение кратчайшего пути с ограничениями на количество вершин</p>
<p>Разработайте алгоритм на языке C#, который находит кратчайший путь между двумя вершинами в графе с ограничением на количество промежуточных вершин, используя динамическое программирование. Нахождение кратчайшего пути с условием обхода всех верши</p>
<p>Напишите программу на языке C#, которая находит кратчайший путь, проходящий через все вершины графа ровно один раз, с использованием метода динамического программирования.</p>
<p>Составьте описание алгоритмов линейного и бинарного поиска, оцените их временную сложность и сравните эффективность поиска в отсортированных и неотсортированных массивах. Оценка сложности алгоритма поиска в графе</p>
<p>Рассмотрите алгоритмы поиска в графе (например, поиск в ширину и поиск в глубину) и определите их временную сложность. Объясните, как эти алгоритмы могут быть применены для различных задач поиска. Оценка сложности алгоритма поиска в дереве</p>
<p>Исследуйте алгоритмы поиска в дереве (например, обход в глубину и обход в ширину) и определите их временную сложность. Объясните, как выбор структуры дерева может влиять на эффективность поиска.</p>

Практическая работа №3. Оценка сложности рекурсивных алгоритмов

Цель практической работы: Изучить алгоритмы линейного и двоичного поиска, провести исследование временных характеристик, сравнить полученные результаты

Задание:

Разработайте рекурсивную функцию, подсчитывающую количество способов разбиения выпуклого многоугольника на треугольники непересекающимися диагоналями.

Эталон ответа:

```

using System;
class GFG
{
    static long binomialCoeff(int n, int k)
    {

```

```

long res = 1;

// Since C(n, k) = C(n, n-k)
if (k > n - k)
    k = n - k;
for (int i = 0; i < k; ++i)
{
    res *= (n - i);
    res /= (i + 1);
}
return res;
}
static long catalan( int n)
{
    // Calculate value of 2nCn
    long c = binomialCoeff(2 * n, n);

    // return 2nCn/(n+1)
    return c / (n + 1);
}
public static void Main()
{
    int n = 3;
    Console.WriteLine(catalan(n));
}
}

```

Варианты:

1. Разработайте рекурсивную функцию, подсчитывающую количество способов разбиения выпуклого многоугольника на треугольники непересекающимися диагоналями.
2. В Фибоначчиевой системе счисления числа формируются по правилам. Используются только символы 0 и 1; Каждый разряд соответствует элементу последовательности Фибоначчи 1, 2, 3, 5, 8, ..., то есть указывает на наличие или отсутствие такового; В соседних разрядах не могут стоять символы 1, так как это автоматически означает формирование следующего за ними разряда. Например, $1710 = 1310 + 310 + 110 = 100101ф$.
3. Составьте программу перевода числа из десятичной системы в Фибоначчиевую. Считать входные данные введенными корректно. Найдите подходящие дроби рационального числа x/y (x – неотрицательно, y – положительно). Например, то есть для $x = 5$, $y = 6$ ответом будет последовательность [0; 1, 5].
4. Вычислите определитель квадратной матрицы размера $n \times n$.
5. Найдите первые N чисел Фибоначчи двумя способами: с помощью рекурсии и с помощью итерации. Сравните эффективность алгоритмов
6. Напишите традиционную функцию умножения двух чисел и функцию, использующую только операцию сложения. Сравните эффективность алгоритмов.
7. Напишите метод, находящий максимальное из двух чисел, не используя операторы `if-else` или любые другие операторы сравнения. Оцените сложность алгоритма
8. Вычислите произведение элементов одномерного массива двумя способами: с помощью рекурсии и с помощью итерации. Оцените сложность алгоритма.

Рекурсивная функция Фибоначчи: Напишите рекурсивную функцию на C#, которая вычисляет n-ое число Фибоначчи. Затем обсудите сложность этого алгоритма и возможные способы оптимизации.

Рекурсивная функция сортировки: Реализуйте рекурсивную функцию сортировки (например, быструю сортировку или сортировку слиянием) на C# и обсудите ее сложность в среднем и в худшем случае.

Рекурсивный алгоритм поиска путей в графе: Напишите рекурсивную функцию на C#, которая ищет все возможные пути между двумя узлами в графе. Обсудите сложность этого алгоритма и его применимость к различным типам графов.

Рекурсивный алгоритм разбиения на подмножества: Разработайте рекурсивную функцию на C#, которая находит все возможные разбиения заданного множества на подмножества. Обсудите сложность этого алгоритма и его применимость к решению задач комбинаторики.

Практическая работа №4. Оценка сложности эвристических алгоритмов

Цель практической работы:

Задание:

Дана доска $n \times n$, содержащая n^2 полей. Конь, который ходит согласно шахматным правилам, помещается на поле с начальными координатами x_0, y_0 . Нужно покрыть всю доску ходами коня, т.е. вычислить обход доски, если он существует, из $n^2 - 1$ ходов, такой, что каждое поле посещается ровно один раз.

Эталон ответа:

```
using System;

class KnightsTour
{
    static int N = 8; // Change to the desired board size
    static int[] moveX = { 2, 1, -1, -2, -2, -1, 1, 2 };
    static int[] moveY = { 1, 2, 2, 1, -1, -2, -2, -1 };

    static bool IsSafe(int x, int y, int[,] sol)
    {
        return (x >= 0 && x < N && y >= 0 && y < N && sol[x, y] == -1);
    }

    static void PrintSolution(int[,] sol)
    {
        for (int i = 0; i < N; i++)
        {
            for (int j = 0; j < N; j++)
            {
                Console.Write(sol[i, j] + "\t");
            }
            Console.WriteLine();
        }
    }

    static bool SolveKnightsTour()
    {
        int[,] sol = new int[N, N];
        for (int x = 0; x < N; x++)
        {
```



```

    for (int y = 0; y < N; y++)
    {
        sol[x, y] = -1;
    }
}

sol[0, 0] = 0;

if (!SolveKnightsTourUtil(0, 0, 1, sol))
{
    Console.WriteLine("Solution does not exist");
    return false;
}
else
{
    PrintSolution(sol);
    return true;
}
}

static bool SolveKnightsTourUtil(int x, int y, int movei, int[,] sol)
{
    int nextX, nextY;
    if (movei == N * N)
    {
        return true;
    }

    for (int i = 0; i < 8; i++)
    {
        nextX = x + moveX[i];
        nextY = y + moveY[i];
        if (IsSafe(nextX, nextY, sol))
        {
            sol[nextX, nextY] = movei;
            if (SolveKnightsTourUtil(nextX, nextY, movei + 1, sol))
            {
                return true;
            }
            else
            {
                sol[nextX, nextY] = -1; // Backtrack
            }
        }
    }

    return false;
}

public static void Main()
{
    SolveKnightsTour();
}

```

```
}
```

Варианты:

Обход доски ходом коня: Поиск задач, где требуется обойти доску 8x8 ходом коня, покрывая все поля без повторов.

Шахматный обход конем: Использование этого термина также может помочь найти подобные задачи или алгоритмы.

Задачи на графы: Обход доски ходом коня может быть связан с задачами на графы, поэтому поиск задач в этой области также может быть полезным.

Практические и лабораторные работы Тема 1.3 Объектно-ориентированное программирование

Практическая работа №1. Работа с классами.

Цель практической работы: научиться определять классы и элементы класса.

Задание:

Создайте отдельный класс, соответствующий классу объектов из вашей предметной области (необходимо задать не менее 5 свойств для создаваемого класса). Добавьте метод для вывода описания объекта класса, используя метод ToString(). Этот метод должен в удобном для читателя виде перечислять все заданные значения свойств объекта класса.

Эталон ответа:

```
using System;
namespace LabRab_2
{
    public class Programmer
    {
        public string Name { get; set; } //имя
        public string Surname { get; set; } //фамилия
        public byte Age { get; set; } //возраст
        public byte Experience { get; set; } //стаж работы
        public string Language { get; set; } //язык на котором пишет
        public override string ToString()
        {
            return $"Имя: {Name}\r\nФаммилия: {Surname}\r\nВозраст: {Age} лет\r\nСтаж работы:
            {Experience} лет\r\nПишет на {Language}";
        }
    }
    class Program
    {
        static void Main(string[] args)
        {
            Programmer programmer = new Programmer();
            programmer.Name = "Вася";
            programmer.Surname = "Пупкин";
            programmer.Age = 25;
            programmer.Experience = 7;
            programmer.Language = "C#";
            Console.WriteLine(programmer.ToString());
        }
    }
}
```

Варианты:

Создайте отдельный класс, соответствующий классу объектов из вашей предметной

области (необходимо задать не менее 5 свойств для создаваемого класса). Добавьте метод для вывода описания объекта класса, используя метод ToString(). Этот метод должен в удобном для читателя виде перечислять все заданные значения свойств объекта класса.

1. Студент, преподаватель, персона, заведующий кафедрой
2. Служащий, персона, рабочий, инженер
3. Рабочий, кадры, инженер, администрация
4. Деталь, механизм, изделие, узел 5. Организация, страховая компания, нефтегазовая компания, завод
6. Журнал, книга, печатное издание, учебник
7. Тест, экзамен, выпускной экзамен, испытание
8. Место, область, город, мегаполис
9. Игрушка, продукт, товар, молочный продукт
10. Квитанция, накладная, документ, счет
11. Автомобиль, поезд, транспортное средство, экспресс
12. Двигатель, турбина, дизель, электромотор
13. Республика, монархия, королевство, государство
14. Млекопитающее, парнокопытное, птица, животное
15. Корабль, пароход, парусник, корвет
16. Дерево, растение, мох, гриб
17. Винтовка, гранатомет, оружие, кинжал.
18. Вертолет, дельтаплан, самолет, транспорт.
19. Дом, комната, жилье, квартира.
20. Радиоволны, электромагнитное излучение, ультрафиолет, рентгеновские лучи.
21. Математика, геометрия, наука, биология
22. Дождь, погода, снег, ветер
23. Доска, стройматериалы, кирпич, брус
24. Семья, нация, народность, общество
25. Полк, дивизия, армия, бригад

Практическая работа №2. Перегрузка методов.

Цель практической работы: научиться реализовывать перегрузку методов (method overloading) на языке C#

Задание:

Создать калькулятор с перегружаемыми методами

Эталон ответа:

```
class Calculator {
public void Add(int a, int b)
{
int result = a + b;
Console.WriteLine(
$"Result is
{result}");
}
public void Add(int a, int b, int c)
{
int result = a + b + c;
Console.WriteLine($"Result is {result}");
} public int Add(int a, int b, int c, int d)
{ int result = a + b + c + d;
Console.WriteLine($"Result is {result}");
return result; }
```

```
public void Add(double a, double b) {  
    double result = a + b;  
    Console.WriteLine($"Result is {result}");  
}  
}
```

Варианты:

Задание: Написать программу, для определения максимального из трех вещественных чисел, которые вводятся с клавиатуры, используя перегрузку метода для поиска максимального из двух значений.

Задание: Напишите перегруженные методы, в сокращенной форме (лямбда), которые принимают на вход два целых числа, находят их произведение и в зависимости от типа прибавляют к результату: 1 – для uint, 2 – для int, 3 – для long.

Задание: Определите сигнатуру метода:

Практическая работа №3. Определение операций в классе.

Цель практической работы: формирование целостного представления о сущности и назначении классов, основных операциях классов, овладение практическими навыками работы с классами в C#.

Задание:

Создать и подключить библиотеку классов. Сформировать dll файл.

Эталон ответа:

```
using System;  
using MyLib;  
// подключение пространства имен из библиотеки классов namespace HelloApp  
{  
    class Program {  
        static void Main(string[] args)  
        {  
            Person tom = new Person  
            { name = "Tom", age = 35 };  
            Console.WriteLine(tom.name);  
        }  
    }  
}
```

Варианты:

Создать и подключить библиотеку классов. Сформировать dll файл.

Вывести на экран данные о номере зачетной книжки

Практическая работа №4. Создание наследованных классов

Цель практической работы: : ознакомление с основой объектного подхода в языке C#, максимальное использование наследования.

Задание:

Реализовать конструкцию базового и наследованного класса

Эталон ответа:

```
class Vehicle // base class (parent)  
{  
    public string brand = "Ford"; // Vehicle field  
    public void honk() // Vehicle method  
    {  
        Console.WriteLine("Tuut, tuut!");  
    }  
}
```

```

class Car : Vehicle // derived class (child)
{
    public string modelName = "Mustang"; // Car field
}

class Program
{
    static void Main(string[] args)
    {
        // Create a myCar object
        Car myCar = new Car();

        // Call the honk() method (From the Vehicle class) on the myCar object
        myCar.honk();

        // Display the value of the brand field (from the Vehicle class) and the value of the
        modelName from the Car class
        Console.WriteLine(myCar.brand + " " + myCar.modelName);
    }
}

```

Варианты:

1. Студент, преподаватель, персона, заведующий кафедрой
2. Служащий, персона, рабочий, инженер
3. Рабочий, кадры, инженер, администрация
4. Деталь, механизм, изделие, узел 5. Организация, страховая компания, нефтегазовая компания, завод
6. Журнал, книга, печатное издание, учебник
7. Тест, экзамен, выпускной экзамен, испытание
8. Место, область, город, мегаполис
9. Игрушка, продукт, товар, молочный продукт
10. Квитанция, накладная, документ, счет
11. Автомобиль, поезд, транспортное средство, экспресс
12. Двигатель, турбина, дизель, электромотор
13. Республика, монархия, королевство, государство
14. Млекопитающее, парнокопытное, птица, животное
15. Корабль, пароход, парусник, корвет
16. Дерево, растение, мох, гриб
17. Винтовка, гранатомет, оружие, кинжал.
18. Вертолет, дельтаплан, самолет, транспорт.
19. Дом, комната, жилье, квартира.
20. Радиоволны, электромагнитное излучение, ультрафиолет, рентгеновские лучи.
21. Математика, геометрия, наука, биология
22. Дождь, погода, снег, ветер
23. Доска, стройматериалы, кирпич, брус
24. Семья, нация, народность, общество
25. Полк, дивизия, армия, бригад

Практическая работа №5. Работа с объектами через интерфейсы

Цель практической работы: ознакомление с основой объектного подхода в языке C#, максимальное использование интерфейсов.

Задание:

Разработать классы и реализовать интерфейсы.

Эталон ответа:

```
class Program
{
    static void Main(string[] args)
    {
        IMovable tom = new Person();
        Car tesla = new Car(); tom.Move();
        // Walking
        tesla.Move();
        // Driving
    }
}
interface
IMovable
{
    void Move()
    { Console.WriteLine("Walking");
    }
}
class Person : IMovable
{
}
class Car : IMovable
{
    public void Move()
    {
        Console.WriteLine("Driving");
    }
}
```

Варианты:

1. Студент, преподаватель, персона, заведующий кафедрой
2. Служащий, персона, рабочий, инженер
3. Рабочий, кадры, инженер, администрация
4. Деталь, механизм, изделие, узел 5. Организация, страховая компания, нефтегазовая компания, завод
6. Журнал, книга, печатное издание, учебник
7. Тест, экзамен, выпускной экзамен, испытание
8. Место, область, город, мегаполис
9. Игрушка, продукт, товар, молочный продукт
10. Квитанция, накладная, документ, счет
11. Автомобиль, поезд, транспортное средство, экспресс
12. Двигатель, турбина, дизель, электромотор
13. Республика, монархия, королевство, государство
14. Млекопитающее, парнокопытное, птица, животное
15. Корабль, пароход, парусник, корвет
16. Дерево, растение, мох, гриб
17. Винтовка, гранатомет, оружие, кинжал.
18. Вертолет, дельтаплан, самолет, транспорт.
19. Дом, комната, жилье, квартира.
20. Радиоволны, электромагнитное излучение, ультрафиолет, рентгеновские лучи.

21. Математика, геометрия, наука, биология
22. Дождь, погода, снег, ветер
23. Доска, стройматериалы, кирпич, брус
24. Семья, нация, народность, общество
25. Полк, дивизия, армия, бригад

Практическая работа №6. Использование стандартных интерфейсов

Цель практической работы: ознакомление с основой объектного подхода в языке C#, максимальное использование интерфейсов.

Задание:

Создать класс и реализовать интерфейс на основе класса. Для определения интерфейса используется ключевое слово `interface`. Как правило, названия интерфейсов в C# начинаются с заглавной буквы I, например, `IComparable`, `IEnumerable` (так называемая венгерская нотация), однако это не обязательное требование, а больше стиль программирования.

Эталон ответа:

```
interface IAnimal
{
    void animalSound(); // interface method (does not have a body)
}

class Pig : IAnimal
{
    public void animalSound()
    {
        Console.WriteLine("The pig says: wee wee");
    }
}

class Program
{
    static void Main(string[] args)
    {
        Pig myPig = new Pig(); // Create a Pig object
        myPig.animalSound();
    }
}
```

Варианты:

1. Студент, преподаватель, персона, заведующий кафедрой
2. Служащий, персона, рабочий, инженер
3. Рабочий, кадры, инженер, администрация
4. Деталь, механизм, изделие, узел 5. Организация, страховая компания, нефтегазовая компания, завод
6. Журнал, книга, печатное издание, учебник
7. Тест, экзамен, выпускной экзамен, испытание
8. Место, область, город, мегаполис
9. Игрушка, продукт, товар, молочный продукт
10. Квитанция, накладная, документ, счет
11. Автомобиль, поезд, транспортное средство, экспресс
12. Двигатель, турбина, дизель, электромотор
13. Республика, монархия, королевство, государство

14. Млекопитающее, парнокопытное, птица, животное
15. Корабль, пароход, парусник, корвет
16. Дерево, растение, мох, гриб
17. Винтовка, гранатомет, оружие, кинжал.
18. Вертолет, дельтаплан, самолет, транспорт.
19. Дом, комната, жилье, квартира.
20. Радиоволны, электромагнитное излучение, ультрафиолет, рентгеновские лучи.
21. Математика, геометрия, наука, биология
22. Дождь, погода, снег, ветер
23. Доска, стройматериалы, кирпич, брус
24. Семья, нация, народность, общество
25. Полк, дивизия, армия, бригад

Практическая работа №7. Работа с типом данных структура

Цель практической работы: изучить использование типов данных структура на практике

Задание:

составить инвентарную ведомость игрушек, включив следующие данные: название игрушки, ее стоимость (в руб.), возрастные границы детей, для которых предназначена игрушка. Вывести в новый файл информацию о тех игрушках, которые предназначены для детей от **N** до **M** лет, отсортировав их по стоимости.

Эталон ответа:

```
Файл Toy.cs
using System;
using static System.Math;
namespace ToysComparer
{
    public struct Toy: IComparable
    {
        public string Name { get; set; }
        public int Price { get; set; }
        public int AgeMin { get; set; }
        public int AgeMax { get; set; }
        public int CompareTo(object other)
        {
            if (other is not Toy)
                throw new ArgumentException("Сравнивать можно только структуры типа Toy");
            return Sign(Price - ((Toy)other).Price);
        }
        public override string ToString()
        {
            return $"{Name};{Price};{AgeMin};{AgeMax}";
        }
    }
}

Файл Program.cs
using System;
using System.Linq;
using System.IO;
namespace ToysComparer
{
    internal class Program
    {
```



```

public static void WriteFile(string fileName, Toy[] toys)
{
    using (StreamWriter sw = new StreamWriter(fileName))
    {
        foreach (Toy toy in toys)
            sw.WriteLine(toy.ToString());
    }
}

public static Toy[] ReadFile(string fileName)
{
    string[] lines = File.ReadAllLines(fileName);
    Toy[] toys = new Toy[lines.Length];
    int i = 0;
    foreach (string s in lines)
    {
        string[] toyFields = s.Split(new[] { ';' });
        toys[i].Name = toyFields[0];
        toys[i].Price = Convert.ToInt32(toyFields[1]);
        toys[i].AgeMin = Convert.ToInt32(toyFields[2]);
        toys[i].AgeMax = Convert.ToInt32(toyFields[3]);
        i++;
    }
    return toys;
}

public static void Display(Toy[] toys)
{
    foreach (Toy toy in toys)
    {
        Console.WriteLine($"Наименование: {toy.Name}\tСтоимость: {toy.Price}
руб.\tВозрастные ограничения: от {toy.AgeMin} до {toy.AgeMax} лет");
    }
}

static void Main(string[] args)
{
    string inputFile = "input.txt";
    string outputFile = "output.txt";
    Toy[] toys = ReadFile(inputFile);
    //выводим исходный массив игрушек
    Console.WriteLine("-----Исходный массив-----");
    Display(toys);
    Console.WriteLine("-----");
    Console.WriteLine("Введите минимальный возраст");
    int MinAge = Convert.ToInt32(Console.ReadLine());
    Console.WriteLine("Введите максимальный возраст");
    int MaxAge = Convert.ToInt32(Console.ReadLine());
    //Выбираем все игрушки с допустимым диапазоном возрастов
    var selected = toys.Where(w => (Enumerable.Range(w.AgeMin,
w.AgeMax).Contains(MinAge)&&(Enumerable.Range(w.AgeMin,
w.AgeMax).Contains(MaxAge))));
    Toy[] outToys = selected.ToArray();
    Console.WriteLine("-----Итоговый массив-----");
    Array.Sort(outToys); //сортируем массив
}

```

```
WriteFile(outputFile, outToys);//записываем массив в файл
Display(outToys);
Console.WriteLine("-----");
}}}
```

Варианты:

1. Студент, преподаватель, персона, заведующий кафедрой
2. Служащий, персона, рабочий, инженер
3. Рабочий, кадры, инженер, администрация
4. Деталь, механизм, изделие, узел
5. Организация, страховая компания, нефтегазовая компания, завод
6. Журнал, книга, печатное издание, учебник
7. Тест, экзамен, выпускной экзамен, испытание
8. Место, область, город, мегаполис
9. Игрушка, продукт, товар, молочный продукт
10. Квитанция, накладная, документ, счет
11. Автомобиль, поезд, транспортное средство, экспресс
12. Двигатель, турбина, дизель, электромотор
13. Республика, монархия, королевство, государство
14. Млекопитающее, парнокопытное, птица, животное
15. Корабль, пароход, парусник, корвет
16. Дерево, растение, мох, гриб
17. Винтовка, гранатомет, оружие, кинжал.
18. Вертолет, дельтаплан, самолет, транспорт.
19. Дом, комната, жилье, квартира.
20. Радиоволны, электромагнитное излучение, ультрафиолет, рентгеновские лучи.
21. Математика, геометрия, наука, биология
22. Дождь, погода, снег, ветер
23. Доска, стройматериалы, кирпич, брус
24. Семья, нация, народность, общество
25. Полк, дивизия, армия, бригад

Практическая работа №8. Коллекции. Параметризованные классы

Цель практической работы: изучить представление и правила работы параметризованными классами в с#

Задание:

Реализовать на языке С# параметризованный класс «Матрица», типы элементов которого могут быть заданы в соответствии с вариантом.

Эталон ответа:

```
using System;
class Substance
{
    public double Mass { get; set; }
    public double Temperature { get; set; }

    public static Substance operator +(Substance a, Substance b)
    {
        Substance res = new Substance();
        res.Mass = a.Mass + b.Mass;
        res.Temperature = (a.Temperature * a.Mass + b.Temperature * b.Mass) / (a.Mass +
        b.Mass);
        return res;
    }
}
```

```

    }
    public static Substance RandVal()
    {
        Substance res = new Substance();
        Random rand = new Random();
        res.Mass = rand.Next(5, 2005) / 100.0;
        res.Temperature = rand.Next(10, 2010) / 100.0;
        return res;
    }
    public override string ToString()
    {
        return $"M={Mass:F2};T={Temperature:F2}";
    }
}
class Matrix<T> where T : Substance, new()
{
    private T[,] m;
    public Matrix()
    {
        m = new T[4, 4];
        for (int i = 0; i < 4; i++)
        {
            for (int j = 0; j < 4; j++)
            {
                m[i, j] = T.RandVal();
            }
        }
    }
    public void Print()
    {
        Console.WriteLine("Matrix");
        for (int i = 0; i < 4; i++)
        {
            for (int j = 0; j < 4; j++)
            {
                Console.Write($"{m[i, j]} ");
            }
            Console.WriteLine();
        }
        Console.WriteLine();
    }
    public static Matrix<T> operator +(Matrix<T> a, Matrix<T> b)
    {
        Matrix<T> res = new Matrix<T>();
        for (int i = 0; i < 4; i++)
        {
            for (int j = 0; j < 4; j++)
            {
                res.m[i, j] = a.m[i, j] + b.m[i, j];
            }
        }
        return res;
    }
}

```

```

    }
}
class Program
{
    static void Main(string[] args)
    {
        Matrix<Substance> matrix1 = new Matrix<Substance>();
        Matrix<Substance> matrix2 = new Matrix<Substance>();
        Matrix<Substance> matrix3 = matrix1 + matrix2;
        matrix1.Print();
        matrix2.Print();
        matrix3.Print();
    }
}

```

Варианты:

создать массив объектов CollectionType, запросы – найти коллекции размера n; найти максимальную и минимальную коллекцию в массиве по количеству элементов. Обобщенная коллекция – LinkedList
создать массив объектов CollectionType, запросы – найти коллекции с отрицательными элементами (выбрать любое поле объекта), найти максимальную и минимальную коллекцию в массиве, содержащую указанный элемент. Обобщенная коллекция – Dictionary.
Создать массив объектов CollectionType, запросы - найти количество коллекций равных заданному размеру, найти максимальную и минимальную коллекцию в массиве. Обобщенная коллекция – List
Создать массив объектов CollectionType, запросы - найти количество коллекций, содержащих только 2 элемента, найти максимальную и минимальную коллекцию в массиве по заданному значению поля объекта (можно выбрать любое поле). Обобщенная коллекция – List
Создать массив объектов CollectionType, запросы - найти количество коллекций, содержащих заданное значение (выбрать любое поле объекта), найти максимальную и минимальную коллекцию в массиве. Обобщенная коллекция – LinkedList
Создать массив объектов CollectionType, запросы - найти количество коллекций, сумма которых больше указанного значения (для суммирования выбрать любое поле объекта), найти максимальную и минимальную коллекцию в массиве. Обобщенная коллекция – ArrayList
Создать массив объектов CollectionType, запросы - найти количество коллекций, содержащих указанный элемент, найти максимальную коллекцию, содержащую указанный элемент. Обобщенная коллекция – Dictionary.

Практическая работа №9. Использование регулярных выражений

Цель практической работы: Получить практические навыки использования регулярных выражений

Задание:

Составить программу в соответствии с условиями поиска

Эталон ответа:

```

string pattern = @"не";
string text = "Не может быть совсем не может быть";
RegexOptions option=RegexOptions.IgnoreCase;
Regex newReg = new Regex(pattern,option); MatchCollection matches =
newReg.Matches(text);

```

```
foreach(Match mat in matches)
{
Console.WriteLine("Значениенайденногообъекта {0}",mat.Value);
}
Console.WriteLine("Числонайденныхсовпадений {0}",matches.Count);
```

Варианты:

1 Дана строка, проверить является ли она электронным адресом.
2 Дан текстовый файл, состоящий из нескольких строк. Найти в нем слово «мир», в другой текстовый файл записать это слово и количество встречающихся экземпляров.
3 Дан текст, найти в нем слово «авто» и однокоренные с ним слова, и выделить их цветом.
4 Дан текстовый файл, состоящий из нескольких строк. Проверить начинается ли он с введенного слова. Выдать сообщение о результате проверки
5 Дан текст, найти в нем слова, заканчивающиеся на букву «а», и вывести эти слова.
6 Дан текстовый файл, состоящий из нескольких строк. Проверить заканчивается ли он введенным словом. Выдать сообщение о результате проверки
7 Дан текст и слово, вывести строки, в которых оно встречается и заканчивающиеся «?».
8 Дан текстовый файл, состоящий из нескольких строк. Найти в нем введенное слово, организовав поиск с конца в начало.
9 Дана строка, проверить является ли она номером телефона.
10 Дан текст и слово, вывести все слова из текста не абсолютно совпадающие с введенным словом.
11 Дана строка, проверить является ли она IP-адресом.
12 Дан текст, найти в нем введенное слово, написанное прописными буквами.
13 Дан текстовый файл, состоящий из нескольких строк. Найти в нем слово «куб». В другой файл записать строки, в которых оно встречается и заканчивающиеся «!».
14 Дан текст, найти в нем слова, содержащие введенные буквы, и вывести их количество.

Практическая работа №10. Операции со списками

Цель практической работы: получить навыки работы со списками

Задание:

Создать консольное приложение на смоделированной структуре из ЛР 5, осуществляющее поиск по экземплярам (с использованием коллекций List<T> или ArrayList, без LINQ).

Приложение должно содержать:

- Экземпляры классов, по которым будет осуществляться поиск (см.пример).
- Интерфейс командной строки: вопрос о намерении пользователя осуществить поиск (y/n), уточнение класса экземпляра (точное название), поиск по любому из атрибутов, содержащихся в классах (число, текст, дата).
- После выбора экземпляра пользователь вводит любые данные для соответствующей структуры (например, название жанра литературы или ФИО автора), поиск должен осуществляться по всем свойствам.
- Логичный вывод релевантной информации, связанной с искомым объектом. Студент должен самостоятельно определить, что будет выводить приложение.

Эталон ответа:

```
using System;
using System.Collections.Generic;
// Simple business object. A PartId is used to identify the type of part
// but the part name can change.
public class Part : IEquatable<Part>
```

```

{
    public string PartName { get; set; }

    public int PartId { get; set; }

    public override string ToString()
    {
        return "ID: " + PartId + " Name: " + PartName;
    }
    public override bool Equals(object obj)
    {
        if (obj == null) return false;
        Part objAsPart = obj as Part;
        if (objAsPart == null) return false;
        else return Equals(objAsPart);
    }
    public override int GetHashCode()
    {
        return PartId;
    }
    public bool Equals(Part other)
    {
        if (other == null) return false;
        return (this.PartId.Equals(other.PartId));
    }
    // Should also override == and != operators.
}

public class Example
{
    public static void Main()
    {
        // Create a list of parts.
        List<Part> parts = new List<Part>();
        // Add parts to the list.
        parts.Add(new Part() { PartName="crank arm", PartId=1234});
        parts.Add(new Part() { PartName = "chain ring", PartId = 1334 });
        parts.Add(new Part() { PartName = "regular seat", PartId = 1434 });
        parts.Add(new Part() { PartName = "banana seat", PartId = 1444 });
        parts.Add(new Part() { PartName = "cassette", PartId = 1534 });
        parts.Add(new Part() { PartName = "shift lever", PartId = 1634 });
        // Write out the parts in the list. This will call the overridden ToString method
        // in the Part class.
        Console.WriteLine();
        foreach (Part aPart in parts)
        {
            Console.WriteLine(aPart);
        }
        // Check the list for part #1734. This calls the IEquatable.Equals method
        // of the Part class, which checks the PartId for equality.
        Console.WriteLine("\nContains(\"1734\"): {0}",
            parts.Contains(new Part {PartId=1734, PartName="" }));
        // Insert a new item at position 2.
    }
}

```

```

Console.WriteLine("\nInsert(2, \"1834\")");
parts.Insert(2, new Part() { PartName = "brake lever", PartId = 1834 });
//Console.WriteLine();
foreach (Part aPart in parts)
{
    Console.WriteLine(aPart);
}
Console.WriteLine("\nParts[3]: {0}", parts[3]);
Console.WriteLine("\nRemove(\"1534\")");
// This will remove part 1534 even though the PartName is different,
// because the Equals method only checks PartId for equality.
parts.Remove(new Part(){PartId=1534, PartName="cogs"});

Console.WriteLine();
foreach (Part aPart in parts)
{
    Console.WriteLine(aPart);
}
Console.WriteLine("\nRemoveAt(3)");
// This will remove the part at index 3.
parts.RemoveAt(3);

Console.WriteLine();
foreach (Part aPart in parts)
{
    Console.WriteLine(aPart);
}
/*
    ID: 1234 Name: crank arm
    ID: 1334 Name: chain ring
    ID: 1434 Name: regular seat
    ID: 1444 Name: banana seat
    ID: 1534 Name: cassette
    ID: 1634 Name: shift lever
    Contains("1734"): False
    Insert(2, "1834")
    ID: 1234 Name: crank arm
    ID: 1334 Name: chain ring
    ID: 1834 Name: brake lever
    ID: 1434 Name: regular seat
    ID: 1444 Name: banana seat
    ID: 1534 Name: cassette
    ID: 1634 Name: shift lever
    Parts[3]: ID: 1434 Name: regular seat
    Remove("1534")
    ID: 1234 Name: crank arm
    ID: 1334 Name: chain ring
    ID: 1834 Name: brake lever
    ID: 1434 Name: regular seat
    ID: 1444 Name: banana seat
    ID: 1634 Name: shift lever
    RemoveAt(3)

```

```
ID: 1234 Name: crank arm
ID: 1334 Name: chain ring
ID: 1834 Name: brake lever
ID: 1444 Name: banana seat
ID: 1634 Name: shift lever
*/
}
}
```

Варианты:

1. Студент, преподаватель, персона, заведующий кафедрой
2. Служащий, персона, рабочий, инженер
3. Рабочий, кадры, инженер, администрация
4. Деталь, механизм, изделие, узел
5. Организация, страховая компания, нефтегазовая компания, завод
6. Журнал, книга, печатное издание, учебник
7. Тест, экзамен, выпускной экзамен, испытание
8. Место, область, город, мегаполис
9. Игрушка, продукт, товар, молочный продукт
10. Квитанция, накладная, документ, счет
11. Автомобиль, поезд, транспортное средство, экспресс
12. Двигатель, турбина, дизель, электромотор
13. Республика, монархия, королевство, государство
14. Млекопитающее, парнокопытное, птица, животное
15. Корабль, пароход, парусник, корвет
16. Дерево, растение, мох, гриб
17. Винтовка, гранатомет, оружие, кинжал.
18. Вертолет, дельтаплан, самолет, транспорт.
19. Дом, комната, жилье, квартира.
20. Радиоволны, электромагнитное излучение, ультрафиолет, рентгеновские лучи.
21. Математика, геометрия, наука, биология
22. Дождь, погода, снег, ветер
23. Доска, стройматериалы, кирпич, брус
24. Семья, нация, народность, общество
25. Полк, дивизия, армия, бригад

Практические и лабораторные работы Тема 1.4 Паттерны проектирования

Практическая работа №1 Использование основных шаблонов

Цель практической работы: Изучение синтаксиса и принципов работы с шаблонами классов C#.

Задание:

Определить функционал для реализации шаблона

Эталон ответа:

```
public abstract class Vehicle { public abstract void Move(); } public class Car : Vehicle {
public override void Move() { Console.WriteLine("Машина едет"); } } public class Bus :
Vehicle { public override void Move() { Console.WriteLine("Автобус едет"); } } public class
Tram : Vehicle { public override void Move() { Console.WriteLine("Трамвай едет"); } }
```

Варианты:

1. Студент, преподаватель, персона, заведующий кафедрой
2. Служащий, персона, рабочий, инженер
3. Рабочий, кадры, инженер, администрация

4. Деталь, механизм, изделие, узел 5. Организация, страховая компания, нефтегазовая компания, завод
6. Журнал, книга, печатное издание, учебник
7. Тест, экзамен, выпускной экзамен, испытание
8. Место, область, город, мегаполис
9. Игрушка, продукт, товар, молочный продукт
10. Квитанция, накладная, документ, счет
11. Автомобиль, поезд, транспортное средство, экспресс
12. Двигатель, турбина, дизель, электромотор
13. Республика, монархия, королевство, государство
14. Млекопитающее, парнокопытное, птица, животное
15. Корабль, пароход, парусник, корвет
16. Дерево, растение, мох, гриб
17. Винтовка, гранатомет, оружие, кинжал.
18. Вертолет, дельтаплан, самолет, транспорт.
19. Дом, комната, жилье, квартира.
20. Радиоволны, электромагнитное излучение, ультрафиолет, рентгеновские лучи.
21. Математика, геометрия, наука, биология
22. Дождь, погода, снег, ветер
23. Доска, стройматериалы, кирпич, брус
24. Семья, нация, народность, общество
25. Полк, дивизия, армия, бригад

Практическая работа №2 2. Использование порождающих шаблонов.

Цель практической работы: Изучение синтаксиса и принципов работы с шаблонами классов C#.

Задание:

Определить функционал для реализации шаблона

Эталон ответа:

```
abstract class Product { } class ConcreteProductA : Product { } class ConcreteProductB :
Product { } abstract class Creator { public abstract Product FactoryMethod(); } class
ConcreteCreatorA : Creator { public override Product FactoryMethod() { return new
ConcreteProductA(); } } class ConcreteCreatorB : Creator { public override Product
FactoryMethod() { return new ConcreteProductB(); } }
```

Варианты:

1. Студент, преподаватель, персона, заведующий кафедрой
2. Служащий, персона, рабочий, инженер
3. Рабочий, кадры, инженер, администрация
4. Деталь, механизм, изделие, узел 5. Организация, страховая компания, нефтегазовая компания, завод
6. Журнал, книга, печатное издание, учебник
7. Тест, экзамен, выпускной экзамен, испытание
8. Место, область, город, мегаполис
9. Игрушка, продукт, товар, молочный продукт
10. Квитанция, накладная, документ, счет
11. Автомобиль, поезд, транспортное средство, экспресс
12. Двигатель, турбина, дизель, электромотор
13. Республика, монархия, королевство, государство
14. Млекопитающее, парнокопытное, птица, животное
15. Корабль, пароход, парусник, корвет
16. Дерево, растение, мох, гриб

17. Винтовка, гранатомет, оружие, кинжал.
18. Вертолет, дельтаплан, самолет, транспорт.
19. Дом, комната, жилье, квартира.
20. Радиоволны, электромагнитное излучение, ультрафиолет, рентгеновские лучи.
21. Математика, геометрия, наука, биология
22. Дождь, погода, снег, ветер
23. Доска, стройматериалы, кирпич, брус
24. Семья, нация, народность, общество
25. Полк, дивизия, армия, бригад

Практическая работа №3 3. Использование структурных шаблонов.

Цель практической работы: Изучение синтаксиса и принципов работы с шаблонами классов C#.

Задание:

Определить функционал для реализации шаблона

Эталон ответа:

```
abstract class Component { public abstract void Operation(); }
class ConcreteComponent : Component { public override void Operation() {} }
abstract class Decorator : Component {
protected Component component;
public void SetComponent(Component component) {
this.component = component;
}
public override void Operation() { if (component != null)
component.Operation(); }
}
class ConcreteDecoratorA : Decorator { public override void
Operation() { base.Operation(); } }
class ConcreteDecoratorB : Decorator { public override
void Operation() { base.Operation(); } }
```

Варианты:

1. Студент, преподаватель, персона, заведующий кафедрой
2. Служащий, персона, рабочий, инженер
3. Рабочий, кадры, инженер, администрация
4. Деталь, механизм, изделие, узел 5. Организация, страховая компания, нефтегазовая компания, завод
6. Журнал, книга, печатное издание, учебник
7. Тест, экзамен, выпускной экзамен, испытание
8. Место, область, город, мегаполис
9. Игрушка, продукт, товар, молочный продукт
10. Квитанция, накладная, документ, счет
11. Автомобиль, поезд, транспортное средство, экспресс
12. Двигатель, турбина, дизель, электромотор
13. Республика, монархия, королевство, государство
14. Млекопитающее, парнокопытное, птица, животное
15. Корабль, пароход, парусник, корвет
16. Дерево, растение, мох, гриб
17. Винтовка, гранатомет, оружие, кинжал.
18. Вертолет, дельтаплан, самолет, транспорт.
19. Дом, комната, жилье, квартира.
20. Радиоволны, электромагнитное излучение, ультрафиолет, рентгеновские лучи.
21. Математика, геометрия, наука, биология
22. Дождь, погода, снег, ветер
23. Доска, стройматериалы, кирпич, брус
24. Семья, нация, народность, общество
25. Полк, дивизия, армия, бригад

Практическая работа №4 4. Использование поведенческих шаблонов.

Цель практической работы: Изучение синтаксиса и принципов работы с шаблонами классов C#.

Задание:

Определить функционал для реализации шаблона

Эталон ответа:

```
public interface IStrategy { void Algorithm(); } public class ConcreteStrategy1 : IStrategy {
public void Algorithm() {} } public class ConcreteStrategy2 : IStrategy { public void
Algorithm() {} } public class Context { public IStrategy ContextStrategy { get; set; } public
Context(IStrategy _strategy) { ContextStrategy = _strategy; } public void ExecuteAlgorithm()
{ ContextStrategy.Algorithm(); } }
```

Варианты:

1. Студент, преподаватель, персона, заведующий кафедрой
2. Служащий, персона, рабочий, инженер
3. Рабочий, кадры, инженер, администрация
4. Деталь, механизм, изделие, узел
5. Организация, страховая компания, нефтегазовая компания, завод
6. Журнал, книга, печатное издание, учебник
7. Тест, экзамен, выпускной экзамен, испытание
8. Место, область, город, мегаполис
9. Игрушка, продукт, товар, молочный продукт
10. Квитанция, накладная, документ, счет
11. Автомобиль, поезд, транспортное средство, экспресс
12. Двигатель, турбина, дизель, электромотор
13. Республика, монархия, королевство, государство
14. Млекопитающее, парнокопытное, птица, животное
15. Корабль, пароход, парусник, корвет
16. Дерево, растение, мох, гриб
17. Винтовка, гранатомет, оружие, кинжал.
18. Вертолет, дельтаплан, самолет, транспорт.
19. Дом, комната, жилье, квартира.
20. Радиоволны, электромагнитное излучение, ультрафиолет, рентгеновские лучи.
21. Математика, геометрия, наука, биология
22. Дождь, погода, снег, ветер
23. Доска, стройматериалы, кирпич, брус
24. Семья, нация, народность, общество
25. Полк, дивизия, армия, бригад

Практические и лабораторные работы Тема 1.5 Событийно управляемое программирование

Практическая работа №1. Разработка приложения с использованием текстовых компонентов

Цель практической работы: получение навыка разработки приложения с использованием текстовых компонентов

Задание:

Разработать консольное приложение калькулятор с возможностью выбора операции.

Эталон ответа:

```
using System;
namespace Calculator
{
class Calculator
```

```

{
public static double DoOperation(double num1, double num2, string op)
{
double result = double.NaN;
switch (op)
{
case "a": result = num1 + num2;
break;
case "s": result = num1 - num2; break;
case "m": result = num1 * num2;
break;
case "d":
if (num2 != 0) { result = num1 / num2;
}
break;
default:
break;
} return result;
}
}
class Program
{
static void Main(string[] args)
{
bool endApp = false;
Console.WriteLine("Console Calculator in C#\r");
Console.WriteLine("-----\n");
while (!endApp) {
// Declare variables and set to empty.
string numInput1 = "";
string numInput2 = "";
double result = 0;
Console.Write("Type a number, and then press Enter: "); numInput1 = Console.ReadLine();
double cleanNum1 = 0; while (!double.TryParse(numInput1, out cleanNum1))
{
Console.Write("This is not valid input. Please enter an integer value: ");
numInput1 = Console.ReadLine(); }
Console.Write("Type another number, and then press Enter: ");
numInput2 = Console.ReadLine();
double cleanNum2 = 0;
while (!double.TryParse(numInput2, out cleanNum2))
{
Console.Write("This is not valid input. Please enter an integer value: ");
numInput2 = Console.ReadLine();
}
Console.WriteLine("Choose an operator from the following list:");
Console.WriteLine("\ta - Add");
Console.WriteLine("\ts - Subtract");
Console.WriteLine("\tm - Multiply");
Console.WriteLine("\td - Divide"); Console.Write("Your option? ");
string op = Console.ReadLine();
try { result = Calculator.DoOperation(cleanNum1, cleanNum2, op); if (double.IsNaN(result))

```

```

{ Console.WriteLine("This operation will result in a mathematical error.\n");
}
else Console.WriteLine("Your result: {0:0.##}\n", result); } catch (Exception e)
{
Console.WriteLine("Oh no! An exception occurred trying to do the math.\n - Details: " +
e.Message);
}
Console.WriteLine("-----\n");
Console.Write("Press 'n' and Enter to close the app, or press any other key and Enter to
continue: "); if (Console.ReadLine() == "n") endApp = true;
Console.WriteLine("\n");
} return ; } } }

```

Варианты:

1. Счет в банке представляет собой структуру с полями: номер счета, код счета, фамилия владельца, сумма на счете, дата открытия счета, годовой процент начисления. Поиск по номеру счета, дате открытия и владельцу.

2. Запись о товаре на складе представляет собой структуру с полями: номер склада, код товара, наименование товара, дата поступления на склад, срок хранения в днях, количество единиц товара, цена за единицу товара. Поиск по номеру склада, коду товара, дате поступления и сроку хранения (просроченные и не просроченные товары).

3. Запись о преподаваемой дисциплине представляется структурой: код дисциплины в учебном плане, наименование дисциплины, фамилия преподавателя, код группы, количество студентов в группе, количество часов лекций, количество часов практики, наличие курсовой работы, вид итогового контроля (зачет или экзамен). Зачет 0,35 ч на одного студента; экзамен — 0,5 ч на студента. Поиск осуществлять по фамилии преподавателя, коду группы, наличию курсовой, виду итогового контроля.

4. Информационная запись о книге, выданной на руки абоненту, представляет собой структуру следующего вида: номер читательского билета, фамилия абонента, дата выдачи, срок возврата (количество дней), автор, название, год издания, издательство, цена. Поиск по полям: номер читательского билета, автор, издательство, дата возврата (просроченные).

5. Информационная запись о файле содержит поля: каталог, имя файла, расширение, дата и время создания, атрибуты «только чтение», «скрытый», «системный», признак удаления, количество выделенных секторов (размер сектора принять равным 512 байт). Поиск выполнять по каталогу, дате создания, по признаку удаления.

6. Разовый платеж за телефонный разговор является структурой с полями: фамилия плательщика, номер телефона, дата разговора, тариф за минуту разговора, скидка (в процентах), время начала разговора, время окончания разговора. Поиск по фамилии, дате разговора, номеру телефона.

7. Модель компьютера характеризуется кодом и названием марки компьютера, типом процессора, частотой работы процессора, объемом оперативной памяти, объемом жесткого диска, объемом памяти видеокарты, стоимостью компьютера в условных единицах и количеством экземпляров, имеющих в наличии. Поиск по типу процессора, объему ОЗУ, памяти видеокарты и жесткого диска.

8. Список абонентов сети кабельного телевидения состоит из элементов следующей структуры: фамилия, район, адрес, телефон, номер договора, дата заключения договора, оплата установки, абонентская плата ежемесячно, дата последнего платежа. Поиск по фамилии, району, дате заключения договора, дате последнего платежа.

9. Сотрудник представлен структурой Person с полями: табельный номер, номер отдела, фамилия, оклад, дата поступления на работу, процент надбавки, подоходный налог, количество отработанных дней в месяце, количество рабочих дней в месяце, начислено,

<p>удержано. Поиск по номеру отдела, полу, дате поступления, фамилии.</p>
<p>10. Запись о багаже пассажира авиарейса содержит следующие поля: номер рейса, дата и время вылета, пункт назначения, фамилия пассажира, количество мест багажа, суммарный вес багажа. Поиск выполнять по номеру рейса, дате вылета, пункту назначения, весу багажа (превышение максимально допустимого).</p>
<p>11. Одна учетная запись посещения спорткомплекса имеет структуру: фамилия клиента, код и вид спортивного занятия, фамилия тренера, дата и время начала, количество минут, тариф за минуту. Поиск по фамилии клиента и тренера, по виду занятия, по дате начала, по количеству минут (больше или меньше).</p>
<p>12. Одна запись о медикаменте содержит следующие поля: номер аптеки, название лекарства, количество упаковок, имеющееся в наличии в данной аптеке, стоимость одной упаковки, дата поступления в аптеку, срок хранения (в днях). Поиск по номеру аптеки, наименованию препарата, дате поступления.</p>
<p>13. Одна запись журнала содержит поля: код игрушки, название игрушки, тип игрушки, возрастные границы (например, от 10 до 15), цена за единицу, количество в наличии, дата поступления в магазин, поставщик. Поиск по дате поступления, поставщику, возрастным границам</p>
<p>14. Один элемент — автомобиль — представляет собой в базе данных структуру с полями: фамилия владельца, код марки автомобиля, марка автомобиля, требуемая марка бензина, мощность двигателя, объем бака, остаток бензина, объем масла. Дана фиксированная цена литра бензина и заливки масла. Поиск по марке автомобиля, марке бензина, мощности двигателя, фамилии владельца</p>
<p>15. Одна запись в журнале зимней экзаменационной сессии представляет собой структуру с полями: курс, код группы, фамилия студента, номер зачетной книжки, дисциплина, оценка за экзамен по дисциплине. Вычисляются средние баллы по дисциплине, по группе, по курсу. Поиск по курсу, по группе, по номеру зачетной книжки, по фамилии, по оценкам</p>

Практическая работа №2. Разработка приложения с несколькими формами.

Цель практической работы: получение навыка работы разработки приложения с несколькими формами с использованием JSON

Задание:

Разработка калькулятора с добавлением пакета NuGet

Эталон ответа:

```
public double DoOperation(double num1, double num2, string op)
{
    double result = double.NaN;
    writer.WriteStartObject();
    writer.WritePropertyName("Operand1");
    writer.WriteValue(num1);
    writer.WritePropertyName("Operand2");
    writer.WriteValue(num2);
    writer.WritePropertyName("Operation");
    switch (op)
    {
        {
        case "a": result = num1 + num2;
        writer.WriteValue("Add");
        break;
        case "s": result = num1 - num2;
        writer.WriteValue("Subtract");
        break;
        case "m": result = num1 * num2; writer.WriteValue("Multiply");
```

```

break; case "d":
if (num2 != 0)
{
result = num1 / num2;
writer.WriteValue("Divide");
}
break;
default: break;
} writer.WritePropertyName("Result");
writer.WriteValue(result);
writer.WriteEndObject();
return result;
}

```

Варианты:

1. Счет в банке представляет собой структуру с полями: номер счета, код счета, фамилия владельца, сумма на счете, дата открытия счета, годовой процент начисления. Поиск по номеру счета, дате открытия и владельцу.
2. Запись о товаре на складе представляет собой структуру с полями: номер склада, код товара, наименование товара, дата поступления на склад, срок хранения в днях, количество единиц товара, цена за единицу товара. Поиск по номеру склада, коду товара, дате поступления и сроку хранения (просроченные и не просроченные товары).
3. Запись о преподаваемой дисциплине представляется структурой: код дисциплины в учебном плане, наименование дисциплины, фамилия преподавателя, код группы, количество студентов в группе, количество часов лекций, количество часов практики, наличие курсовой работы, вид итогового контроля (зачет или экзамен). Зачет 0,35 ч на одного студента; экзамен — 0,5 ч на студента. Поиск осуществлять по фамилии преподавателя, коду группы, наличию курсовой, виду итогового контроля.
4. Информационная запись о книге, выданной на руки абоненту, представляет собой структуру следующего вида: номер читательского билета, фамилия абонента, дата выдачи, срок возврата (количество дней), автор, название, год издания, издательство, цена. Поиск по полям: номер читательского билета, автор, издательство, дата возврата (просроченные).
5. Информационная запись о файле содержит поля: каталог, имя файла, расширение, дата и время создания, атрибуты «только чтение», «скрытый», «системный», признак удаления, количество выделенных секторов (размер сектора принять равным 512 байт). Поиск выполнять по каталогу, дате создания, по признаку удаления.
6. Разовый платеж за телефонный разговор является структурой с полями: фамилия плательщика, номер телефона, дата разговора, тариф за минуту разговора, скидка (в процентах), время начала разговора, время окончания разговора. Поиск по фамилии, дате разговора, номеру телефона.
7. Модель компьютера характеризуется кодом и названием марки компьютера, типом процессора, частотой работы процессора, объемом оперативной памяти, объемом жесткого диска, объемом памяти видеокарты, стоимостью компьютера в условных единицах и количеством экземпляров, имеющихся в наличии. Поиск по типу процессора, объему ОЗУ, памяти видеокарты и жесткого диска.
8. Список абонентов сети кабельного телевидения состоит из элементов следующей структуры: фамилия, район, адрес, телефон, номер договора, дата заключения договора, оплата установки, абонентская плата ежемесячно, дата последнего платежа. Поиск по фамилии, району, дате заключения договора, дате последнего платежа.
9. Сотрудник представлен структурой Person с полями: табельный номер, номер отдела, фамилия, оклад, дата поступления на работу, процент надбавки, подоходный налог,

количество отработанных дней в месяце, количество рабочих дней в месяце, начислено, удержано. Поиск по номеру отдела, полу, дате поступления, фамилии.
10. Запись о багаже пассажира авиарейса содержит следующие поля: номер рейса, дата и время вылета, пункт назначения, фамилия пассажира, количество мест багажа, суммарный вес багажа. Поиск выполнять по номеру рейса, дате вылета, пункту назначения, весу багажа (превышение максимально допустимого).
11. Одна учетная запись посещения спорткомплекса имеет структуру: фамилия клиента, код и вид спортивного занятия, фамилия тренера, дата и время начала, количество минут, тариф за минуту. Поиск по фамилии клиента и тренера, по виду занятия, по дате начала, по количеству минут (больше или меньше).
12. Одна запись о медикаменте содержит следующие поля: номер аптеки, название лекарства, количество упаковок, имеющееся в наличии в данной аптеке, стоимость одной упаковки, дата поступления в аптеку, срок хранения (в днях). Поиск по номеру аптеки, наименованию препарата, дате поступления.
13. Одна запись журнала содержит поля: код игрушки, название игрушки, тип игрушки, возрастные границы (например, от 10 до 15), цена за единицу, количество в наличии, дата поступления в магазин, поставщик. Поиск по дате поступления, поставщику, возрастным границам
14. Один элемент — автомобиль — представляет собой в базе данных структуру с полями: фамилия владельца, код марки автомобиля, марка автомобиля, требуемая марка бензина, мощность двигателя, объем бака, остаток бензина, объем масла. Дана фиксированная цена литра бензина и заливки масла. Поиск по марке автомобиля, марке бензина, мощности двигателя, фамилии владельца
15. Одна запись в журнале зимней экзаменационной сессии представляет собой структуру с полями: курс, код группы, фамилия студента, номер зачетной книжки, дисциплина, оценка за экзамен по дисциплине. Вычисляются средние баллы по дисциплине, по группе, по курсу. Поиск по курсу, по группе, по номеру зачетной книжки, по фамилии, по оценкам

Практическая работа №3 Разработка приложения с не визуальными компонентами

Цель практической работы: получение навыка разработки приложения с не визуальными компонентами

Задание:

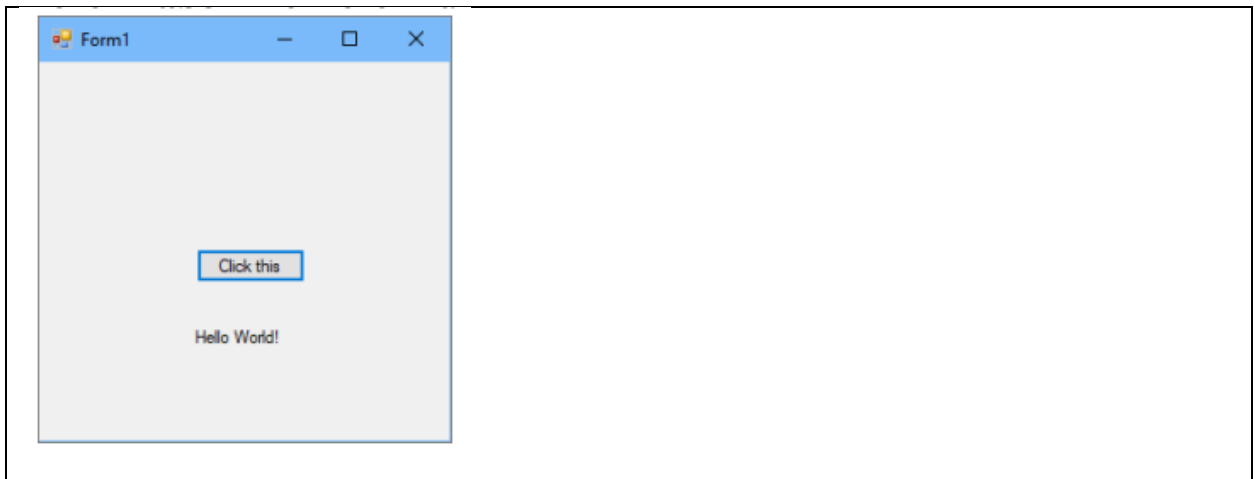
Разработать приложение Windows с использованием компонентов Button, Label

Эталон ответа:

```

1  using System;
2  using System.Windows.Forms;
3
4  namespace HelloWorld
5  {
6      public partial class Form1 : Form
7      {
8          public Form1()
9          {
10             InitializeComponent();
11         }
12
13         private void btnClickThis_Click(object sender, EventArgs e)
14         {
15             lblHelloWorld.Text = "Hello World!";
16         }
17     }
18
19

```

Варианты:

Разработать приложение Windows с использованием компонентов Button, Label вывести на экран:

Вывести свое ФИО и номер зачетной книжки

Практическая работа №4 Разработка игрового приложения

Цель практической работы:

Задание:

Разработать простую мини игру «Угадай число». компьютер загадывает число от 0 до 100, выдаст подсказку – больше ли это число 50 или нет, затем сравнит введенное пользователем число с загаданным.

Эталон ответа:

```
class Program
{
    static void Main(string[] args)
    {
        Console.OutputEncoding = Encoding.GetEncoding(866);
        Console.InputEncoding = Encoding.GetEncoding(866);
        char again = 'y';
        Random rand = new Random();
        while (again == 'y') { int i = rand.Next(100);
        Console.WriteLine("Компьютер загадал число от 0 до 100");
        if (i < 50) Console.WriteLine("Число меньше 50");
        else Console.WriteLine("Число больше или равно 50");
        int x = Convert.ToInt32(Console.ReadLine());
        if (i == x) Console.WriteLine("Поздравляем! Вы победили свой компьютер!");
        else Console.WriteLine("Вы проиграли! Компьютер загадал число {0}", i);
        Console.WriteLine("Попробовать еще? (y = Да, n = Нет)");
        again = Convert.ToChar(Console.ReadLine());
        }
}
```

Варианты:

Разработать простую мини игру «Угадай число». компьютер загадывает число от 0 до 50, выдаст подсказку – больше ли это число 50 или нет, затем сравнит введенное пользователем число с загаданным.

Разработать простую мини игру «Угадай число». компьютер загадывает число от 0 до 40, выдаст подсказку – больше ли это число 50 или нет, затем сравнит введенное

пользователем число с загаданным.

Разработать простую мини игру «Угадай число». компьютер загадывает число от 0 до 20, выдаст подсказку – больше ли это число 50 или нет, затем сравнит введенное пользователем число с загаданным.

Разработать простую мини игру «Угадай число». компьютер загадывает число от 0 до 30, выдаст подсказку – больше ли это число 50 или нет, затем сравнит введенное пользователем число с загаданным.

Разработать простую мини игру «Угадай число». компьютер загадывает число от 0 до 70, выдаст подсказку – больше ли это число 50 или нет, затем сравнит введенное пользователем число с загаданным.

Разработать простую мини игру «Угадай число». компьютер загадывает число от 0 до 80, выдаст подсказку – больше ли это число 50 или нет, затем сравнит введенное пользователем число с загаданным.

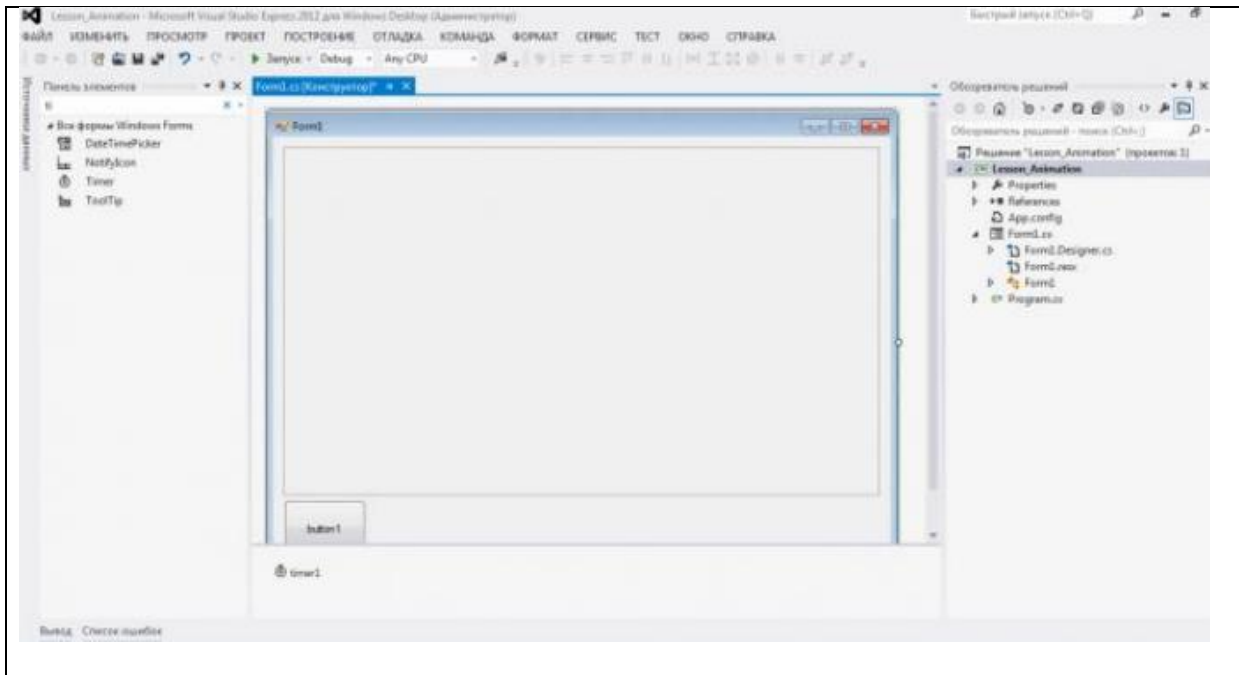
Практическая работа №5 Разработка приложения с анимацией

Цель практической работы: получить навыки разработки приложений с анимацией

Задание:

Создать какую-нибудь примитивную анимацию, при помощи PictureBox и Timer, которая запустится после нажатия на кнопку (Button).

Эталон ответа:



Варианты:

Реализовать программу таким образом что б она описывала одну из геометрических фигур



Практические и лабораторные работы Тема 1.6 Оптимизация и рефакторинг кода

Практическая работа №1. Оптимизация и рефакторинг кода

Цель практической работы: Изучить технологию модифицирования программного кода

Задание: Модифицируйте 2 программы, реализованные на C#

Эталон ответа:

Плохой код

```
def func(a,b):
```

```
c = a + b
```

```
return c
```

Этот код демонстрирует несколько признаков плохого кода, таких как отсутствие отступов, неправильное именование переменных и отсутствие комментариев.

Если у вас есть конкретные вопросы или запросы относительно кода, пожалуйста, уточните, и я постараюсь помочь вам.

Варианты:

```
// Плохой код
```

```
for (int i = 0; i < 10; i++)
```

```
{
```

```
    // Что означает число 10?
```

```
}
```

```
// Хороший код
```

```
const int MAX_ITERATIONS = 10;
```

```
for (int i = 0; i < MAX_ITERATIONS; i++)
```

```
{
```

```
    // Более понятно, сколько итераций будет выполнено
```

```
}
```

```
// Плохой код
```

```
for (int i = 0; i < 10; i++)
```

```
{
```

```
    // Что означает число 10?
```

```
}
```

```
// Хороший код
```

```
const int MAX_ITERATIONS = 10;
```

```
for (int i = 0; i < MAX_ITERATIONS; i++)
```

<pre>{ // Более понятно, сколько итераций будет выполнено }</pre>
<pre>/ Плохой код int result = number1 + number2; Console.WriteLine(result); int result2 = number1 + number2; Console.WriteLine(result2); // Хороший код int result = number1 + number2; Console.WriteLine(result); int result2 = number1 * number2; Console.WriteLine(result2);</pre>
<pre>// Плохой код int x = 5; int y = 10; int z = x + y; // Хороший код int firstNumber = 5; int secondNumber = 10; int sum = firstNumber + secondNumber;</pre>
<pre>// Плохой код try { // Код, который может вызвать исключение } catch { // Пустой блок catch, который не обрабатывает исключение }</pre>
<pre># Плохой код def calculate_sum(lst): sum = 0 for i in range(len(lst)): sum = sum + lst[i] return sum</pre>
<pre># Плохой код def check_prime(num): prime = True for i in range(2, num): if num % i == 0: prime = False return prime</pre>

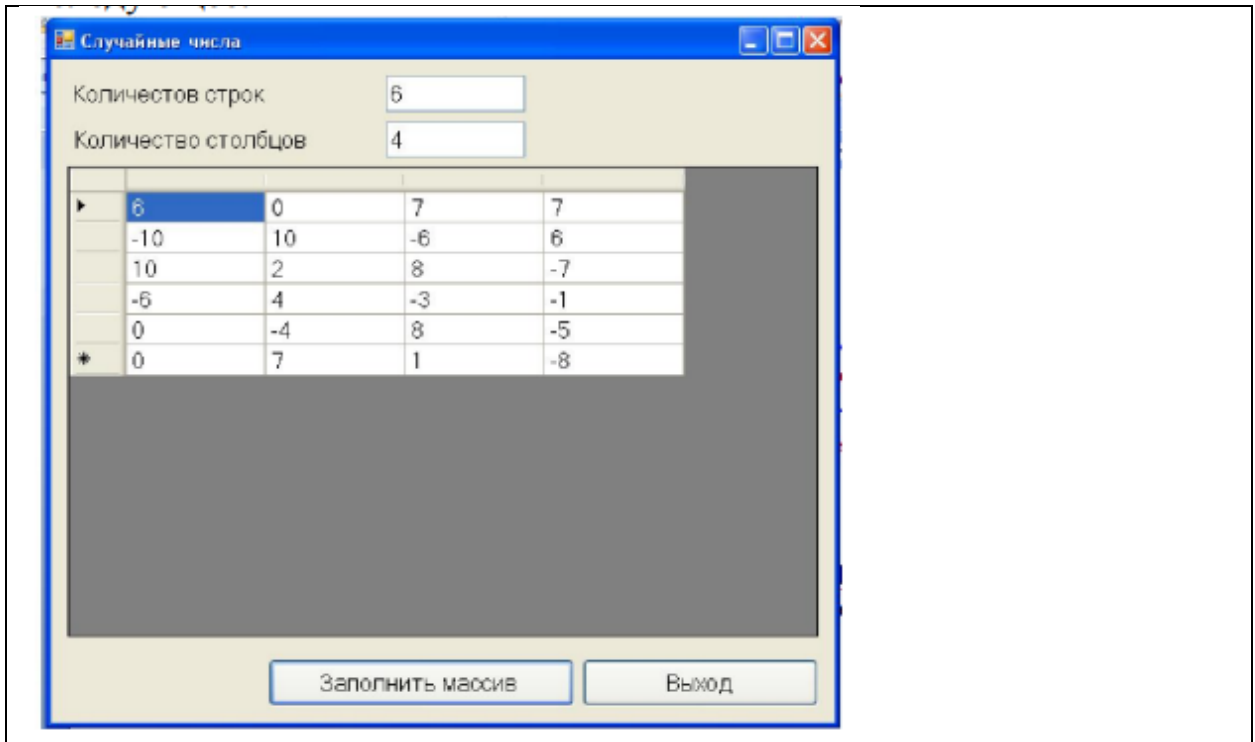
Практические и лабораторные работы Тема 1.7 Разработка пользовательского интерфейса
Практическая работа №1. Разработка интерфейса пользователя

Цель практической работы: Познакомиться с основными элементами управления (виджетами) и приобрести навыки проектирования графического интерфейса пользователя.

Задание:

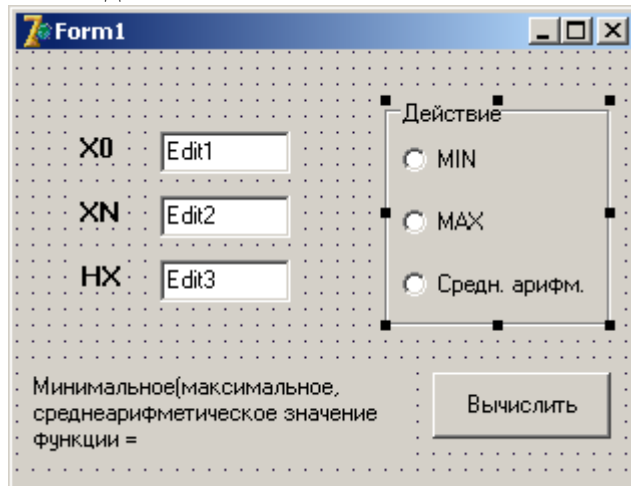
Разработать программу Отображение и ввод массивов в C# и платформе .NET

Эталон ответа:



Варианты:

1) Составить программу табулирования функции в диапазоне x_0 (h_x) x_n и вычислить необходимое значение в соответствии с положением переключателя.



2) Составить программу для вычисления значения функции соответствии с положением переключателя. Функцию для каждого положения переключателя задать самостоятельно.

3) Составить программу для вычисления значения функции в соответствии с положением переключателя.

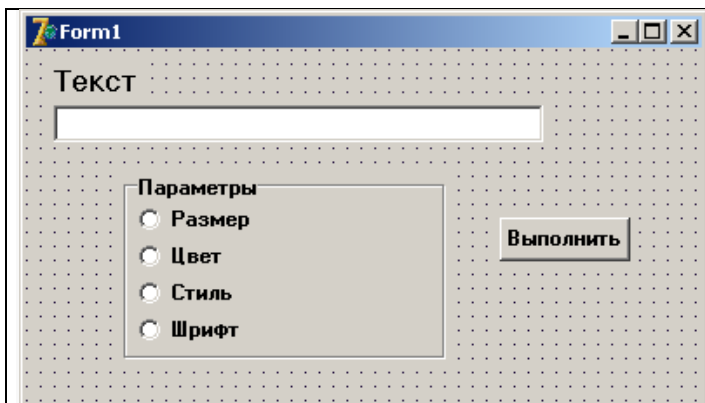
4) Составить программу для вывода на форму фигуры в соответствии с выбранным переключателем. Цвет заливки и цвет линии выбираются с помощью выпадающего списка. Предусмотреть, чтобы пользователь мог ввести в поле Толщина линии только целые числовые данные.

5) Составить программу вычисления в соответствии с положением переключателя. Предусмотреть, чтобы пользователь мог ввести только числовые данные. Фраза ответа должна соответствовать выбранному действию.

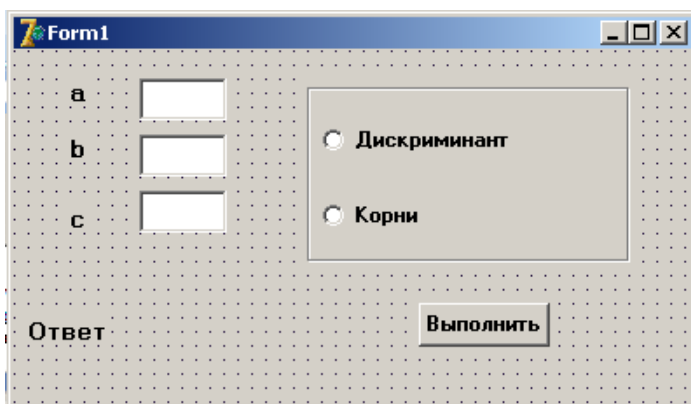
6) Составить программу вычисления стоимости заказа на жалюзи. Предусмотреть, чтобы пользователь мог ввести только числовые данные.

7) Составить программу для определения суммы и произведения цифр введенного числа, а также определения нового числа, полученного путем перестановки цифр введенного числа. Предусмотреть, чтобы пользователь мог ввести только целые числовые данные.

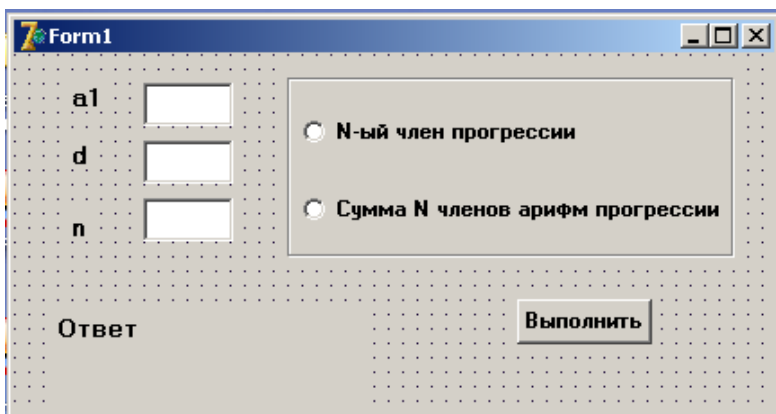
8) Составить программу для изменения внешнего вида введенного текста в соответствии с положением переключателя. Предусмотреть, чтобы пользователь мог ввести только числовые и буквенные данные.



9) Составить программу нахождения дискриминанта и корней квадратного уравнения в соответствии с положением переключателя. Предусмотреть, чтобы пользователь мог ввести только числовые данные.



10) Составить программу нахождения N -го члена арифметической прогрессии и суммы первых N членов арифметической прогрессии. Первый член прогрессии, разность прогрессии и N вводятся с формы. Ответ соответствует выбранному переключателю. Предусмотреть, чтобы пользователь мог ввести только числовые данные.

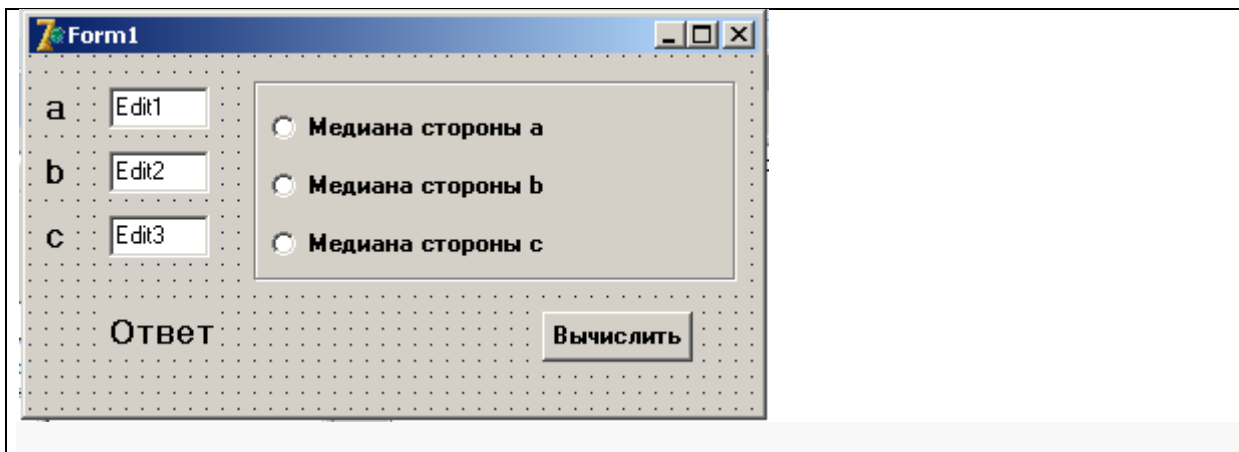


11) Составить программу вычисления степени и корня введенного числа в соответствии с положением переключателя. Предусмотреть, чтобы пользователь мог ввести только числовые данные.

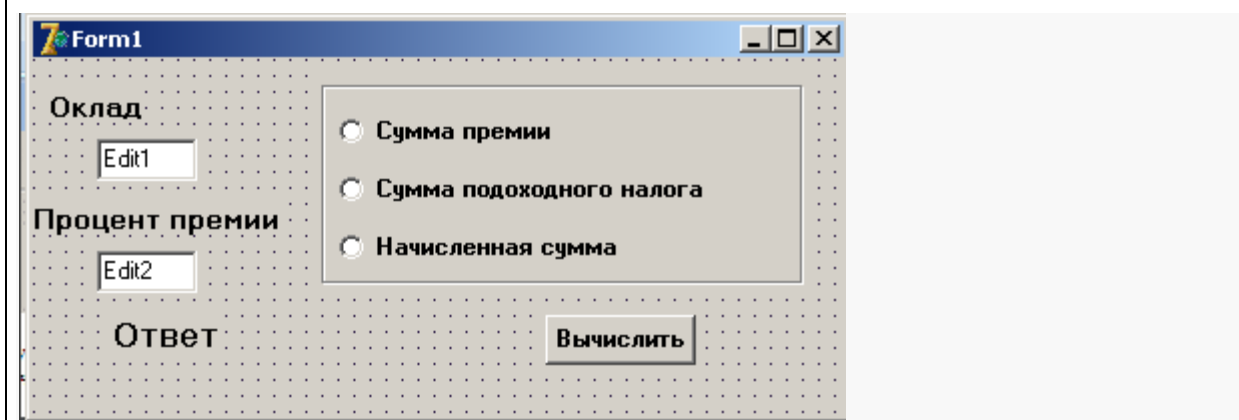
12) Составить программу нахождения высот треугольника, опущенных на стороны a,b,c. Предусмотреть, чтобы пользователь мог ввести только числовые данные.

13) Составить программу нахождения длины диаметра окружности, длины окружности и площади круга, ограниченной данной окружностью. Предусмотреть, чтобы пользователь мог ввести только числовые данные.

14) Составить программу нахождения медиан сторон a,b,c треугольника. Предусмотреть, чтобы пользователь мог ввести только числовые данные.



15) Составить программу вычисления размера премии, размера подоходного налога и итоговой суммы выплаты. Предусмотреть, чтобы пользователь мог ввести только числовые данные.



Практические и лабораторные работы Тема 1.8 Основы ADO.Net

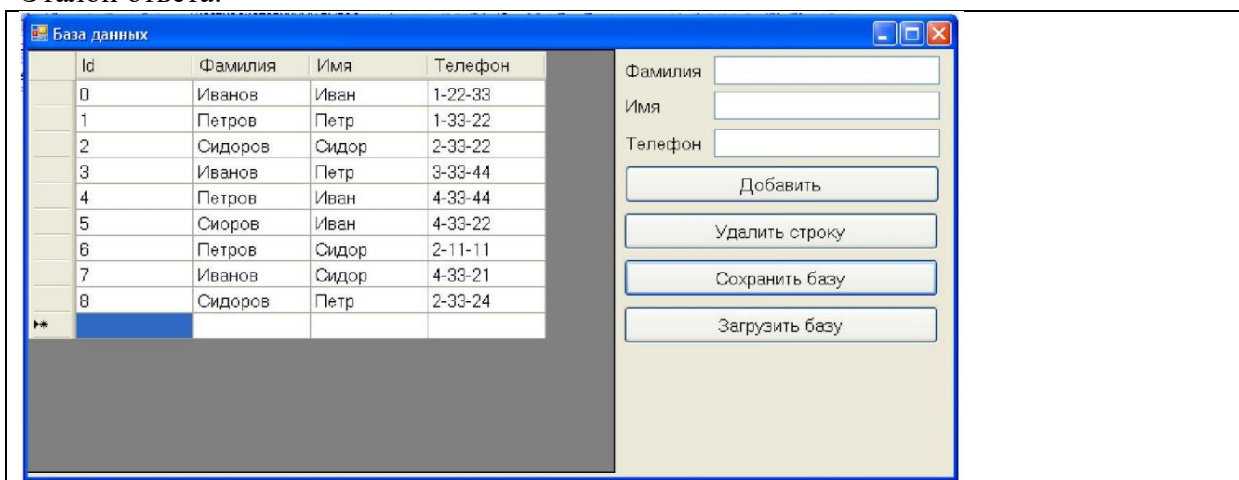
Практическая работа №1. Создание приложения с БД

Цель практической работы: получить навыки разработки приложений баз данных

Задание:

Разработка пользовательского интерфейса для работы с базой данных

Эталон ответа:



Варианты:

Разработка пользовательского интерфейса для работы с базой данных

Система управления книжным магазином с таблицами для книг, авторов, заказов и клиентов.

База данных для учета студентов, преподавателей, курсов и оценок в учебном заведении.

Система управления складом с таблицами для товаров, поставщиков, складских остатков и отгрузок.

База данных для интернет-форума с таблицами для пользователей, тем, сообщений и комментариев.

Система учета банковских транзакций с таблицами для клиентов, счетов, операций и транзакций.

База данных для системы бронирования гостиницы с таблицами для номеров, бронирований, гостей и услуг.

Система управления персоналом компании с таблицами для сотрудников, отделов, проектов и рабочего времени.

База данных для онлайн-магазина с таблицами для товаров, категорий, заказов и покупателей.

Система учета медицинских записей пациентов с таблицами для диагнозов, лечения, врачей и пациентов.

База данных для системы онлайн-обучения с таблицами для курсов, студентов, заданий и преподавателей.

База данных для системы управления проектами с таблицами для задач, сотрудников, времени и статусов проектов.

Система учета заказов в ресторане с таблицами для меню, заказов, клиентов и доставки.

База данных для системы управления отношениями с клиентами (CRM) с таблицами для контактов, сделок, компаний и истории взаимодействий.

Система учета авиабилетов с таблицами для рейсов, пассажиров, багажа и бронирований.

База данных для системы онлайн-резервирования столов в ресторане с таблицами для столов, бронирований, гостей и меню.

Практическая работа №2. Создание запросов к БД

Цель практической работы: получить навыки формирования SQL запросов на добавление, изменение, извлечение и удаление данных на примере созданной согласно варианту базы данных. Изучить основы создания простейших триггеров.

Задание:

Создать запросы на извлечение данных (Требования: запросы должны отражать потребности реальных пользователей, например, найти самую дорогую книгу, самую покупаемую вещь, определить наиболее частых клиентов и т.д.)

Эталон ответа:

```

1. Вывод данных таблицы CATALOGS с присвоением псевдонима
SELECT
ID_CATALOG AS 'Идентификатор категории', -- команда AS изменяет имя столбца в
результатирующей таблице на имя, указанное после этой команды в кавычках
(ID_CATALOG изменяется на 'Идентификатор категории')
NAME AS 'Имя категории'
FROM CATALOGS
2. Добавление данных с помощью SELECT в результирующую таблицу
SELECT NAME, ID_CATALOG, 5, 'COMMENTS' FROM CATALOGS

```

Варианты:

```

1. Вывод записей, удовлетворяющих условию, заданному текстом: вывести все записи,
не содержащие слово процессор
Вывод записей, удовлетворяющих условию, заданному частью текста

```

Сортировка по значению одного из столбцов
Изменение порядка сортировки (по умолчанию, сортировка производится в прямом порядке)
Извлечение первых пяти записей с обратной сортировкой по полю
Вывод числа записей, соответствующих каждому из уникальных значений
Добавим вложенный запрос для получения числовых значений (максимальных, минимальных, средних)

Практическая работа №3. Создание хранимых процедур

Цель практической работы: Научиться создавать, изменять и удалять хранимые процедуры. Применять параметры и использовать выходные параметры в хранимых процедурах.

Задание:

1. Создать хранимую процедуру для выполнения запроса, который выбирает фамилию студента, его адрес и телефон из таблицы Студенты. Вызвать ее на выполнение.
2. Изменить хранимую процедуру так чтобы в запрос включалась еще и группа, в которой учиться студент. Вызвать ее на выполнение

Эталон ответа:

<pre> create proc proc_p1 @Фамилия nvarchar(50) as select * from Туристы where Фамилия=@Фамилия </pre>
--

Варианты:

1. Создать хранимую процедуру для выполнения запроса, который выбирает все записи из таблицы Кадровый состав. Вызвать ее на выполнение.
2. Изменить хранимую процедуру так чтобы в запрос включались только Фамилия, адрес и телефон преподавателя. Вызвать ее на выполнение.
3. Создать хранимую процедуру, которая выводила бы первых пять значений из таблицы Кадровый состав.
4. Создать хранимую процедуру, которая выводила бы все поля таблицы Кадровый состав, содержащие в поле Должность значение «профессор».
5. Создать хранимую процедуру, которая подсчитывала бы все записи таблицы Кадровый состав, содержащие в поле Должность значение «профессор».
6. Создать хранимую процедуру, которая извлекала бы записи из таблицы Кадровый состав с заданным Именем.
7. Создать хранимую процедуру, которую можно использовать для вставки в таблицу Кадровый состав данных о новом преподавателе. Вызвать ее на выполнение с новыми параметрами. Вызвать на исполнение первую хранимую процедуру, чтобы ознакомиться с произошедшими изменениями.
8. Подготовить еще один вариант хранимой процедуры, предназначенной для вставки данных о преподавателях. Обозначить параметр, соответствующий адресу и номеру телефона, как необязательный.
9. Создать хранимую процедуру, которая выводила бы фамилию преподавателя по заданному коду.
10. Создать хранимую процедуру, которая подсчитывала бы количество преподавателей живущих по улице Садовая.

Темы контрольных работ

МДК 01.01 Разработка программных модулей

1. Форма предоставления контрольной работы

Контрольная работа формируется в папку с указанием Ф.И.О. и номера курса (например: *Абдурахманов М.М. П-СП-18*). Папка включает в себя: текстовый файл с программой, презентацию с представлением работы, папка с приложением).

2. Эталон ответа

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace part_B
{
    internal class Program
    {
        public class HeatingSystem
        {
            public int temperature = 75;
            public bool heatingSystemEnabled = false;
            public bool heatingSystemFilledWithWater = false;

            public void enableHeatingSystem()
            {
                heatingSystemEnabled = true;
                Console.WriteLine("Heating system was successfully enabled");
            }

            public void disableHeatingSystem()
            {
                heatingSystemEnabled = false;
                Console.WriteLine("Heating system was disabled");
            }

            public void determineBuildingTemperature()
            {
                Console.WriteLine("Building temperature is " + temperature + " celcius");
            }
            public void fillWaterInHeatingSystem()
            {
                heatingSystemFilledWithWater = true;
                Console.WriteLine("Heating system was filled with water");
            }
            virtual public void drainWaterFromHeatingSystem()
            {
```

```

        heatingSystemFilledWithWater = false;
        Console.WriteLine("Heating system were about to be drained");
    }
}
public class Building : HeatingSystem
{
    public string height = "0";
    public override void drainWaterFromHeatingSystem()
    {
        Console.WriteLine("Heating system was drained");
    }
    public Building(string height)
    {
        this.height = height;
    }
}
static void Main(string[] args)
{
    Building building = new Building("100");
    int userInput = 7;
    Console.WriteLine("Welcome!");
    while (userInput != 0)
    {
        Console.WriteLine("To enable heating system write: 1");
        Console.WriteLine("To disable heating system write: 2");
        Console.WriteLine("To determine buildings temperature write: 3");
        Console.WriteLine("To fill the heating system with water write: 4");
        Console.WriteLine("To drain the heating system write: 5");
        Console.WriteLine("To exit the app write anything else");
        userInput = Convert.ToInt32(Console.ReadLine());
        switch (userInput)
        {
            case 1:
                building.enableHeatingSystem();
                Console.WriteLine("-----");
                break;
            case 2:
                building.disableHeatingSystem();
                Console.WriteLine("-----");
                break;
            case 3:
                building.determineBuildingTemperature();
                Console.WriteLine("-----");
                break;
            case 4:
                building.fillWaterInHeatingSystem();
                Console.WriteLine("-----");
                break;
            case 5:
                building.drainWaterFromHeatingSystem();
                Console.WriteLine("-----");
                break;
        }
    }
}

```

```
default: break;  
}      }      }  } }
```

3. Варианты заданий:

<p>1 Создать класс квадрат, члены класса – длина стороны. Предусмотреть в классе методы вычисления и вывода сведений о фигуре – диагональ, периметр, площадь. Создать производный класс – правильная квадратная призма с высотой H, добавить в класс метод определения объема фигуры, перегрузить методы расчета площади и вывода сведений о фигуре. Написать программу, демонстрирующую работу с этими классами: дано N квадратов и M призм, найти квадрат с максимальной площадью и призму с максимальной диагональю.</p>
<p>2 Создать класс треугольник, члены класса – длины трех сторон. Предусмотреть в классе методы проверки существования треугольника, вычисления и вывода сведений о фигуре – длины сторон, углы, периметр, площадь. Создать производный класс – равносторонний треугольник, перегрузить в классе проверку, является ли треугольник равносторонним, и метод вывода сведений о фигуре. Написать программу, демонстрирующую работу с классом: дано K треугольников и L равносторонних треугольников, найти среднюю площадь для K треугольников и наибольший равносторонний треугольник</p>
<p>3 Создать класс окружность, член класса – радиус R. Предусмотреть в классе методы вычисления и вывода сведений о фигуре – площади, длины окружности. Создать производный класс – круглый прямой цилиндр с высотой h, добавить в класс метод определения объема фигуры, перегрузить методы расчета площади и вывода сведений о фигуре. Написать программу, демонстрирующую работу с классом: дано N окружностей и M цилиндров, найти окружность максимальной площади и средний объем цилиндров.</p>
<p>4 Создать класс квадрат, члены класса – длина стороны. Предусмотреть в классе методы вычисления и вывода сведений о фигуре – диагональ, периметр, площадь. Создать производный класс – правильная пирамида с апофемой h, добавить в класс метод определения объема фигуры, перегрузить методы расчета площади и вывода сведений о фигуре. Написать программу, демонстрирующую работу с классом: дано N квадратов и M пирамид, найти квадрат с минимальной площадью и количество пирамид с высотой более числа a (a вводить).</p>
<p>5 Создать класс четырехугольник, члены класса – координаты четырех точек. Предусмотреть в классе методы проверки существования четырехугольника вычисления и вывода сведений о фигуре – длины сторон, диагоналей, периметр, площадь. Создать производный класс – параллелограмм, предусмотреть в классе проверку, является ли фигура параллелограммом. Написать программу, демонстрирующую работу с классом: дано N четырехугольников и M параллелограммов, найти среднюю площадь N четырехугольников и параллелограммы наименьшей и наибольшей площади</p>
<p>6 Создать класс треугольник, члены класса – координаты трех точек. Предусмотреть в классе методы проверки существования треугольника, вычисления и вывода сведений о фигуре – длины сторон, углы, периметр, площадь. Создать производный класс – равносторонний треугольник, предусмотреть в классе проверку, является ли треугольник равносторонним. Написать программу, демонстрирующую работу с классом: дано N треугольников и M равносторонних треугольников, вывести номера одинаковых треугольников и равносторонний треугольник с наименьшей медианой.</p>
<p>7 Создать класс прямоугольник, члены класса – длины сторон a и b. Предусмотреть в классе методы вычисления и вывода сведений о фигуре – длины сторон, диагоналей, периметр, площадь. Создать производный класс – параллелепипед с высотой c, добавить в класс метод определения объема фигуры, перегрузить методы расчета</p>

<p>площади и вывода сведений о фигуре. Написать программу, демонстрирующую работу с классом: дано N прямоугольников и M параллелепипедов, найти количество прямоугольников, у которых площадь больше средней площади прямоугольников и количество кубов (все ребра равны).</p>
<p>8 Создать класс окружность, член класса – радиус R. Предусмотреть в классе методы вычисления и вывода сведений о фигуре – площади, длины окружности. Создать производный класс – конус с высотой h, добавить в класс метод определения объема фигуры, перегрузить методы расчета площади и вывода сведений о фигуре. Написать программу, демонстрирующую работу с классом: дано N окружностей и M конусов, найти количество окружностей, у которых площадь меньше средней площади всех окружностей, и наибольший по объему конус.</p>
<p>9 Создать класс четырехугольник, члены класса – координаты четырех точек. Предусмотреть в классе методы вычисления и вывода сведений о фигуре длины сторон, диагоналей, периметр, площадь. Создать производный класс – равнобокая трапеция, предусмотреть в классе проверку, является ли фигура равнобокой трапецией. Написать программу, демонстрирующую работу с классом: дано N четырехугольников и M трапеций, найти максимальную площадь четырехугольников и количество четырехугольников, имеющих максимальную площадь, и трапецию с наименьшей диагональю.</p>
<p>10 Создать класс равносторонний треугольник, член класса – длина стороны. Предусмотреть в классе методы вычисления и вывода сведений о фигуре – периметр, площадь. Создать производный класс – правильная треугольная призма с высотой H, добавить в класс метод определения объема фигуры, перегрузить методы расчета площади и вывода сведений о фигуре. Написать программу, демонстрирующую работу с классом: дано N треугольников и M призм. Найти количество треугольников, у которых площадь меньше средней площади треугольников, и призму с наибольшим объемом.</p>
<p>11 Создать класс треугольник, члены класса – длины трех сторон. Предусмотреть в классе методы проверки существования треугольника, вычисления и вывода сведений о фигуре – длины сторон, углы, периметр, площадь. Создать производный класс – прямоугольный треугольник, предусмотреть в классе проверку, является ли треугольник прямоугольным. Написать программу, демонстрирующую работу с классом: дано N треугольников и M прямоугольных треугольников, найти треугольник с максимальной площадью и прямоугольный треугольник с наименьшей гипотенузой.</p>
<p>12 Создать класс четырехугольник, члены класса – координаты четырех точек. Предусмотреть в классе методы вычисления и вывода сведений о фигуре – длины сторон, диагоналей, периметр, площадь. Создать производный класс – квадрат, предусмотреть в классе проверку, является ли фигура квадратом. Написать программу, демонстрирующую работу с классом: дано N четырехугольников и M квадратов; найти четырехугольники с минимальной и максимальной площадью и номера одинаковых квадратов.</p>
<p>13 Создать класс треугольник, члены класса – длины трех сторон. Предусмотреть в классе методы проверки существования треугольника, вычисления и вывода сведений о фигуре – длины сторон, углы, периметр, площадь. Создать производный класс – равнобедренный треугольник, предусмотреть в классе проверку, является ли треугольник равнобедренным. Написать программу, демонстрирующую работу с классом: дано N треугольников и M равнобедренных треугольников, найти среднюю площадь для N треугольников и равнобедренный треугольник с наименьшей площадью.</p>
<p>14 Создать класс квадрат, член класса – длина стороны. Предусмотреть в классе методы вычисления и вывода сведений о фигуре – периметр, площадь, диагональ. Создать производный класс – куб, добавить в класс метод определения объема фигуры, перегрузить методы расчета площади и вывода сведений о фигуре. Написать</p>

программу, демонстрирующую работу с классом: дано N1 квадратов и N2 кубов. Найти среднюю площадь квадратов и количество кубов с наибольшей площадью.

15 Создать класс четырехугольник, члены класса – координаты четырех точек. Предусмотреть в классе методы вычисления и вывода сведений о фигуре – длины сторон, диагоналей, периметр, площадь. Создать производный класс – ромб, предусмотреть в классе проверку, является ли фигура ромбом. Написать программу, демонстрирующую работу с этими классами: дано N четырехугольников и M ромбов; найти четырехугольник с минимальным периметром и среднюю площадь ромбов.

Критерии оценки:

отметка «5»: Задание выполнено в полном объеме с соблюдением необходимой последовательности. Студент работал полностью самостоятельно.

отметка «4»: Практическое задание выполнено студентом в полном объеме и самостоятельно. Допускается отклонение от необходимой последовательности выполнения, не влияющее на правильность конечного результата. Допускаются неточности и небрежность в оформлении результатов задания.

отметка «3»: Практическое задание выполнено и оформлено студентом с помощью преподавателя или хорошо подготовленных и уже выполнивших на «отлично» данную работу студентов. На выполнение задания затрачено много времени.

отметка «2»: Выставляется в том случае, когда студент оказался неподготовленным к выполнению задания. Полученные результаты не позволяют сделать правильных выводов и полностью расходятся с поставленной целью. Обнаружено плохое знание теоретического материала и отсутствие необходимых умений. Руководство и помощь со стороны преподавателя неэффективны из-за плохой подготовки студента.

Составители:

_____ Ж.А. Кадышева

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ ДАГЕСТАН

Частное профессиональное образовательное учреждение
«Республиканский полипрофессиональный колледж»

Вопросы к экзамену

МДК 01.01 Разработка программных модулей

1. Понятие ЖЦ ПО. Этапы ЖЦ ПО
2. Технология структурного программирования
3. Инструментальные средства оформления и документирования алгоритмов программ
4. Оценка сложности алгоритма: классификация, классы алгоритмов, неразрешимые задачи
5. Основные принципы объектно-ориентированного программирования. Классы: основные понятия
6. Перегрузка методов.
7. Операции класса.
8. Иерархия классов.
9. Синтаксис интерфейсов.
10. Интерфейсы и наследование.
11. Структуры.
12. Делегаты.
13. Регулярные выражения
14. Коллекции. Параметризованные классы.

15. Указатели
16. Операции со списками
17. Назначение и виды паттернов.
18. Основные шаблоны.
19. Порождающие шаблоны.
20. Структурные шаблоны.
21. Поведенческие шаблоны.
22. Событийно-управляемое программирование
23. Элементы управления. Диалоговые окна. Обработчики событий.
24. Введение в графику
25. Методы оптимизации программного кода.
26. Цели и методы рефакторинга.
27. Правила разработки интерфейсов пользователя
28. Работа с базами данных
29. Доступ к данным
30. Создание таблицы, работа с записями.
31. Способы создания команд

Задачи:

Задача 1

Создать библиотеку классов для моделей. На основе задания по варианту, определить классы, необходимые для его реализации, Какие у них будут поля, свойства, методы. Реализовать управление взаимодействием между объектами через интерфейс пользователя

Название классов, свойств классов должно соответствовать логике вашего задания по варианту.

Кондитерская. В качестве компонентов выступают различные виды шоколада и наполнители, типа орехов, изюма и т.п.

Задача 2

Создать библиотеку классов для моделей. На основе задания по варианту, определить классы, необходимые для его реализации, Какие у них будут поля, свойства, методы. Реализовать управление взаимодействием между объектами через интерфейс пользователя

Название классов, свойств классов должно соответствовать логике вашего задания по варианту.

Швейная фабрика. В качестве компонентов выступают различные заготовки для штор, покрывал и т.п.

Задача 3

Создать библиотеку классов для моделей. На основе задания по варианту, определить классы, необходимые для его реализации, Какие у них будут поля, свойства, методы. Реализовать управление взаимодействием между объектами через интерфейс пользователя

Название классов, свойств классов должно соответствовать логике вашего задания по варианту.

Столярная мастерская. В качестве компонентов выступают различные деревянные заготовки. Изделия – деревянные игрушки, утварь и т.п.

Задача 4

Создать библиотеку классов для моделей. На основе задания по варианту, определить классы, необходимые для его реализации, Какие у них будут поля, свойства, методы. Реализовать управление взаимодействием между объектами через интерфейс пользователя

Название классов, свойств классов должно соответствовать логике вашего задания по варианту.

Лавка с мороженым. В качестве компонентов выступают различные виды мороженого и добавки (орехи, шоколад и т.п.)

Задача 5

Создать библиотеку классов для моделей. На основе задания по варианту, определить классы, необходимые для его реализации, Какие у них будут поля, свойства, методы. Реализовать управление взаимодействием между объектами через интерфейс пользователя

Название классов, свойств классов должно соответствовать логике вашего задания по варианту.

Ремонт сантехники. В качестве компонентов выступают различные трубы, прокладки, смесители т.п. Изделия – замены смесителей, труб и т.п.

Задача 6

Создать библиотеку классов для моделей. На основе задания по варианту, определить классы, необходимые для его реализации, Какие у них будут поля, свойства, методы. Реализовать управление взаимодействием между объектами через интерфейс пользователя

Название классов, свойств классов должно соответствовать логике вашего задания по варианту.

Заказы еды. В качестве компонентов выступают различные блюда. Изделия – это наборы блюд (типа обеденный набор, или утренний набор, или набор для пикника).

Задача 7

Создать библиотеку классов для моделей. На основе задания по варианту, определить классы, необходимые для его реализации, Какие у них будут поля, свойства, методы. Реализовать управление взаимодействием между объектами через интерфейс пользователя

Название классов, свойств классов должно соответствовать логике вашего задания по варианту.

Система безопасности. В качестве компонентов выступают различные камеры, датчики и т.п. Изделия – базовые комплектации охраны, продвинутые, для предприятий, для частных и т.п.

Задача 8

Создать библиотеку классов для моделей. На основе задания по варианту, определить классы, необходимые для его реализации, Какие у них будут поля, свойства, методы. Реализовать управление взаимодействием между объектами через интерфейс пользователя

Название классов, свойств классов должно соответствовать логике вашего задания по варианту.

Магазин подарков. В качестве компонентов выступают различные упаковочные материалы, ленты и подарки. Изделия – подарочные наборы.

Задача 9

Создать библиотеку классов для моделей. На основе задания по варианту, определить классы, необходимые для его реализации, Какие у них будут поля, свойства, методы. Реализовать управление взаимодействием между объектами через интерфейс пользователя

Название классов, свойств классов должно соответствовать логике вашего задания по варианту.

Авиастроительный завод. В качестве компонентов выступают различные части для сборки самолета (двигатели, крылья, фюзеляж и т.п.).

Задача 10

Создать библиотеку классов для моделей. На основе задания по варианту, определить классы, необходимые для его реализации, Какие у них будут поля, свойства, методы. Реализовать управление взаимодействием между объектами через интерфейс пользователя

Название классов, свойств классов должно соответствовать логике вашего задания по варианту.

Ювелирная лавка. В качестве компонентов выступают различные драгоценные камни и металлы.

Задача 11

Создать библиотеку классов для моделей. На основе задания по варианту, определить классы, необходимые для его реализации, Какие у них будут поля, свойства, методы. Реализовать управление взаимодействием между объектами через интерфейс пользователя

Название классов, свойств классов должно соответствовать логике вашего задания по варианту.

Туристическая фирма. В качестве компонентов выступают различные условия поездки (отель проживания, туры в рамках поездок). Изделия – туристические путевки.

Задача 12

Создать библиотеку классов для моделей. На основе задания по варианту, определить классы, необходимые для его реализации, Какие у них будут поля, свойства, методы. Реализовать управление взаимодействием между объектами через интерфейс пользователя

Название классов, свойств классов должно соответствовать логике вашего задания по варианту.

Юридическая фирма. В качестве компонентов выступают различные бланки для документов. Изделия – пакеты документов, например, для страховки или завещания.

Задача 13

Создать библиотеку классов для моделей. На основе задания по варианту, определить классы, необходимые для его реализации, Какие у них будут поля, свойства, методы. Реализовать управление взаимодействием между объектами через интерфейс пользователя

Название классов, свойств классов должно соответствовать логике вашего задания по варианту.

Автомобильный завод. В качестве компонентов выступают различные части для сборки автомобилей (кузов, двигатель, стекла и т.п.).

Задача 14

Создать библиотеку классов для моделей. На основе задания по варианту, определить классы, необходимые для его реализации, Какие у них будут поля, свойства, методы.

Реализовать управление взаимодействием между объектами через интерфейс пользователя

Название классов, свойств классов должно соответствовать логике вашего задания по варианту.

Типография. В качестве компонентов выступают различные типы бумаг, тонер или чернила и т.п. Изделия – печатная продукция (листовки, брошюры, книги).

Задача 15

Создать библиотеку классов для моделей. На основе задания по варианту, определить классы, необходимые для его реализации, Какие у них будут поля, свойства, методы. Реализовать управление взаимодействием между объектами через интерфейс пользователя

Название классов, свойств классов должно соответствовать логике вашего задания по варианту.

Закусочная. В качестве компонентов выступают различные продукты для закусок (колбаса, сыр, хлеб и т.п.). Изделия – различные закуски.

Задача 16

Создать библиотеку классов для моделей. На основе задания по варианту, определить классы, необходимые для его реализации, Какие у них будут поля, свойства, методы. Реализовать управление взаимодействием между объектами через интерфейс пользователя

Название классов, свойств классов должно соответствовать логике вашего задания по варианту.

Пиццерия. В качестве компонентов выступают различные ингредиенты для пицц (тесто, соусы, паста и т.д.).

Критерии оценки:

оценки «хорошо» заслуживает студент, обнаруживший полные знания учебно-программного материала, успешно выполняющий предусмотренные в программе задания, усвоивший основную литературу, рекомендованную в программе. Как правило, оценка «хорошо» выставляется студентам, показавшим систематический характер знаний по дисциплине и способным к их самостоятельному пополнению и обновлению в ходе дальнейшей учебной работы и профессиональной деятельности;

оценки «удовлетворительно» заслуживает студент, обнаруживший знание учебно-программного материала в объеме, необходимом для дальнейшей учебы и предстоящей работы по профессии, справляющийся с выполнением заданий, предусмотренных программой, знакомый с основной литературой, рекомендованной программой. Как правило, оценка «удовлетворительно» выставляется студентам, допустившим погрешность в ответе на экзамене и при выполнении экзаменационных заданий, но обладающим необходимыми знаниями для их устранения под руководством преподавателя;

оценка «неудовлетворительно» выставляется студенту, обнаружившему пробелы в знаниях основного учебно-программного материала, допустившему принципиальные ошибки в выполнении предусмотренных программой заданий. Как правило, оценка «неудовлетворительно» ставится студентам, которые не могут продолжить обучение или приступить к профессиональной деятельности по окончании вуза без дополнительных занятий по соответствующей дисциплине».

Составители:

_____ Ж.А. Кадышева

Эталон ответа

Понятие ЖЦ ПО. Этапы ЖЦ ПО

ЖЦ ПО означает "Жизненный Цикл Программного Обеспечения". Этот термин относится к процессу разработки и поддержки программного обеспечения от начала до конца.

Этапы ЖЦ ПО обычно включают:

Планирование: определение требований, составление плана разработки.

Анализ: изучение требований пользователей, определение функциональности ПО.

Дизайн: разработка архитектуры, интерфейса и структуры ПО.

Разработка: создание кода и тестирование ПО.

Тестирование: проверка на соответствие требованиям и исправление ошибок.

Внедрение: установка и запуск ПО.

Сопровождение: поддержка, обновления и устранение ошибок после выпуска.

Каждый этап ЖЦ ПО важен для создания качественного программного обеспечения.

Технология структурного программирования

Структурное программирование - это методология программирования, которая фокусируется на использовании структурированных методов для разработки программ. Она подчеркивает использование последовательных блоков кода с четко определенными входами и выходами, избегая использования операторов безусловного перехода, таких как GOTO. Эта методология способствует созданию более понятного, поддерживаемого и эффективного кода.

Структурное программирование - это методология программирования, которая подчеркивает использование структурированных подходов к разработке программного обеспечения. Она обычно включает в себя использование последовательных, условных и циклических конструкций для создания четко организованных и легко понятных программ. Структурное программирование способствует улучшению читаемости, тестируемости и поддерживаемости кода.

Инструментальные средства оформления и документирования алгоритмов программ

Инструментальные средства оформления и документирования алгоритмов программ могут включать в себя различные инструменты и технологии, такие как текстовые редакторы, интегрированные среды разработки (IDE), языки разметки (например, Markdown, LaTeX), графические инструменты для создания диаграмм (например, UML-диаграммы), системы контроля версий (например, Git), а также специализированные программные средства для документирования алгоритмов и кода (например, Doxygen, Javadoc). Эти средства помогают разработчикам создавать понятную документацию и оформлять алгоритмы программ в удобном и понятном виде.

Оценка сложности алгоритма: классификация, классы алгоритмов, неразрешимые задачи

Оценка сложности алгоритма - это процесс определения количества ресурсов (таких как время и память), необходимых для выполнения алгоритма в зависимости от размера входных данных. Она включает в себя классификацию алгоритмов и изучение неразрешимых задач.

Классификация алгоритмов может включать в себя разделение алгоритмов на классы в зависимости от их сложности и производительности. Например, алгоритмы могут быть классифицированы как константной сложности, логарифмической сложности, линейной сложности, квадратичной сложности и т. д., в зависимости от того, как их производительность изменяется относительно размера входных данных.

Неразрешимые задачи - это задачи, для которых не существует алгоритма, способного решить их за конечное время. Такие задачи могут быть формально описаны, но не могут быть решены с использованием алгоритма. Примером неразрешимой задачи является

проблема останова в теории вычислимости.

Изучение оценки сложности алгоритмов и неразрешимых задач помогает разработчикам понять производительность и возможности алгоритмов, а также определить границы их применимости.

Основные принципы объектно-ориентированного программирования. Классы: основные понятия

Основные принципы объектно-ориентированного программирования (ООП) включают в себя:

Инкапсуляция: Скрытие внутренних деталей объекта и предоставление доступа к ним только через установленные интерфейсы. Это позволяет изолировать данные и методы объекта от внешнего мира, обеспечивая их безопасность и целостность.

Наследование: Возможность создания новых классов на основе существующих классов, путем наследования и расширения их функциональности. Это способствует повторному использованию кода и созданию иерархий классов.

Полиморфизм: Возможность объектов различных классов реагировать на одинаковые сообщения или методы по-разному. Полиморфизм позволяет использовать общие интерфейсы для работы с различными типами объектов.

Классы в объектно-ориентированном программировании представляют собой шаблоны, описывающие состояние и поведение объектов. Они включают в себя следующие основные понятия:

Атрибуты (поля): Переменные, хранящие состояние объекта.

Методы: Функции, определяющие поведение объекта.

Конструкторы: Специальные методы, вызываемые при создании нового объекта и инициализирующие его состояние.

Деструкторы: Специальные методы, вызываемые при уничтожении объекта и освобождении ресурсов.

Классы позволяют создавать объекты, инкапсулируя их состояние и поведение, и обеспечивают основу для реализации принципов ООП.

Перегрузка методов.

В C# перегрузка методов позволяет создавать несколько методов с одинаковым именем в одном классе, но с различными параметрами. Компилятор различает эти методы по количеству или типу параметров.

Пример перегрузки методов на C#:

```
csharp
using System;
public class MyClass
{
    public void DoSomething(int x)
    {
        // Логика для обработки целочисленного аргумента
    }
    public void DoSomething(double x)
    { // Логика для обработки вещественного аргумента }}

```

В этом примере класс MyClass содержит два метода с одинаковым именем DoSomething, но с различными типами параметров (int и double). При вызове метода DoSomething компилятор определяет, какой именно метод следует вызывать на основе переданных аргументов.

При использовании перегрузки методов в C# следует учитывать, что различие между перегруженными методами должно быть таким, чтобы компилятор мог однозначно определить, какой метод вызывать в зависимости от переданных аргументов.

Операции класса

В C# операции класса могут включать в себя перегрузку операторов, что позволяет

определять специальное поведение для встроенных операторов языка. Например, вы можете перегрузить операторы сложения (+), вычитания (-) и другие, чтобы определить, как объекты вашего класса должны вести себя при выполнении этих операций.

Пример перегрузки оператора сложения в C#:

```
csharp
public class ComplexNumber
{
    public int Real { get; set; }
    public int Imaginary { get; set; }
    public static ComplexNumber operator +(ComplexNumber c1, ComplexNumber c2)
    {
        return new ComplexNumber
        { Real = c1.Real + c2.Real,
          Imaginary = c1.Imaginary + c2.Imaginary };
    }
}
```

В этом примере оператор "+" перегружен для класса ComplexNumber, чтобы позволить сложение двух комплексных чисел.

Кроме того, операции класса в C# могут включать в себя методы для изменения состояния объекта, доступ к его свойствам и другие пользовательские операции, определенные внутри класса.

Иерархия классов

Иерархия классов в C# представляет собой организацию классов в иерархическую структуру, где классы могут наследовать свойства и методы от других классов. В C# один класс может быть производным от другого класса, называемого базовым классом. Базовый класс передает свои члены производному классу, что позволяет использовать наследование для повторного использования кода и создания иерархий классов.

Пример иерархии классов в C#:

```
csharp
public class Animal
{
    public void Eat()
    {
        Console.WriteLine("The animal is eating.");
    }
}
public class Dog : Animal
{
    public void Bark()
    {
        Console.WriteLine("The dog is barking.");
    }
}
```

В этом примере класс Dog наследует от класса Animal, что позволяет ему использовать метод Eat, определенный в базовом классе. При этом класс Dog также имеет свой собственный метод Bark.

Использование иерархии классов позволяет организовать код более логично, создавая отношения "является" между классами (например, "собака является животным")

Иерархия классов

В C# иерархия классов обычно строится с использованием ключевого слова "class" для создания новых классов. Классы могут быть унаследованы от других классов с помощью ключевого слова "class" после имени базового класса. Например:

```
csharp
// Базовый класс
public class Animal
```



```

{
    public void Eat()
    {
        // Реализация метода
    }
}
// Производный класс, наследуется от Animal
public class Dog : Animal
{
    public void Bark()
    {
        // Реализация метода
    }
}
// Производный класс, наследуется от Animal
public class Cat : Animal
{
    public void Meow()
    {
        // Реализация метода
    }
}

```

В этом примере классы Dog и Cat наследуются от базового класса Animal. Таким образом, в C# создается иерархия классов, позволяющая использовать наследование для повторного использования кода и создания отношений между классами.

Синтаксис интерфейсов.

В C# интерфейсы определяются с использованием ключевого слова "interface" и содержат сигнатуры методов, свойств, событий или индексов без их реализации. Вот пример синтаксиса интерфейса в C#:

```

csharp
// Определение интерфейса
public interface IShape
{
    // Сигнатура метода
    double CalculateArea();

    // Сигнатура свойства
    string Name { get; set; }
}
// Класс, реализующий интерфейс
public class Circle : IShape
{
    // Реализация метода из интерфейса
    public double CalculateArea()
    {
        // Расчет площади круга
        return Math.PI * radius * radius;
    }
    // Реализация свойства из интерфейса
    public string Name { get; set; }
}

```

В этом примере интерфейс IShape определяет метод CalculateArea и свойство Name без

их реализации. Затем класс Circle реализует интерфейс IShape, предоставляя конкретную реализацию метода и свойства. Интерфейсы в C# позволяют создавать абстрактные контракты, которые классы могут реализовывать, обеспечивая гибкость и повторное использование кода.

Интерфейсы и наследование

В C# интерфейсы предоставляют механизм для реализации множественного наследования, поскольку класс может реализовывать несколько интерфейсов. Это позволяет классу обеспечивать реализацию методов, свойств и событий, определенных в каждом интерфейсе.

Пример:

```
csharp
// Определение интерфейса
public interface IShape
{
    double CalculateArea();
}

// Определение второго интерфейса
public interface IDrawable
{
    void Draw();
}

// Класс, реализующий интерфейсы
public class Circle : IShape, IDrawable
{
    public double CalculateArea()
    {
        // Расчет площади круга
        return Math.PI * radius * radius;
    }

    public void Draw()
    {
        // Реализация метода Draw для отображения круга
    }
}
```

В этом примере класс Circle реализует два интерфейса: IShape и IDrawable. Это позволяет классу Circle предоставлять реализацию методов CalculateArea и Draw, определенных в каждом интерфейсе. Таким образом, интерфейсы в C# обеспечивают гибкость при определении поведения классов и позволяют им реализовывать множество контрактов.

Структуры

В C# структуры (structures) являются типом значений, похожим на классы, но существенно отличающимся по некоторым аспектам. Вот пример синтаксиса определения структуры в C#:

```
csharp
// Определение структуры
public struct Point
{
    public int X;
    public int Y;
}
```

```
// Конструктор
public Point(int x, int y)
{
    X = x;
    Y = y;
}
// Метод
public void Move(int deltaX, int deltaY)
{
    X += deltaX;
    Y += deltaY;
}}
```

В данном примере определяется структура Point с двумя полями X и Y, конструктором и методом Move. Структуры в C# могут содержать поля, методы, свойства, индексы, операторы и события, но не могут содержать явно объявленные параметризованные конструкторы без параметров или деструкторы. Структуры являются типами значений, что означает, что они хранят свои данные непосредственно, а не по ссылке, как классы. Они обычно используются для представления простых объектов, таких как координаты, размеры и другие небольшие объекты данных.

Делегаты

В C# делегаты (delegates) представляют собой тип, который представляет ссылку на метод с определенной сигнатурой. Они предоставляют гибкий способ обращения к методам, подобно указателям на функции в языках C и C++.

Вот пример синтаксиса объявления делегата в C#:

```
csharp
```

```
// Объявление делегата
```

```
public delegate void MyDelegate(string message);
```

```
// Использование делегата
```

```
public class MyClass
```

```
{
```

```
    public void MethodA(string message)
```

```
    {
```

```
        Console.WriteLine("MethodA: " + message);
```

```
    }
```

```
    public void MethodB(string message)
```

```
    {
```

```
        Console.WriteLine("MethodB: " + message);
```

```
    }
```

```
}
```

```
public class Program
```

```
{
```

```
    static void Main()
```

```
    {
```

```
        MyClass obj = new MyClass();
```

```
        MyDelegate del = obj.MethodA; // Создание экземпляра делегата, указывающего на MethodA
```

```
        del("Hello"); // Вызов делегата, вызовет MethodA
```

```
        del = obj.MethodB; // Изменение делегата, теперь он указывает на MethodB
```

```
del("World"); // Вызов делегата, вызовет MethodB
}
}
```

В этом примере делегат MyDelegate объявляется для методов, принимающих строку в качестве аргумента и не возвращающих никакого значения. Затем создается экземпляр делегата, который может быть использован для вызова различных методов, соответствующих сигнатуре делегата.

Делегаты широко используются в C# для реализации событий, обратного вызова и других шаблонов проектирования

Регулярные выражения

В C# регулярные выражения предоставляют мощный инструмент для работы с текстом, поиска и обработки подстрок на основе шаблонов. Вот пример использования регулярных выражений в C#:

```
csharp
using System;
using System.Text.RegularExpressions;
class Program
{
    static void Main()
    {
        string input = "Регулярные выражения - мощный инструмент";
        string pattern = "регулярн[a-я]+ выражен[a-я]+";
        // Создание регулярного выражения
        Regex regex = new Regex(pattern, RegexOptions.IgnoreCase);
        // Поиск совпадений
        Match match = regex.Match(input);
        if (match.Success)
        {
            Console.WriteLine("Найдено совпадение: " + match.Value);
        }
        else
        {
            Console.WriteLine("Совпадений не найдено");
        }
        // Замена совпадений
        string replaced = regex.Replace(input, "Regex");
        Console.WriteLine("Замененная строка: " + replaced);
    }
}
```

В этом примере создается регулярное выражение для поиска слова "регулярные выражения" в строке, игнорируя регистр. Затем производится поиск совпадений и замена найденных совпадений.

Регулярные выражения в C# предоставляют широкие возможности для работы с текстом, включая поиск, замену, извлечение подстрок и многое другое. Они являются важным инструментом для обработки текстовых данных в приложениях на C#

Коллекции. Параметризованные классы.

В C# параметризованные классы (generic classes) позволяют создавать классы, структуры, интерфейсы и методы, которые могут работать с различными типами данных без необходимости повторного создания кода для каждого типа. Они обеспечивают безопасность типов и повышают повторное использование кода.

Вот пример использования параметризованной коллекции List<T> в C#, где T - это параметр типа:

```
csharp
```

```

using System;
using System.Collections.Generic;

class Program
{
    static void Main()
    {
        // Создание параметризованной коллекции List<T>
        List<int> numbers = new List<int>();
        numbers.Add(10);
        numbers.Add(20);
        numbers.Add(30);
        // Перебор элементов коллекции
        foreach (int number in numbers)
        {
            Console.WriteLine(number);
        }
    }
}

```

В этом примере `List<int>` представляет параметризованную коллекцию, способную хранить целые числа. Параметризованные классы позволяют использовать один и тот же код для работы с различными типами данных, обеспечивая безопасность типов и повторное использование кода.

Помимо `List<T>`, в стандартной библиотеке C# также предоставляются другие параметризованные коллекции, такие как `Dictionary<TKey, TValue>`, `Queue<T>`, `Stack<T>` и другие, которые могут быть использованы для работы с различными типами данных

Указатели

В C# указатели (pointers) представляются как неотъемлемая часть языка, однако они доступны только в небезопасном коде (unsafe code). Указатели позволяют непосредственно манипулировать памятью, что делает их мощным инструментом, но требует осторожности из-за потенциальных проблем безопасности и утечек памяти. Вот пример использования указателей в небезопасном коде C#:

```

csharp
using System;
class Program
{
    static unsafe void Main()
    {
        int number = 10;
        // Объявление указателя
        int* pointer = &number;
        // Использование указателя для изменения значения
        *pointer = 20;

        Console.WriteLine("Значение переменной: " + number);
    }
}

```

В этом примере ключевое слово "unsafe" используется для объявления метода Main как небезопасного, что позволяет использовать указатели. Затем создается указатель на переменную number, и значение переменной изменяется через указатель.

Использование указателей в C# обычно ограничено и рекомендуется только в специальных случаях, таких как работа с низкоуровневыми операциями или взаимодействие с некоторыми внешними библиотеками. Однако в большинстве случаев

рекомендуется избегать использования указателей в C# в пользу безопасных конструкций языка

Операции со списками

В C# операции со списками обычно выполняются с использованием коллекций, таких как List<T> из пространства имен System.Collections.Generic. Вот несколько основных операций, которые можно выполнять со списками в C#:

Создание списка и добавление элементов:

```
csharp
List<int> numbers = new List<int>();
numbers.Add(10);
numbers.Add(20);
numbers.Add(30);
```

Доступ к элементам списка по индексу:

```
int firstNumber = numbers[0]; // Получение первого элемента списка
```

Удаление элементов из списка:

```
numbers.Remove(20); // Удаление элемента со значением 20
```

Поиск элементов в списке:

```
int index = numbers.IndexOf(30); // Поиск индекса элемента со значением 30
```

Перебор элементов списка с помощью цикла foreach:

```
foreach (int number in numbers)
{
    Console.WriteLine(number);
}
```

Сортировка элементов списка:

```
numbers.Sort(); // Сортировка элементов списка
```

Проверка наличия элемента в списке:

```
bool contains = numbers.Contains(10); // Проверка наличия элемента со значением 10
```

Это лишь небольшой набор операций, которые можно выполнять со списками в C#.

List<T> предоставляет широкий набор методов для управления элементами списка, поиска, сортировки и многих других операций.

Операции со списками

В C# операции со списками обычно выполняются с использованием коллекций, таких как List<T> из пространства имен System.Collections.Generic. Вот несколько основных операций, которые можно выполнять со списками в C#:

Создание списка и добавление элементов:

```
List<int> numbers = new List<int>();
numbers.Add(10);
numbers.Add(20);
numbers.Add(30);
```

Доступ к элементам списка по индексу:

```
int firstNumber = numbers[0]; // Получение первого элемента списка
```

Удаление элементов из списка:

```
numbers.Remove(20); // Удаление элемента со значением 20
```

Поиск элементов в списке:

```
int index = numbers.IndexOf(30); // Поиск индекса элемента со значением 30
```

Перебор элементов списка с помощью цикла foreach:

```
foreach (int number in numbers)
{
    Console.WriteLine(number);
}
```

Сортировка элементов списка:

```
numbers.Sort(); // Сортировка элементов списка
```

Проверка наличия элемента в списке:

```
bool contains = numbers.Contains(10);
```

```
// Проверка наличия элемента со значением 10
```

Это лишь небольшой набор операций, которые можно выполнять со списками в C#. List<T> предоставляет широкий набор методов для управления элементами списка, поиска, сортировки и многих других операций.

Назначение и виды паттернов.

Паттерны проектирования — это шаблоны или образцы, применяющиеся для создания информационных систем. В повседневной работе программисты сталкиваются с большим количеством однотипных задач. Чтобы не тратить время на составление решения алгоритма функции вручную, разрабатываются паттерны. Они значительно ускоряют и упрощают процесс создания новых цифровых продуктов. Каждый IT-специалист должен знать, что такое паттерны проектирования в программировании и как ими пользоваться

Типы паттернов проектирования

Порождающие паттерны

Шаблоны, ответственные за процесс инстанцирования. Основная функция — уменьшение зависимости программы метода формирования композиции и представления объектов.

Виды порождающих паттернов проектирования:

Прототип (Prototype). Шаблон создается для представления конкретных объектов. При повторном использовании экземпляра возможна его трансформация или применение базового кода для создания новых элементов команды. Основной принцип создания подобных образцов — программирование через интерфейсы или внешний вид. Внутри системы объекты разделяются по классам в иерархической взаимосвязи. При помощи прототипов можно создавать объекты клонированием, а не конструированием.

Фабричный метод (Factory Method). Паттерн, анализирующий внешний вид для объединения нескольких элементов в отдельный супер-класс.

Абстрактная фабрика (Abstract factory). Шаблон для компоновки объектов по конкретному признаку интерфейса. Ранее определенные классы при этом не используются. Для новой группы создается отдельная «фабрика». Паттерн применяется для создания компонентов системы.

Одиночка (Singleton). Шаблон для формирования эксклюзивной точки софта. Элемент формируется в единственном экземпляре, но наделяется глобальным доступом.

Строитель (Builder). Образец создания сложных объектов, позволяющий реализовать идею программиста за счет автоматизированного разделения экземпляра на компоненты — конструирование и представление.

Ленивая инициализация (Lazy Initialization). Решение проектирования, когда классы и объекты подключаются по мере необходимости.

Основные шаблоны.

В C# основные шаблоны включают в себя:

Шаблон проектирования "Фабрика" (Factory): используется для создания объектов без указания конкретных классов.

Шаблон проектирования "Одиночка" (Singleton): гарантирует, что класс имеет только один экземпляр, и предоставляет глобальную точку доступа к этому экземпляру.

Шаблон проектирования "Стратегия" (Strategy): определяет семейство алгоритмов, инкапсулирует каждый из них и делает их взаимозаменяемыми.

Шаблон проектирования "Наблюдатель" (Observer): устанавливает отношение "один ко многим" между объектами, так что при изменении состояния одного объекта все зависящие от него объекты уведомляются и обновляются.

Это лишь несколько примеров. В C# также используются многие другие шаблоны проектирования для решения различных задач.

Порождающие шаблоны

Порождающие шаблоны - это паттерны проектирования, которые используются для создания объектов. Они помогают сделать процесс создания экземпляров классов более гибким и удобным. Некоторые из наиболее известных порождающих шаблонов включают в себя "Фабричный метод", "Абстрактная фабрика", "Одиночка" и "Строитель". Каждый из них предоставляет специфический подход к созданию объектов в соответствии с определенными потребностями проектирования.

Структурные шаблоны

Структурные шаблоны - это паттерны проектирования, которые направлены на организацию различных классов и объектов в более крупные структуры. Они помогают создавать более гибкие и эффективные программные системы путем определения отношений между классами и объектами. Некоторые из наиболее известных структурных шаблонов включают в себя такие концепции, как адаптер, мост, компоновщик, декоратор, фасад, прокси и другие. Каждый из этих шаблонов предоставляет свои собственные способы организации кода для достижения определенных целей, таких как повышение гибкости, уменьшение связанности или улучшение повторного использования кода.

Поведенческие шаблоны

Поведенческие шаблоны - это паттерны проектирования, которые управляют взаимодействием между объектами и классами, помогая им работать вместе более эффективно. Эти шаблоны фокусируются на способах организации обязанностей между объектами, управлении алгоритмами, управлении обменом информацией и управлении потоком управления. Некоторые из наиболее известных поведенческих шаблонов включают в себя такие концепции, как стратегия, состояние, наблюдатель, цепочка обязанностей, команда, посредник и другие. Каждый из этих шаблонов предоставляет свои собственные способы управления взаимодействием между объектами для достижения определенных целей, таких как уменьшение связанности, управление изменениями или увеличение гибкости системы.

Событийно-управляемое программирование

Событийно-управляемое программирование (Event-Driven Programming) – это подход к программированию, основанный на обработке событий, которые происходят в системе или приложении. В данной парадигме код реагирует на события, такие как нажатие кнопки, получение данных или изменение состояния программы, и выполняет соответствующие действия.

Основные принципы событийно-управляемого программирования включают следующее:

1. Разделение ответственностей: Каждый компонент приложения отвечает только за обработку определенных событий и взаимодействие с другими компонентами через событийную модель.
2. Асинхронность: Событийно-управляемое программирование позволяет выполнению кода не последовательно, а в ответ на возникновение событий, что обеспечивает быстрое действие и отзывчивость приложения.
3. Модульность: Событийно-управляемое программирование способствует созданию модульного кода, который легко поддерживать и расширять.
4. Гибкость: Событийно-управляемое программирование позволяет быстро и удобно изменять поведение приложения, добавляя новые события и обработчики.

Этот подход широко используется в различных областях разработки программного обеспечения, включая интерфейсы пользователя, мобильные приложения, серверные приложения и веб-разработку.

Элементы управления. Диалоговые окна. Обработчики событий.

Элементы управления - это различные компоненты интерфейса пользователя, такие как кнопки, поля ввода, выпадающие списки и т.д., которые позволяют пользователю

взаимодействовать с приложением.

Диалоговые окна - это специальные окна, которые могут появляться на экране для отображения важной информации или запроса дополнительных данных или подтверждения действия от пользователя.

Обработчики событий - это функции или методы, которые вызываются в ответ на определенное событие, например, нажатие кнопки или ввод текста. Обработчики событий позволяют программе реагировать на действия пользователя и выполнить соответствующие действия.

Введение в графику

Графика в C# представляет собой процесс создания и отображения изображений или графических элементов на экране. C# предоставляет различные способы работы с графикой, включая использование графических библиотек и API.

Для работы с графикой в C# обычно используется библиотека System.Drawing, которая предоставляет классы и методы для создания и редактирования изображений. С помощью класса Graphics можно рисовать на объекте Bitmap или на форме Windows Forms. Также для работы с 3D-графикой можно использовать библиотеки типа OpenTK или DirectX.

Для начала работы с графикой в C# необходимо создать новый проект в Visual Studio и подключить библиотеку System.Drawing. Затем можно создавать объекты Graphics, задавать им параметры и рисовать на них различные графические элементы, такие как линии, круги, прямоугольники и т.д.

Также в C# можно использовать библиотеки для работы с графическими форматами, такими как JPEG, PNG, BMP и другие. Это позволяет работать с изображениями и их форматами непосредственно в коде программы.

В целом, работа с графикой в C# достаточно проста и удобна, благодаря широким возможностям, предоставляемым языком и библиотеками. Она позволяет создавать различные визуальные эффекты и элементы интерфейса, делая программы более привлекательными и функциональными.

Методы оптимизации программного кода

Использование эффективных структур данных, таких как хэш-таблицы, деревья и очереди для более быстрого доступа к данным.

Пересмотр алгоритмов работы программы с использованием алгоритмов оптимизации, таких как динамическое программирование или жадные алгоритмы.

Оптимизация циклов и итераций в программе, чтобы избежать лишних операций или повторного вычисления.

Использование многопоточности и параллелизма для увеличения производительности программы.

Минимизация использования ресурсов, таких как память и процессорное время, путем оптимизации алгоритмов и структур данных.

Использование инструментов профилирования кода для выявления и исправления узких мест в программе.

Использование более эффективных методов работы с коллекциями, такими как LINQ, для уменьшения затрат на операции с данными.

Оптимизация работы с файлами и вводом/выводом данных для ускорения работы программы.

Избегание избыточной проверки условий и операций, которые не влияют на результат выполнения программы.

Постоянная проверка и оптимизация кода с использованием лучших практик и рекомендаций по оптимизации кода в C#.

Правила разработки интерфейсов пользователя

Понимание целей и потребностей пользователей. Прежде чем приступить к разработке интерфейса, необходимо понять, какие задачи будут выполнять пользователи и какие

функции им необходимы.

Простота и понятность. Интерфейс должен быть интуитивно понятным и легко воспринимаемым пользователями. Избегайте излишней сложности и перегруженности информацией.

Согласованность и единообразие. Все элементы интерфейса должны быть выполнены в одном стиле и соответствовать общей концепции дизайна.

Навигация. Разместите элементы интерфейса так, чтобы пользователи могли легко находить необходимую информацию и выполнять нужные действия.

Адаптивность. Учтите возможность использования интерфейса на различных устройствах (компьютеры, планшеты, смартфоны) и различных разрешениях экранов.

Визуальное оформление. Используйте цвета, шрифты, иконки и другие визуальные элементы таким образом, чтобы они подчеркивали основные функции и информацию.

Тестирование. Проведите тестирование интерфейса на реальных пользователях, чтобы выявить возможные проблемы и улучшить его эффективность.

Обратная связь. Предоставьте пользователям возможность дать обратную связь о работе интерфейса и предложить свои идеи по его улучшению.

Работа с базами данных

Базы данных являются основой многих современных систем управления данными. Они позволяют организовать и хранить данные в структурированном виде, что обеспечивает удобный доступ, обновление и поиск информации.

Основные понятия

База данных - это организованный набор данных, относящихся к определенной предметной области.

Таблица - это основная структура базы данных, представляющая собой набор записей (строк) и полей (столбцов).

Запись - это набор данных, относящихся к одному конкретному объекту или событию.

Поле - это отдельная единица информации в записи, например, имя, адрес или дата рождения.

Типы баз данных

Существует множество типов баз данных, каждый из которых предназначен для конкретных целей:

Реляционные базы данных - используют табличную структуру и устанавливают связи между таблицами.

Иерархические базы данных - представляют данные в древовидной структуре.

Сетевые базы данных - используют связи типа "многие ко многим" для соединения записей.

Объектно-реляционные базы данных - сочетают реляционную модель с объектно-ориентированными концепциями.

Системы управления базами данных (СУБД)

СУБД - это программное обеспечение, которое позволяет создавать, управлять и администрировать базы данных. Она предоставляет инструменты для создания таблиц, добавления, обновления и удаления записей, выполнения запросов и обеспечения безопасности.

Языки баз данных

Для взаимодействия с базами данных используются специальные языки:

SQL (Structured Query Language) - стандартный язык запросов к базам данных.

PL/SQL (Procedural Language/SQL) - расширение SQL, позволяющее выполнять сложные процедуры.

NoSQL (Not Only SQL) - группа языков, разработанных для управления нереляционными базами данных.

Доступ к данным

Доступ к данным в базе данных осуществляется через систему управления базами

данных (СУБД). Пользователи могут взаимодействовать с базой данных с помощью различных методов:

Языки запросов

Языки запросов, такие как SQL (Structured Query Language), позволяют пользователям извлекать и манипулировать данными базы данных. Пользователи могут создавать запросы для выборки определенных записей, обновления данных, вставки новых записей или удаления существующих записей.

Интерфейсы прикладного программирования (API)

API предоставляют программный интерфейс для взаимодействия с базой данных.

Программисты могут использовать API для создания приложений, которые подключаются к базе данных и выполняют различные операции, такие как извлечение данных, обновление записей или выполнение транзакций.

Графические пользовательские интерфейсы (GUI)

Многие СУБД предоставляют графические пользовательские интерфейсы (GUI), которые позволяют пользователям взаимодействовать с базой данных без необходимости писать код. GUI обычно включают в себя такие функции, как навигация по таблицам, редактирование записей и выполнение простых запросов.

Роли и разрешения

СУБД позволяют администраторам базы данных управлять доступом пользователей к данным путем назначения ролей и разрешений. Роли определяют набор привилегий, которые пользователь имеет для выполнения определенных операций с базой данных. Разрешения назначаются отдельным объектам базы данных, таким как таблицы, представления и процедуры.

Тип доступа

Существует несколько типов доступа к данным:

Чтение - пользователи могут только извлекать данные из базы данных.

Запись - пользователи могут вставлять, обновлять и удалять данные в базе данных.

Выполнение - пользователи могут выполнять хранимые процедуры и другие исполняемые объекты в базе данных.

Контроль доступа к данным важен для обеспечения безопасности и целостности данных. Администраторы базы данных должны тщательно управлять ролями и разрешениями пользователей, чтобы предотвратить несанкционированный доступ к конфиденциальным данным

Создание таблицы, работа с записями.

Чтобы создать таблицу в базе данных, можно использовать следующий синтаксис SQL:

```
CREATE TABLE table_name (  
    column1 data_type,  
    column2 data_type,  
    ...  
    columnN data_type  
);
```

Например, чтобы создать таблицу с именем **students** с полями **id**, **name** и **age**, можно использовать следующий запрос:

```
CREATE TABLE students (  
    id INT NOT NULL AUTO_INCREMENT,  
    name VARCHAR(255) NOT NULL,  
    age INT NOT NULL  
);
```

В этом примере:

id - поле типа **INT** (целое число), которое является первичным ключом таблицы.

name - поле типа **VARCHAR(255)**, которое может хранить строку длиной до 255 символов.

`age` - поле типа `INT`, которое может хранить целые числа.

Работа с записями

После создания таблицы можно вставлять, обновлять и удалять записи с помощью следующих операторов SQL:

Вставка записей:

```
INSERT INTO table_name (column1, column2, ..., columnN) VALUES (value1, value2, ..., valueN);
```

Например, чтобы вставить запись в таблицу `students` с именем `John` и возрастом `21`, можно использовать следующий запрос:

```
INSERT INTO students (name, age) VALUES ('John', 21);
```

Обновление записей:

```
UPDATE table_name SET column1 = value1, column2 = value2, ..., columnN = valueN WHERE condition;
```

Например, чтобы обновить имя студента с `id` равным `1` на `Jane`, можно использовать следующий запрос:

```
UPDATE students SET name = 'Jane' WHERE id = 1;
```

Удаление записей:

```
DELETE FROM table_name WHERE condition;
```

Например, чтобы удалить студента с `id` равным `1` из таблицы `students`, можно использовать следующий запрос:

```
DELETE FROM students WHERE id = 1;
```

Пример

Рассмотрим следующий пример создания таблицы и работы с записями:

```
-- Создать таблицу
CREATE TABLE students (
  id INT NOT NULL AUTO_INCREMENT,
  name VARCHAR(255) NOT NULL,
  age INT NOT NULL
);
-- Вставить записи
INSERT INTO students (name, age) VALUES ('John', 21);
INSERT INTO students (name, age) VALUES ('Jane', 22);
INSERT INTO students (name, age) VALUES ('Peter', 23);
-- Обновить запись
UPDATE students SET name = 'John Doe' WHERE id = 1;
-- Удалить запись
DELETE FROM students WHERE id = 3;
```

Способы создания команд

Существует несколько способов создания команд SQL:

Использование клиентской программы

Многие СУБД предоставляют клиентские программы, которые позволяют пользователям взаимодействовать с базой данных и выполнять команды SQL.

Например, для PostgreSQL есть клиентская программа под названием `psql`, а для MySQL - клиент `mysql`.

Встраивание в код приложения

Можно встраивать команды SQL в код приложения на любом языке программирования, который поддерживает доступ к базе данных. Например, в Python можно использовать библиотеку `psycopg2` для подключения к базе данных PostgreSQL и выполнения команд SQL.

Использование веб-интерфейса

Некоторые СУБД предоставляют веб-интерфейсы, которые позволяют пользователям выполнять команды SQL без необходимости использования клиентской программы.

Например, в PostgreSQL есть веб-интерфейс под названием pgAdmin, а в MySQL - phpMyAdmin.

Написание скриптов SQL

Команды SQL можно также записывать в файлы скриптов, которые затем можно выполнять с помощью клиентской программы или утилиты командной строки.

Например, можно создать файл с именем script.sql, содержащий следующие команды:

```
CREATE TABLE students (  
  id INT NOT NULL AUTO_INCREMENT,  
  name VARCHAR(255) NOT NULL,  
  age INT NOT NULL  
);
```

```
INSERT INTO students (name, age) VALUES ('John', 21);
```

```
INSERT INTO students (name, age) VALUES ('Jane', 22);
```

Затем этот скрипт можно выполнить с помощью клиентской программы, указав файл скрипта в качестве аргумента командной строки:

```
psql -f script.sql
```

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ ДАГЕСТАН

Частное профессиональное образовательное учреждение

«Республиканский полипрофессиональный колледж»

Самостоятельная работа при изучении МДК.01.01 Разработка программных модулей

Самостоятельная работа 1.1 Разработка интерфейса типа «Вопрос-ответ»

Порядок выполнения

1. Запишите вариант в отчет.

2. Согласно поставленной задаче выполните ручную отладку:

- Разработайте с помощью компонента TabControl форму для прохождения тестирования;
- Опишите спецификацию программы;
- Запишите алгоритм программы;
- Выполните отладку логики программы методом «грубой силы» с помощью соседа;
- Составьте тестовые наборы для проверки функционала системы.

3. Результаты выполнения практического задания запишите в отчет.

Контрольные вопросы:

1. В чем заключается функция компонента Table?
2. В чем различия компонентов radiobutton и checkbox?
3. Опишите методы отладки.

Самостоятельная работа 1.2 Описание диалогового окна и стандартных элементов

Порядок выполнения

1. Запишите вариант в отчет.

2. Согласно поставленной задаче выполните ручную отладку:

- Разработайте с помощью компонента MenuStrip главное окно программы;
- Опишите спецификацию программы;
- Запишите алгоритм программы;
- Выполните отладку логики программы методом «грубой силы» с помощью соседа;
- Составьте тестовые наборы для проверки функционала системы.

3. Результаты выполнения практического задания запишите в отчет.

Контрольные вопросы

1. В чем заключается функция компонента Table?
2. В чем различия визуальных и не визуальных компонентов?
3. Опишите методы отладки.

Самостоятельная работа № 1.3 Составление алгоритмов для решения задач

Порядок выполнения

1. Запишите вариант в отчет.
2. Согласно поставленной задаче выполните ручную отладку:
 - Опишите математическую модель задачи с указанием имен и назначения переменных;
 - Опишите спецификацию программы;
 - Запишите алгоритм программы;
 - Выполните отладку логики программы методом «грубой силы» с помощью соседа;
 - Составьте тестовые наборы для проверки функционала системы.
3. Результаты выполнения практического задания запишите в отчет.

Контрольные вопросы

1. В чем заключается ручная отладка ПО?
2. На каком этапе проводится ручная отладка?
3. Опишите методы отладки.

Варианты заданий

Создать Windows-приложение, реализующие линейный и разветвляющийся алгоритмы, которые размещены на разных вкладках окна формы. На вкладке линейного алгоритма предусмотреть поля ввода значений переменных и поле вывода результата вычисления. На вкладке разветвляющегося алгоритма предусмотреть поля для ввода значений переменных, поле вывода результатов расчета по одной из трех формул в зависимости от результата выполнения условия. В качестве $f(x)$ использовать по выбору: $\cos(x)$ или x^2 или e^x . Пример рабочей формы представлен на рисунке 1.

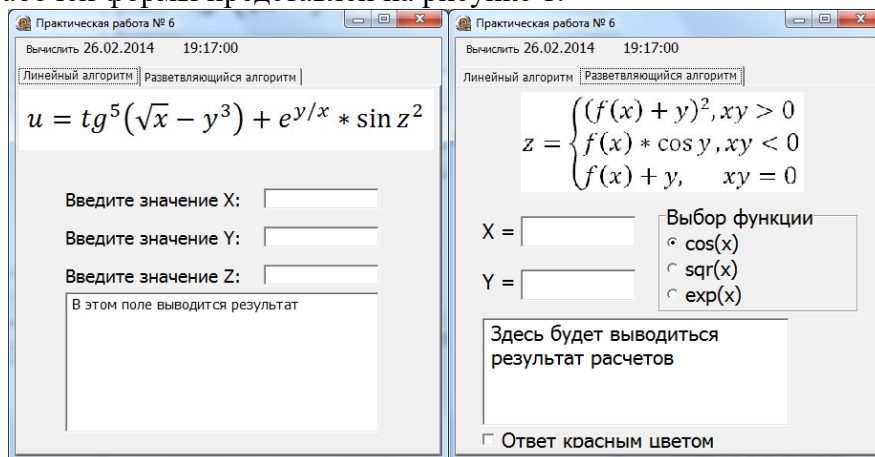


Рисунок 1 – Windows-приложение
Линейный алгоритм:

1. $t = \frac{2 \cos(x - \frac{\pi}{6})}{0.5 + \sin^2 y} \left(1 + \frac{z^2}{3 - \frac{z^2}{5}} \right)$.
2. $u = \frac{\sqrt[3]{8 + |x - y|^2 + 1}}{x^2 + y^2 + 2} - e^{|x - y|} * (tg^2 z + 1)^x$.
3. $v = \frac{1 + \sin^2(x + y)}{\left| x - \frac{2y}{1 + x^2 y^2} \right|} * x^{|y|} + \cos^2 \left(\text{arctg} \frac{1}{z} \right)$.

$$4. w = |\cos x - \cos y|^{1+2 \sin^2 y} * \left(1 + z + \frac{z^2}{2} + \frac{z^3}{3} + \frac{z^4}{4}\right).$$

$$5. \alpha = \ln\left(y^{-\sqrt{|x|}}\right) * \left(x - \frac{y}{2}\right) + \sin^2 \operatorname{arctg}(z).$$

$$6. \beta = \sqrt{10(\sqrt[3]{x} + x^{y+2})} * (\arcsin^2 z - |x - y|).$$

$$7. \gamma = 5 \operatorname{arctg}(x) - \frac{1}{4} \arccos(x) * \frac{x+3|x-y|+x^2}{|x-y|z+x^2}.$$

Разветвляющийся алгоритм:

$$1. a = \begin{cases} \ln(y+2) + f(x), & x/y > 0 \\ \ln|y| - \operatorname{tg}(f(x)), & x/y < 0 \\ f(x) * y^3, & \text{иначе} \end{cases} \quad 2. b = \begin{cases} \ln|y + f(x)| + 3, & 3 < xy < 8 \\ \cos(f(x)) - y, & xy > 12 \\ \operatorname{sh}(f(x) + \operatorname{cs}(y)), & \text{иначе} \end{cases}$$

$$3. c = \begin{cases} f^3(x) + \operatorname{ctg}(y), & xy > 12 \\ \operatorname{sh}(f^3(x)) + y^2, & xy < 7 \\ \cos(x - f^3(x)), & \text{иначе} \end{cases} \quad 4. d = \begin{cases} \operatorname{ctg}(y) + f(x), & x/y > 0 \\ \ln|y| + \operatorname{tg}(f(x)), & x/y < 0 \\ f(x) * y^3, & \text{иначе} \end{cases}$$

$$5. k = \begin{cases} (f(x) + y)^2, & 4 > xy > 1 \\ f(x) * \operatorname{tg}(y), & 8 < xy < 10 \\ f(x) + y, & \text{иначе} \end{cases} \quad 6. l = \begin{cases} f^2(x) + \operatorname{arctg}(f(x)), & 1 \leq x < 5 \\ (y - f(x))^2 + \operatorname{arctg}(f(x)), & y > x \\ (y + f(x))^3 + 0.5, & \text{иначе} \end{cases}$$

$$7. m = \begin{cases} f^3(x) + \sin(y), & xy < 5 \\ \operatorname{ch}(f^3(x)) + y^2, & xy > 7 \\ \cos(x + f^3(x)), & \text{иначе} \end{cases} \quad 8. l = \begin{cases} e^{f(x)-|y|}, & 0.5 < xy < 5 \\ \sqrt{|f(x) + y|}, & 0.1 < xy < 0.5 \\ 2f^2(x), & \text{иначе} \end{cases}$$

Содержание отчета

1. Тема.
2. Цель.
3. Оборудование.
4. Результат выполнения практического задания.
5. Ответы на контрольные вопросы.
6. Вывод.

Критерии оценки:

оценки «хорошо» заслуживает студент, обнаруживший полные знания учебно-программного материала, успешно выполняющий предусмотренные в программе задания, усвоивший основную литературу, рекомендованную в программе. Как правило, оценка «хорошо» выставляется студентам, показавшим систематический характер знаний по дисциплине и способным к их самостоятельному пополнению и обновлению в ходе дальнейшей учебной работы и профессиональной деятельности;

оценки «удовлетворительно» заслуживает студент, обнаруживший знание учебно-программного материала в объеме, необходимом для дальнейшей учебы и предстоящей работы по профессии, справляющийся с выполнением заданий, предусмотренных программой, знакомый с основной литературой, рекомендованной программой. Как правило, оценка «удовлетворительно» выставляется студентам, допустившим

погрешность в ответе на экзамене и при выполнении экзаменационных заданий, но обладающим необходимыми знаниями для их устранения под руководством преподавателя;

оценка «неудовлетворительно» выставляется студенту, обнаружившему пробелы в знаниях основного учебно-программного материала, допустившему принципиальные ошибки в выполнении предусмотренных программой заданий. Как правило, оценка «неудовлетворительно» ставится студентам, которые не могут продолжить обучение или приступить к профессиональной деятельности по окончании вуза без дополнительных занятий по соответствующей дисциплине».

Составители:

_____ Ж.А. Кадрышева

3.2. Задания для оценки освоения МДК 01.02 Поддержка и тестирование программных модулей

Практические и лабораторные работы

Практические и лабораторные работы Тема 2.1 Отладка и тестирование программного обеспечения

Практическая работа № 1. Тестирование «белым ящиком»

Цель практической работы: изучить метод тестирования «Белым ящиком»

Задание: Подготовить набор тестовых вариантов для обнаружения ошибок в программе.

Результат оформить в следующем виде:

А	В	С	Ожидаемый результат	Объект проверки
Значение	Значение	Значение	Что должно получиться	Значения вводимых данных, либо ожидаемый результат
...

Эталон ответа:

Задание: Вычислитель площадь кольца по данным внешнему и внутреннему радиусу. Площадь кольца равна разнице между площадью внутреннего и внешнего круга. Объявить переменные вещественного типа. Результат вывести на экран.

Входные данные переменной	Входные данные переменной	Результат	Вычисляемая переменная
R	r		S
1	0.5	2,35619449019234	Math.PI * (R * R – r * r) находим площадь каждого круга и вычисляем разницу между ними

Варианты:

Задание 3. Разработать программу на C#.

Даны длины сторон треугольника, определить вид треугольника и его площадь.

Выполнить контроль вводимых чисел.

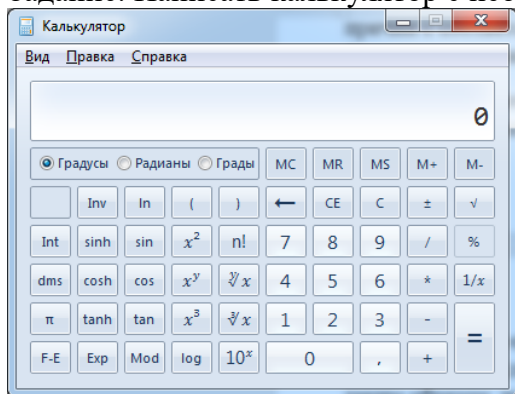
1. Остроугольный треугольник
2. Тупоугольный треугольник
3. Прямоугольный треугольник Ограничения: - три числа не могут быть определены как стороны треугольника; - если хотя бы одно из них меньше или равно 0; - сумма двух из них меньше третьего. Подготовить набор тестовых вариантов для обнаружения ошибок в программе и оформить результат.

Задание 4. На основании проведенных тестов составьте рекомендации по исправлению ошибок, выявленных в ходе тестирования в виде отчета. Пример: 1 тест. В ходе проведения первого теста было обнаружено, что при введении не корректных данных площадь все равно высчитывается. Рекомендуется: в случае, если пользователь введет не корректные данные, следует выводить сообщение с просьбой исправить введенные значения. Добавить в программу проверку введенных значений на соответствие ограничения.

Практическая работа № 2. Тестирование «черным ящиком»

Цель практической работы: изучить метод тестирования «Черным ящиком»

Задание: Написать калькулятор с небольшими багами.



Задание 2. Обменяться программой с другими студентами. Провести тестирование и написать отчет в тетради.

Пример:

Название теста	Описание сценария	Входные данные	Выходные данные	Удачное/неудачное тестирование	Предложения по исправлению найденных ошибок.	Пожелания пользователей
Функция суммы	Сложение двух положительных чисел; Проверка результата	Первая переменная=3 Вторая переменная=8	Результат=11	Неудачное	-	Поле для ввода значений и вывода, объединить

Эталон ответа:

Задание 1: написать калькулятор с небольшими багами.

Рисунок 1 – Работа Рамазана

Задание 2: Обменяться программой с другими студентами. Провести тестирование и написать отчёт.

Таблица 2

Название теста	Описание сценария	Входные данные	Выходные данные	Успешное/неуспешное тестирование	Предложения по исправлению	Положения пользователей
Функция суммы	Сложение двух чисел; Проверка результата	A = 4 B = 4	Результат: 0	Неуспешное	-	нет
Функция разности	Вычитание двух чисел; Проверка результата	A = 2 B = 12	Результат: 14	Успешное	-	нет
Функция произведения	Умножение двух чисел; Проверка результата	A = 3 B = 3	Результат: 6	Успешное	-	нет
Функция деления	Деление двух чисел; Проверка	A = 10 B = 5	Результат: 0	Неуспешное	-	нет

	а результата					
--	-----------------	--	--	--	--	--

Варианты:

Задание 1. Создать приложение для определения вида треугольника и вычисления его периметра и площади. При создании использовать шаблон Windows Forms.

Требования к приложению.

Приложение должно:

Проверять существование треугольника

Определять тип треугольника (разносторонний, равносторонний, равнобедренный, прямоугольный, остроугольный или тупоугольный) по длинам его сторон.

Вычислять площадь треугольника.

Вычислять периметр треугольника.

Задание 2. Составить спецификацию переменных.

Имя переменной в программе	Назначение переменной в программе	Тип переменной	Диапазон типа
а	Сторона треугольника	int	От -2 147 483 648 до 2 147 483 647
...

Задание 3. Подготовить тестовый сценарий (набор тест-кейсов) для обнаружения ошибок в программе.

Результат оформить в следующем виде:

А	В	С	Ожидаемый результат	Реальный результат
Значение	Значение	Значение	Что должно получиться	Что получилось
...

Задание 4. На основании проведенных тестов составьте отчет об ошибках (баг-репорт)

Пример оформления баг-репорта:

№ теста	Описание ошибки	Рекомендации
1	В ходе проведения первого теста было обнаружено, что при введении отрицательных	1) Добавить в программу проверку введенных значений на соответствие ограничения. 2) В случае,

	значений площадь все равно вычисляется.	если пользователь введет некорректные данные, следует выводить сообщение с просьбой исправить введенные значения.
--	--	--

Задание 5. Выполните тестирование по чек-листу из приложения 1.

Практическая работа № 3. Модульное тестирование

Цель практической работы: изучить возможность создания автоматических тестов, для модульного тестирования.

Задание:

Создание проекта программы, модули которого будут тестироваться.

Эталон ответа:

```

1 using System;
2 using Microsoft.VisualStudio.TestTools.UnitTesting;
3 using MathTaskClassLibrary;
4
5 namespace MathTaskClassLibraryTests
6 {
7     [TestClass]
8     public class GeometryTests
9     {
10        [TestMethod]
11        public void RectangleArea_3and5_15returned()
12        {
13            // исходные данные
14            int a = 3;
15            int b = 5;
16            int expected = 15;
17
18            // получение значения с помощью тестируемого метода
19            Geometry g = new Geometry();
20            int actual = g.RectangleArea(a, b);
21
22            // сравнение ожидаемого результата с полученным
23            Assert.AreEqual(expected, actual);
24        }
25    }
26 }

```

Варианты:

Вариант 1. Дана целочисленная прямоугольная матрица $a * mn$. Написать программу, определяющую величины: 1) количество строк, не содержащих ни одного нулевого элемента; 2) максимальное значение из чисел, встречающихся в заданной матрице более одного раза; 3) суммы элементов строк и столбцов матрицы. Суммы элементов строк и столбцов матрицы упорядочить по возрастанию методом обмена.

Вариант 2. Дана целочисленная прямоугольная матрица $a * mn$. Написать программу, определяющую величины: 1) количество столбцов, не содержащих ни одного нулевого элемента; 2) максимальное и минимальное значение элементов матрицы; 3) среднее арифметическое значение элементов строк и столбцов матрицы. Средние

арифметические значения элементов строк и столбцов матрицы упорядочить по возрастанию методом выбора.
Вариант 3. Дана целочисленная прямоугольная матрица $a * mn$. Написать программу, определяющую величины: 1) количество столбцов, содержащих хотя бы один нулевой элемент; 2) номер строки, в которой находится самая длинная серия одинаковых элементов; 3) среднее геометрическое значение элементов строк и столбцов матрицы. Средние геометрические значения элементов строк и столбцов матрицы упорядочить по возрастанию методом вставки
Вариант 4. Дана целочисленная квадратная матрица $a * nn$. Написать программу, определяющую величины: 1) сумму элементов в строках, которые не содержат отрицательных элементов; 2) максимум среди элементов выше главной диагонали матрицы; 3) суммы четных элементов строк и столбцов матрицы. Суммы четных элементов строк и столбцов матрицы упорядочить по убыванию методом обмена.
Вариант 5. Дана целочисленная квадратная матрица $a * nn$. Написать программу, определяющую величины: 1) суммы элементов в столбцах, которые не содержат отрицательных элементов; 2) минимум среди модулей элементов ниже побочной диагонали матрицы; 3) суммы нечетных элементов строк и столбцов матрицы. Суммы нечетных элементов строк и столбцов матрицы упорядочить по убыванию методом выбора.
Вариант 6. Дана целочисленная прямоугольная матрица $a * mn$. Написать программу, определяющую величины: 1) суммы элементов в строках, которые содержат хотя бы один отрицательный элемент; 2) минимальный элемент матрицы; 3) среднее арифметическое значение четных элементов строк и столбцов матрицы. Средние арифметические значения четных элементов строк и столбцов матрицы упорядочить по убыванию методом вставки.
Вариант 7. Дана целочисленная прямоугольная матрица $a * mn$. Написать программу, определяющую величины: 1) количество отрицательных элементов матрицы; 2) максимальный элемент матрицы; 3) суммы элементов матрицы, находящихся ниже главной диагонали. Элементы матрицы, находящиеся выше главной диагонали, упорядочить по возрастанию методом обмена
Вариант 8. Дана целочисленная прямоугольная матрица $a * mn$. Написать программу, определяющую величины: 1) количество положительных элементов матрицы; 2) максимальный и минимальный нечетные элементы строк матрицы; 3) суммы элементов матрицы, находящихся ниже побочной диагонали. Элементы матрицы, находящиеся ниже побочной диагонали, упорядочить по возрастанию методом выбора.
Вариант 9. Дана целочисленная прямоугольная матрица $a * mn$. Написать программу, определяющую величины: 1) количество нулевых элементов матрицы; 2) максимальный и минимальный нечетные элементы столбцов матрицы; 3) суммы элементов матрицы, находящихся выше побочной диагонали. Элементы матрицы, находящиеся выше побочной диагонали, упорядочить по возрастанию методом вставки.
Вариант 10. Дана целочисленная прямоугольная матрица $a * mn$. Написать программу, определяющую величины: 1) количество ненулевых элементов матрицы; 2) максимальный и минимальный четные элементы столбцов матрицы; 3) суммы элементов матрицы, находящихся ниже побочной диагонали. Элементы матрицы, находящиеся ниже побочной диагонали, упорядочить по убыванию методом обмена.
Вариант 11. Дана целочисленная прямоугольная матрица $a * mn$. Написать программу, определяющую величины: 1) количество элементов матрицы, меньших величины B ; 2) максимальный и минимальный нечетные элементы столбцов матрицы; 3) суммы элементов матрицы, находящихся выше побочной диагонали. Элементы матрицы, находящиеся выше побочной диагонали, упорядочить по возрастанию методом вставки.
Вариант 12. Дана целочисленная прямоугольная матрица $a * mn$. Написать программу, определяющую величины: 1) количество элементов матрицы, больших величины B ; 2)

минимальный по модулю элемент матрицы; 3) суммы элементов, расположенных по периметру матрицы. Элементы, расположенные по периметру матрицы, упорядочить по убыванию методом обмена.

Вариант 13. Дана целочисленная квадратная матрица $a * n \times n$. Написать программу, определяющую величины: 1) количество элементов матрицы, находящихся в диапазоне от A до B; 2) максимальный по модулю элемент матрицы; 3) суммы элементов, расположенных на диагоналях матрицы. Элементы, расположенные на диагоналях матрицы, упорядочить по убыванию методом выбора.

Вариант 14. Дана целочисленная прямоугольная матрица $a * m \times n$. Написать программу, определяющую величины: 1) количество элементов матрицы, содержащих простые числа; 2) минимум среди модулей элементов, расположенных ниже побочной диагонали; 3) средние квадратичные значения столбцов и строк матрицы. Средние квадратичные значения столбцов и строк матрицы упорядочить по убыванию методом вставки.

Вариант 15. Дана целочисленная квадратная матрица $a * n \times n$. Написать программу, определяющую величины: 1) количество элементов матрицы, содержащих простые числа; 2) максимум среди модулей элементов, расположенных выше главной диагонали; 3) средние квадратичные значения четных столбцов и нечетных строк матрицы. Средние квадратичные значения четных столбцов и нечетных строк матрицы упорядочить по возрастанию методом обмена.

Вариант 16. Дана целочисленная квадратная матрица $a * n \times n$. Написать программу, определяющую величины: 1) количество элементов матрицы, содержащих простые числа; 2) минимум среди элементов, расположенных выше побочной диагонали; 3) средние квадратичные значения нечетных столбцов и четных строк матрицы. Средние квадратичные значения нечетных столбцов и четных строк матрицы упорядочить по возрастанию методом выбора.

Практическая работа № 4. Интеграционное тестирование

Цель практической работы: овладение навыками интеграционного тестирования

Задание:

закрепить теоретические знания и получить практические навыки в разработке программы тестирования методом восходящего тестирования

Эталон ответа:

создадим тест, который проверяет, создается ли объект типа TCommand, и добавляется ли команда в конец очереди

```
private void TCommandQueueTest1()
{
    LogMessage("///// TCommandQueue Test1 /////");
    LogMessage("Проверяем, создается ли
    объект типа TCommand");
    // В очереди нет команд
    dump();
    // Добавляем команду
    // параметр = -1 означает, что команда
    // должна быть добавлена в конец очереди
    CommandQueue.AddCommand(TCommand.GetR,0,0,0,
    new TBearingParam(),new TAxleParam(),-1);
    LogMessage("Command added");
    // В очереди одна команда
    dump();
}
```

Варианты:

<p>Вариант 1. В одномерном динамическом массиве, состоящем из n вещественных элементов, вычислить: 1) количество элементов массива, меньших C; 2) сумму целых частей элементов массива, расположенных после последнего отрицательного элемента. Преобразовать массив таким образом, чтобы сначала располагались все элементы, отличающиеся от максимального не более чем на 20%, а потом – все остальные.</p>
<p>Вариант 2. В одномерном динамическом массиве, состоящем из n вещественных элементов, вычислить: 1) произведение отрицательных элементов массива; 2) сумму положительных элементов массива, расположенных до максимального элемента. Изменить порядок следования элементов в массиве на обратный.</p>
<p>Вариант 3. В одномерном динамическом массиве, состоящем из n вещественных элементов, вычислить: 1) произведение положительных элементов массива; 2) сумму элементов массива, расположенных до минимального элемента. Упорядочить по возрастанию отдельно элементы, стоящие на четных местах, и элементы, стоящие на нечетных местах.</p>
<p>Вариант 4. В одномерном динамическом массиве, состоящем из n вещественных элементов, вычислить: 1) среднее арифметическое элементов, расположенных до первого и после последнего нулевых значений; 2) поменять местами первый и максимальный элементы, последний и минимальный элементы. Упорядочить по убыванию отдельно элементы, стоящие на четных местах, и элементы, стоящие на нечетных местах.</p>
<p>Вариант 5. В одномерном динамическом массиве, состоящем из n целых элементов, вычислить: 1) произведение элементов, находящихся между минимальным и максимальным элементами массива; 2) удалить из массива простое число. Упорядочить по возрастанию элементы массива.</p>
<p>Вариант 6. В одномерном динамическом массиве, состоящем из n вещественных элементов, вычислить: 1) произведение отрицательных элементов массива; 2) сумму элементов массива, расположенных до максимального элемента. Упорядочить по возрастанию отдельно элементы, стоящие на четных местах, и элементы, стоящие на нечетных местах.</p>
<p>Вариант 7. В одномерном динамическом массиве, состоящем из n вещественных элементов, вычислить: 1) среднее значение элементов, расположенных в массиве между первым последним нулевыми элементами; 2) поменять местами максимальный и минимальный элементы. Упорядочить по возрастанию отдельно элементы, стоящие на четных местах, и элементы, стоящие на нечетных местах.</p>
<p>Вариант 8. В одномерном динамическом массиве, состоящем из n целых элементов, определить: 1) количество совершенных чисел; 2) удалить из массива все нулевые элементы. Упорядочить по убыванию отдельно элементы, стоящие на четных местах, и элементы, стоящие на нечетных местах.</p>
<p>Вариант 9. В одномерном динамическом массиве, состоящем из n вещественных элементов, вычислить: 1) сумму отрицательных элементов массива; 2) произведение элементов массива, расположенных между максимальным и минимальным элементами. Упорядочить элементы массива по возрастанию.</p>
<p>Вариант 10. В одномерном динамическом массиве, состоящем из n вещественных элементов, вычислить: 1) сумму положительных элементов массива; 2) произведение элементов массива, расположенных между максимальным по модулю и минимальным по модулю элементами. Упорядочить элементы массива по убыванию.</p>
<p>Вариант 11. В одномерном динамическом массиве, состоящем из n целых элементов, вычислить: 1) произведение элементов массива с четными номерами; 2) сумму элементов массива, расположенных между первым и последним нулевыми элементами. Преобразовать массив таким образом, чтобы сначала располагались все положительные элементы, а потом – все отрицательные (элементы, равные нулю, считать</p>

положительными).
Вариант 12. В одномерном динамическом массиве, состоящем из n вещественных элементов, вычислить: 1) сумму элементов массива с нечетными номерами; 2) сумму элементов массива, расположенных между первым и последним отрицательными элементами. Сжать массив, удалив из него все элементы, модуль которых не превышает 1. Освободившиеся в конце массива элементы заполнить нулями.
Вариант 13. В одномерном динамическом массиве, состоящем из n вещественных элементов, вычислить: 1) максимальный элемент массива; 2) сумму элементов массива, расположенных до последнего положительного элемента. Сжать массив, удалив из него все элементы, модуль которых находится в интервале $[a, b]$. Освободившиеся в конце массива элементы заполнить нулями.
Вариант 14. В одномерном динамическом массиве, состоящем из n вещественных элементов, вычислить: 1) минимальный элемент массива; 2) сумму элементов массива, расположенных между первым и последним положительными элементами. Преобразовать массив таким образом, чтобы сначала располагались все элементы, равные нулю, а потом – все остальные.
Вариант 15. В одномерном динамическом массиве, состоящем из n целых элементов, вычислить: 1) номер максимального элемента массива; 2) произведение элементов массива, расположенных между первым и вторым нулевыми элементами. Преобразовать массив таким образом, чтобы в первой его половине располагались элементы, стоявшие в нечетных позициях, а во второй – элементы, стоявшие в четных позициях.
Вариант 16. В одномерном динамическом массиве, состоящем из n вещественных элементов, вычислить: 1) номер минимального элемента массива; 2) сумму элементов массива, расположенных между первым и вторым отрицательными элементами. Преобразовать массив таким образом, чтобы сначала располагались все элементы, модуль которых не превышает 1, а потом – все остальные.
Вариант 17. В одномерном динамическом массиве, состоящем из n вещественных элементов, вычислить: 1) максимальный по модулю элемент массива; 2) сумму элементов массива, расположенных между первым и вторым положительными элементами. Преобразовать массив таким образом, чтобы элементы, равные нулю, располагались после всех остальных.

Практические и лабораторные работы Тема 2.2 Документирование

Практическая работа № 1. Оформление документации на программные средства с использованием инструментальных средств

Цель практической работы: Разработать комплект документации на программное средство

Задание:

ПЗ на программное средство должна иметь следующую структуру: 1) Постановка задачи; 2) Входные и выходные данные; 3) Среда разработки и обоснование выбора языка программирования; 4) Описание алгоритма; 5) Описание используемых классов и методов; 6) Заключение. Приложение А – Техническое задание; Приложение Б – Технический проект; Приложение В – Иерархия функциональных диаграмм. Диаграмма сущность-связь. Диаграмма потоков данных Приложение Г UML-диаграммы Приложение Д Листинг программы Содержание отчета: ПЗ на электронном и бумажном носителе

Эталон ответа:

Техническое задание
1 Введение
Наименование работы: «Разработка системы автоматизации формирования сводной отчетности»

Техническое задание распространяется на разработку системы автоматизации. Работа выполняется в рамках курсовой работы по специальности 09.02.03 «Программирование в компьютерных системах».

2 Основания для разработки

1.1. Задание на курсовое проектирование, утвержденное заместителем директора филиала по учебно-методической работе 20.02.2023г.

1.2. Наименование работы: «Разработка системы автоматизации формирования сводной отчетности».

1.3. Исполнитель: Климова Анастасия Вадимовна.

1.4. Соисполнители: нет.

3 Назначения разработки

Программа будет использоваться в экономическом отделе. Предполагается, что программой будет пользоваться назначенный оператор, который будет загружать данные в систему, для дальнейшего формирования сводной ведомости.

4 Требования к программе или программному изделию

4.1 Требования к функциональным характеристикам.

Разрабатываемая система автоматизации должна обеспечивать:

- Сбор информации.

- Корректировку исходной информации

Автоматически формировать сводные ведомости.

Хранение вносимых данных в систему.

- Возможность печати сводных отчетов.

4.2. Требования к надежности.

Для обеспечения надёжность необходимо проверять корректность вносимой информации. Программа должна работать без сбоев в течение 5 дней по 8 часов.

Предусмотреть сохранение информации, восстановление системы после сбоев.

Автоматическое сохранение. Предусмотреть защиту данных, то есть пароль.

4.3. Условия эксплуатации.

Данный программный продукт предполагается эксплуатировать в условиях, благоприятных для оператора ЭВМ. Температура воздуха в помещении, где предполагается эксплуатировать систему, должна составлять примерно 20 – 25 С, влажность воздуха не должна превышать 85%, уровень шума не должен превышать 30 Дб. Предполагается, что обслуживать данный программный продукт будут всего два человека – назначенный оператор и начальник отдела. Назначенный оператор должен загружать в систему все данные и формировать ведомость, а начальник отдела – проверять правильность сформулированной ведомости.

4.4 Требования к составу и параметрам технических средств.

Требования к составу и параметрам технических средств уточняются на этапе эскизного проектирования.

4.5 Требования к информационной и программной совместимости.

Программа должна работать под управлением семейства операционных систем Windows 10. Так как, предприятие для которого разрабатывается система, перешло на операционную систему Windows 10.

4.6. Требования к маркировке и упаковке.

Готовое программное изделие и документация, поставляется на флеш -диске в стандартной упаковке. Специальных требований к маркировке не предъявляется.

4.7. Требования к транспортировке и хранению.

Программа поставляется на флеш - диске. Программная документация поставляется в электронном и печатном виде.

4.8. Специальные требования.

Специальных требований не прилагается.

5 Требования к программной документации

Требования к программной документации регулирует Единая Система Программной Документации (ЕСПД)

6 Технико-экономические показатели

Эффективность системы определяется удобством использования системы для контроля и управления основными параметрами материально-технических ресурсов.

Благодаря внедрению данной информационной системы на предприятие, экономия в год составит:

Экономия в год – 12458,25 (руб.)

7 Стадии и этапы разработки

Этапы разработки программного продукта:

- Код программы;
- Дизайн;
- Тестирование;
- Документирование;

Программные документы:

- Пояснительная записка;
- Описание программы

8 Порядок контроля и приемки

После передачи Исполнителем информационной системы Заказчику последний имеет право протестировать систему в течение 7 дней. После тестирования заказчик может принять работу или отправить на доработку. В случае если заказчика все устраивает, подписываются документы.

Эскизный проект

Введение

Объект разработки: Автоматизированная система формирования сводной отчетности.

Актуальность информационных систем в управлении предприятиями.

1 Общие сведения

1.1 Исследование предметной области

В основе решения задачи лежит автоматизация процесса. Работа предприятия связана с быстрой и качественной работой. При работе с информационной системой, система должны выполняться следующие действия:

- Сохранение внесенных таблиц в систему;
- Просмотр внесенных таблиц;
- Формирование сводной ведомости;

1.2 Описание Бизнес-процессов

Бизнес процесс «Заказчик/Исполнитель»

Заказчик смотрит товар. Если его все удовлетворяет, исполнитель выдает товар заказчику, он принимает его и подписывает все документы.

Диаграммы потоков данных представлены на рисунках 17-18.

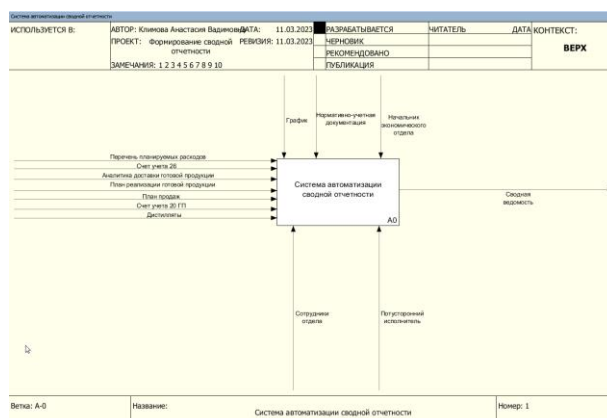


Рисунок 18 – Диаграмма потоков данных

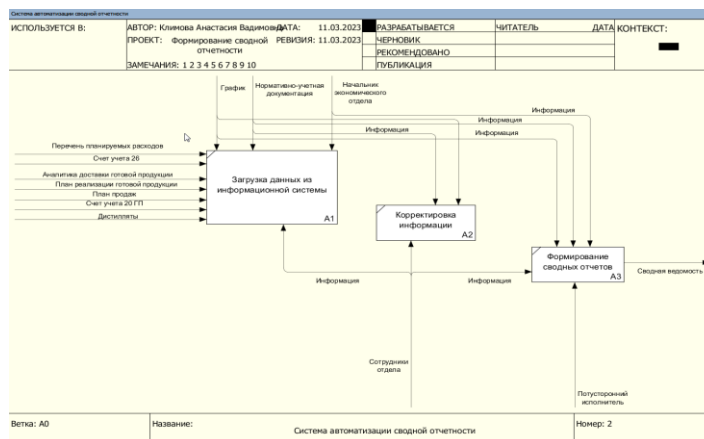


Рисунок 19 – Диаграмма потоков данных

2 Информационная система предприятия

2.1 Описание информационной системы

Функциональная структура информационной системы.

Функциональная структура включает в себя:

внесение, хранение и обработку информации;

Структура программных средств информационной системы.

Программное обеспечение - это комплекс программ обработки, в том числе и передача данных.

Программное обеспечение должно быть достаточным для выполнения всех функций, заданных в системе, а также имеет средства организации всех требуемых процессов обработки данных, позволяющих своевременно выполнять эти функции.

Для нормального функционирования системы необходимо наличие следующих программных средств:

- Операционная система Windows 10 или более новая;
- Microsoft SQL Server Management Studio 18.

Техническое обеспечение для данной автоматизированной системы можно разделить на:

серверную часть,

клиентскую часть.

В серверную часть входит один сервер с объемом жесткого диска не менее 500 Гб и оборудованный необходимым софтом.

Клиентская часть включает в себя одну рабочую станцию. В качестве примера, это может быть ноутбук с доступом интернет.

Связь между сервером и рабочей станцией обеспечивается через интернет соединение.

В итоге можно сказать, что техническое обеспечение данной автоматизированной системы состоит из сервера, рабочей станции и интернет сети.

Работа с данной автоматизированной системой не требует специальной подготовки пользователей, благодаря удобному интерфейсу.

Заключение

В процессе проведения курсовой работы был проведён анализ работы предприятия и на основе этого были разработаны документы для работы с системой.

Основной целью является автоматизация процесса.

Технический проект

Разработка технического проекта автоматизированной системы формирования сводной отчетности

1.1. Наименование проектируемой системы

Полное наименование системы – автоматизированная система формирования сводной отчетности.

1.2. Наименование документов, на основании которых ведется проектирование системы

Создание «Автоматизированной системы» осуществляется на основании требований и положений следующих документов:

- Техническое задание на создание «Автоматизированной системы»;
- Концепция создания «Автоматизированной системы»;

1.3. Перечень организаций, участвующих в разработке системы

Заказчиком создания «Автоматизированной системы» является предприятие АО «Кизлярский Коньячный Завод».

Находящийся по адресу: Россия, Республика Дагестан, город Кизляр, Орджоникидзе ул, д. 60.

Исполнителем работы по созданию «Автоматизированной системы» является разработчик Климова Анастасия Вадимовна.

Соисполнители: нет.

1.4. Цели, назначение и области использования системы

1.4.1. Цели создания системы

Создание системы направлено на повышение эффективности работы экономического отдела предприятия при формировании сводной ведомости.

Критерием оценки достижения поставленных целей является выполнение возложенных функций в установленные сроки.

1.4.2. Назначение системы

Система по виду автоматизированной деятельности относится к системам обработки и передачи информации.

Система предназначена для формирования сводной ведомости.

В систему можно вносить таблицы, просматривать их, и по ним формировать ведомость.

1.4.3. Области использования системы

Объектом автоматизации является деятельность экономического отдела предприятия при формировании сводной отчетности.

Внедрение системы будет осуществляться в условиях функционирования экономического отдела с использованием традиционной технологии.

Одной из основных задач создания системы является упрощение работы сотрудников экономического отдела при формировании сводной ведомости.

2 Основные технические решения

2.1. Решения по структуре системы

При проектировании системы использованы следующие основные принципы:

- Принцип системности, обеспечивающий целостность системы при ее взаимодействии с изменяющейся внешней средой и позволяющий объединять все подсистемы и ресурсы «Автоматизированной системы» на единой целевой, организационной, нормативной правовой, методической и программно-технической основе.

- Принцип иерархичности, состоящий в необходимости построения территориально-распределенных информационных ресурсов системы при формировании отчетов.

Информационный обмен между компонентами системы обеспечивается с использованием открытых технологий и стандартов информационного взаимодействия.

2.2. Решения по режимам функционирования и диагностированию работы системы

Для системы и ее компонентов предусмотрены два режима функционирования:

- Штатный, обеспечивающий функционирование компонентов системы в полном объеме;
- Регламентный, обеспечивающий проведение регламентных работ (без предоставления сервисов конечным пользователям системы);

3 Состав функций, реализуемых системой

Система предназначена для создания сводных отчетов на предприятии.

Модель рабочего процесса включает в себя:

- Описание функций работы системы;
- Последовательность работы системы;

3.1. Исполнение рабочих процессов системы

Система предназначена для автоматического формирования сводного отчета, а также для добавления таблиц в систему, просмотр внесенных таблиц.

Система должна обеспечивать выполнение следующих рабочих процессов:

1. Внесение таблиц;
2. Формирование сводной ведомости;

Для каждого из этапов должна быть представлена экранная форма с элементами управления.

3.2. Решения по программному обеспечению

В качестве платформы создания программного обеспечения системы используется язык программирования C# WPF.

Варианты:

3. Проектирование информационной системы для интернет-магазина продажи цветов
4. Проектирование информационной системы интернет-магазина видеокарт
5. Проектирование информационной системы интернет-магазина одежды
6. Проектирование информационной системы интернет-магазина открыток
7. Проектирование информационной системы контроля знаний
8. Проектирование информационной системы контроля оборудования для ЦДО «ПРИМЕР»
9. Проектирование информационной системы организации правильного питания
10. Проектирование информационной системы организации спортивного турнира
11. Проектирование информационной системы отдела трудоустройства выпускников ВУЗов
12. Проектирование информационной системы охраны и доступа на предприятии
13. Проектирование информационной системы планирования путешествий
14. Проектирование информационной системы по продаже атрибутики университета «Дубна»
15. Проектирование информационной системы поддержки чемпионата по футболу
16. Проектирование информационной системы поиска товаров в гипермаркетах
- Проектирование информационной системы проката автомобилей
17. Проектирование информационной системы профессионального сообщества учителей
18. Проектирование информационной системы распознавания нотной нотации
19. Проектирование информационной системы управления войсками
20. Проектирование информационной системы учебного заведения

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ ДАГЕСТАН

Частное профессиональное образовательное учреждение

«Республиканский полипрофессиональный колледж»

Вопросы для дифференцированного зачета

МДК 01.02 Поддержка и тестирование программных модулей

1. Тестирование как часть процесса верификации программного обеспечения.
2. Виды ошибок и методы отладки. Методы тестирования.
3. Классификация тестирования по уровням.
4. Тестирование производительности.
5. Регрессионное тестирование.
6. Средства разработки технической документации. Технологии разработки документов.
7. Документирование программного обеспечения в соответствии с Единой системой программной документации: основное.
8. Автоматизация разработки технической документации.
9. Автоматизированные средства оформления документации.

Критерии оценки:

Отметка «5»: ответ полный и правильный на основании изученных теорий; материал изложен в определенной логической последовательности, литературным языком. Ответ самостоятельный.

Отметка «4»: ответ полный и правильный на основании изученных теорий; материал изложен в определенной логической последовательности, при этом допущены две-три несущественные ошибки, исправленные по требованию преподавателя.

Отметка «3»: ответ полный, но при этом допущена существенная ошибка, или неполный, несвязный.

Отметка «2»: при ответе обнаружено непонимание обучающимся основного содержания учебного материала или допущены существенные ошибки, которые он не смог исправить при наводящих вопросах преподавателя.

Составители:

_____ Ж.А. Кадрышева

3.3. Задания для оценки освоения МДК 01.03 Разработка мобильных приложений

Практические и лабораторные работы
Практические и лабораторные работы Тема 3.1 Основные платформы и языки
разработки мобильных приложений

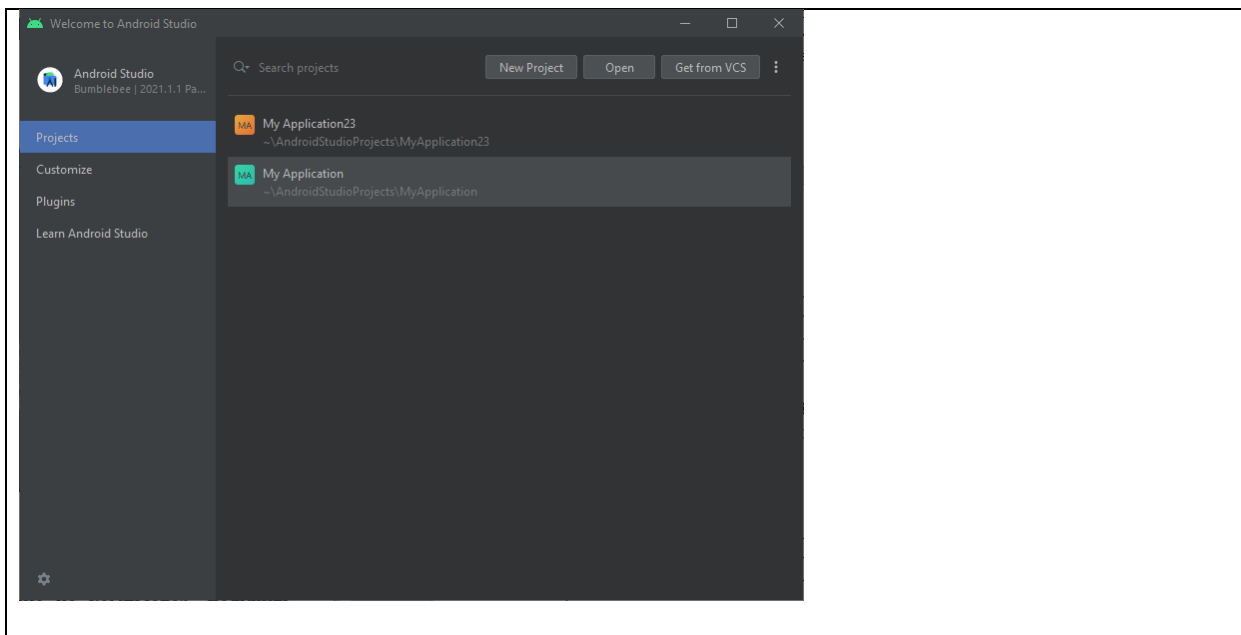
Практическая работа № 1. Установка инструментария и настройка среды для разработки
мобильных приложений

Цель практической работы: выполнить установку среды разработки и настроить ее для
работы

Задание:

Установка и настройка компонентов среды разработки Приложения для Android, как и большинство приложений для коммуникаторов, разрабатываются на стандартном ПК, где ресурсы не ограничены (по сравнению с мобильным устройством) и загружаются на целевой коммуникатор для отладки, тестирования и последующего использования. Приложения можно отлаживать и тестировать на реальном устройстве под управлением Android или на эмуляторе. Для первоначальной разработки и отладки удобнее использовать эмулятор, а затем выполнять окончательное тестирование на реальных устройствах.

Эталон ответа:



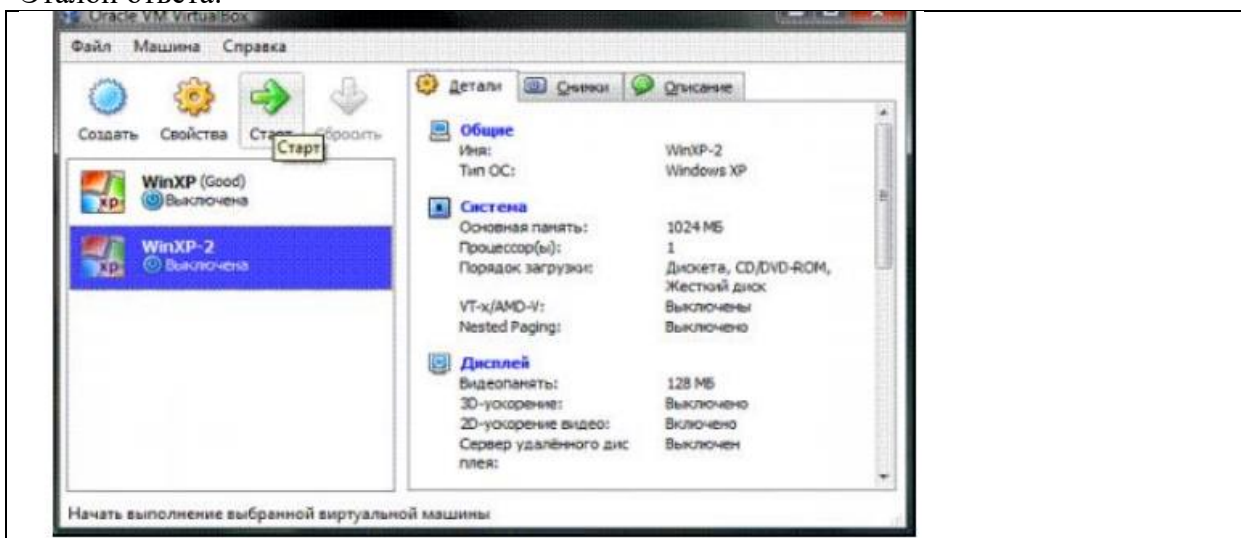
Практическая работа № 2. Установка среды разработки мобильных приложений с применением виртуальной машины

Цель практической работы: научиться устанавливать программное обеспечение на компьютер, получить практические навыки администрирования ОС

Задание:

Установка и настройка платформы виртуализации Oracle VM VirtualBox VirtualBox - программа абсолютно бесплатная и полностью на русском языке, что делает её очень привлекательной для использования как на домашнем, так и на рабочем компьютере. Впервые система была предоставлена в 2007 г. компанией InnoTek в двух вариантах – с открытым и закрытым исходными кодами, причем обе были бесплатны при условии некоммерческого использования. В 2008 г. платформа была перекуплена компанией Sun Microsystems, которая и занимается её разработкой в настоящее время.

Эталон ответа:



Практические и лабораторные работы Тема 3.2 Создание и тестирование модулей для мобильных приложений

Практическая работа № 1. Создание эмуляторов и подключение устройств»

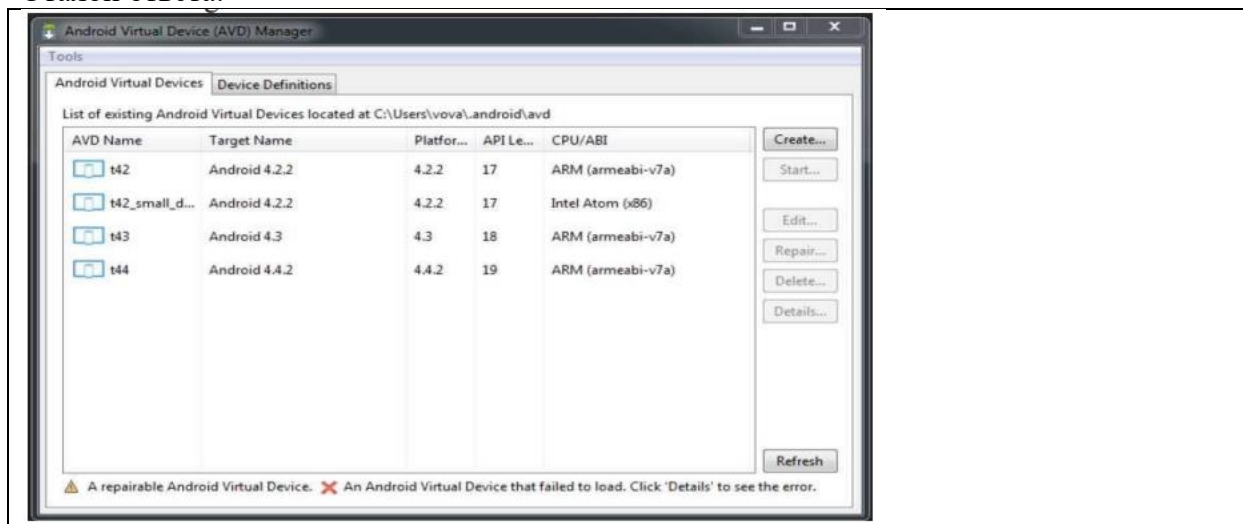
Цель практической работы: Знакомство с интерфейсом среды программирования.

Изучение структуры проекта

Задание:

Рассмотрите основные понятия Android проекта. Структура проекта – згс—«исходный код» приложения (java-классы) – assets —пустая директория. Может использоваться для сохранения raw-файлов. – gen—хранилище генерируемых системных файлов. В частности, здесь располагается файл R.java, в котором хранятся идентификаторы всех ресурсов, создаваемых в проекте (строковые ресурсы и т.п.). – libs—различные библиотеки, используемые приложением – res—ресурсы проекта. – AndroidManifest.xml— файл описания проекта (поддерживаемые версии SDK, версия приложения и т.п.) – project.properties—файл, включающий настройки проекта, такие как build target

Эталон ответа:



Варианты:

Установить AVD с различной конфигурацией

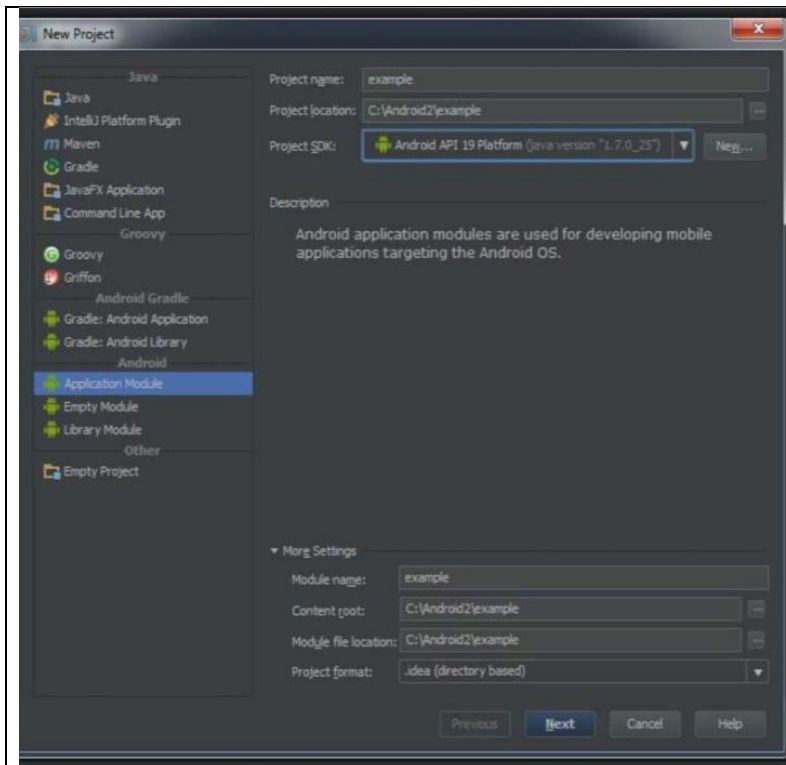
Практическая работа № 2. Настройка режима терминала»

Цель практической работы: : Знакомство с интерфейсом среды программирования.

Изучение структуры проекта

Задание: Рассмотрите основные понятия Android проекта. Структура проекта – згс— «исходный код» приложения (java-классы) – assets —пустая директория. Может использоваться для сохранения raw-файлов. – gen—хранилище генерируемых системных файлов. В частности, здесь располагается файл R.java, в котором хранятся идентификаторы всех ресурсов, создаваемых в проекте (строковые ресурсы и т.п.). – libs—различные библиотеки, используемые приложением – res—ресурсы проекта. – AndroidManifest.xml—файл описания проекта (поддерживаемые версии SDK, версия приложения и т.п.) – project.properties—файл, включающий настройки проекта, такие как build target

Эталон ответа:



Варианты:

Задание 2. Создание приложений с одним экраном (Activity) Необходимо создать два activity и организовать переход между ними. Содержимое Activity 1 кнопка с именем btn1

Содержимое Activity 2: TextView с текстом «Переданный параметр: значение параметра». значение параметра - принятый параметр из Activity 1.

При запуске приложения пользователь должен попадать на экран с Activity 1. После нажатия на кнопку btn1 необходимо осуществить переход к Activity 2 и передавать параметр из Activity 1. В качестве значения передаваемого параметра использовать свою фамилию.

Ключевые классы: Activity, Intent, Button, TextView.

Практическая работа № 3. Создание нового проекта»

Цель практической работы: Изучить работу Android приложения с базой данных.

Задание:

Необходимо создать приложение, взаимодействующее с базой данных. Первое активити должно содержать три кнопки. При нажатии на первую кнопку должно открываться новое активити, выводящее информацию из таблицы «Одногруппники» в удобном для восприятия формате.

При запуске приложения необходимо:

1. Создать БД, если ее не существует.
2. Создать таблицу «Одногруппники», содержащую поля:
3. ID;
4. ФИО;
5. Время добавления записи.
6. Удалять все записи из БД, а затем вносить 5 записей об одногруппниках. При нажатии на вторую кнопку необходимо внести еще одну запись в таблицу.

При нажатии на третью кнопку необходимо заменить ФИО в последней внесенной записи на Иванов Иван Иванович.

Эталон ответа:

```
Реализация метода onCreate public void onCreate( SQLiteDatabase db)
{ db.execSQL( "CREATE TABLE"+ TABLE_NAME + "( _id INTEGER PRIMARY
KEY AUTOINCREMENT ,"+ COL_NAME + " TEXT ,"+ COL_PHONE + " TEXT ) ; " ) ;
Реализация метода onUpgrade @ Override public void onUpgrade( SQLiteDatabase db,
int oldVersion, int newVersion) { db.execSQL( "DROP TABLE IF EXISTS"+ TABLE_NAME
) ; onCreate( db ) ;
}
Чтение из базы данных Cursor query( String table, String [ ] columns, String selection,
String [ ] selectionArgs, String groupBy, String having , String sortOrder)
```

Варианты:

Задание 1. Необходимо создать приложение, взаимодействующее с базой данных. Первое активити должно содержать три кнопки. При нажатии на первую кнопку должно открываться новое активити, выводящее информацию из таблицы «Одногруппники» в удобном для восприятия формате.

Создать новое отдельное приложение на основе приложения, созданного в части 1. Переопределить функцию onUpgrade. При изменении версии БД необходимо удалить таблицу

«Одногруппники», создать таблицу «Одногруппники» содержащую следующие поля:

1. ID;
2. Фамилия;
3. Имя;
4. Отчество;
5. Время добавления записи.

Изменить версию базы данных.

Задание 6.

1. Создайте новый Android проект ListViewSample.

2. В каталоге res/values создайте файл arrays.xml с данными полученными у преподавателя.

3. В каталоге res/layout создайте файл list_item.xml с данными полученными у преподавателя.

4. Модифицируйте метод onCreate вашей Активности:

```
@Override public void onCreate(Bundle savedInstanceState)
```

```
{ super.onCreate(savedInstanceState);
```

```
Resources r = getResources();
```

```
String[] stationsArray = r.getStringArray(R.array.stations);
```

```
ArrayAdapter<String> aa = new ArrayAdapter<String>(this, R.layout.list item,
```

```
stationsArray); setListAdapter(aa); ListView
```

```
lv = getListView(); }
```

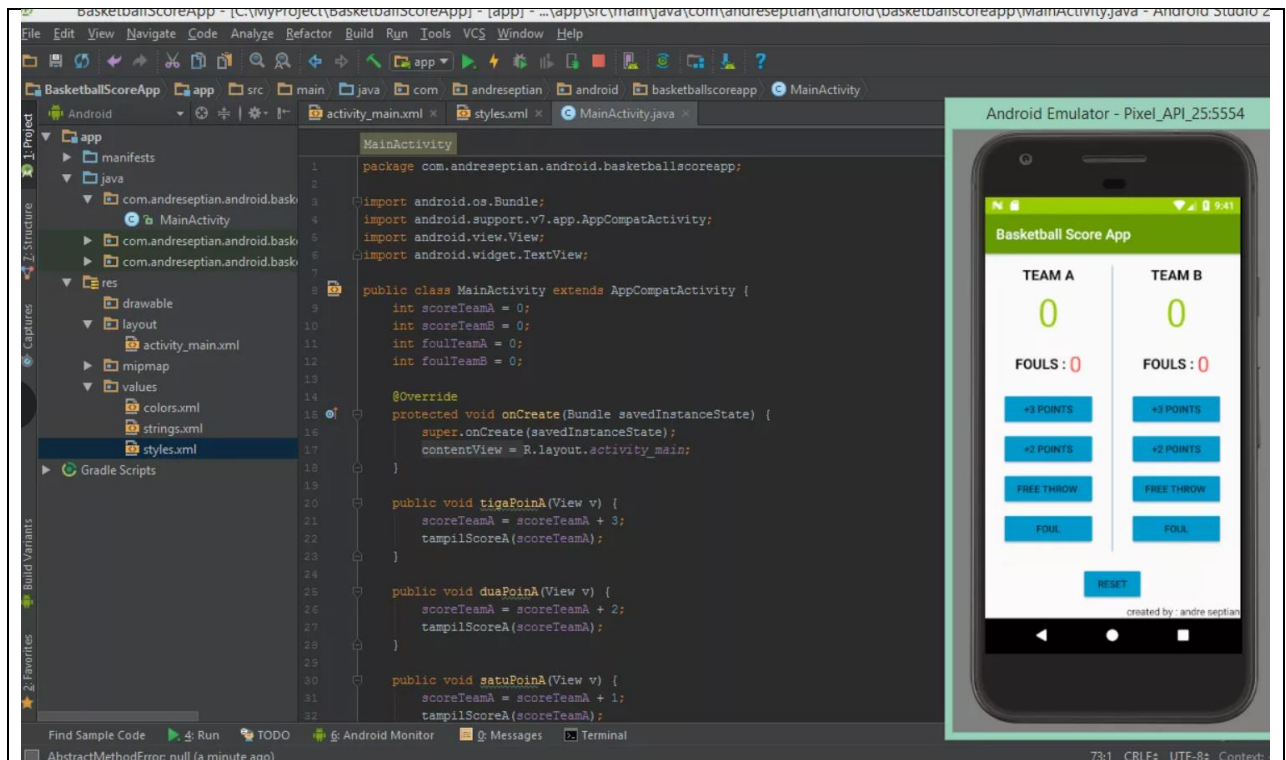
Практическая работа № 4. Изучение и комментирование кода»

Цель практической работы:

Задание:

Создайте новый Android проект ListViewSample. 2. В каталоге res/values создайте файл arrays.xml с данными полученными у преподавателя. 3. В каталоге res/layout создайте файл list_item.xml с данными полученными у преподавателя. 4. Модифицируйте метод onCreate вашей Активности: @Override public void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); Resources r = getResources(); String[] stationsArray = r.getStringArray(R.array.stations); ArrayAdapter aa = new ArrayAdapter(this, R.layout.list item, stationsArray); setListAdapter(aa); ListView lv = getListView(); }

Эталон ответа:



Варианты:

Задание 7. «Использование управляющих элементов в пользовательском интерфейсе»

Подготовка

1. Создайте новый проект ControlsSample.
2. Отредактируйте файл res/layout/main.xml так, чтобы остался только корневой элемент LinearLayout. В него в дальнейшем будут добавляться необходимые дочерние элементы:
 1. Использование графической кнопки Для использования изображения вместо текста на кнопке потребуются три изображения для трех состояний кнопки: обычное, выбранное («в фокусе») и нажатое. Все эти три изображения с соответствующими состояниями описываются в одном XML файле, который используется для создания такой кнопки.
 1. Скопируйте нужные изображения кнопки в каталог res/drawable-mdpi, для обновления списка содержимого каталога в Eclipse можно использовать кнопку F5.
 2. В этом же каталоге создайте файл smile_button.xml, описывающий, какие изображения в каких состояниях кнопки нужно использовать:
 3. Добавьте элемент Button внутри LinearLayout в файле разметки res/layout/main.xml:
 4. Обратите внимание на атрибут android:onClick= "onButtonClicked", указывающий, какой метод из Активности будет использоваться как обработчик нажатия на данную кнопку. Добавьте этот метод в Активность: `public void onButtonClicked(View v) { Toast.makeText (this, "Кнопка нажата", Toast.LENGTH_SHORT).show(); }`
 5. Запустите приложение и посмотрите, как изменяется изображение кнопки в разных состояниях, а также как функционирует обработчик нажатия на кнопку. Использование виджета CheckBox
 1. Добавьте элемент CheckBox внутри LinearLayout в файле разметки res/layout/main.xml:
 2. Атрибут android:onClick= "onCheckboxClicked" определяет, какой метод из Активности будет использоваться как обработчик нажатия на виджет. Добавьте этот метод в Активность: `public void onCheckboxClicked(View v) { if (((CheckBox) v).isChecked()) { Toast.makeText(this, "Отмечено", Toast.LENGTH_SHORT).show(); } else { Toast.makeText(this, "Не отмечено", Toast.LENGTH_SHORT).show(); } }`
 3. Запустите приложение и посмотрите на поведение чекбокса в разных ситуациях. Использование виджета ToggleButton

Данный виджет хорошо подходит в качестве

альтернативы радиокнопкам и чекбоксам, когда требуется переключаться между двумя взаимоисключающими состояниями, например, Включено/Выключено.

1. Добавьте элемент `ToggleButton` внутри `LinearLayout` в файле разметки `res/layout/main.xml`: 2. Атрибут `android:onClick="onToggleClicked"` определяет, какой метод ИЗ Активности будет использоваться как обработчик нажатия на виджет.

Добавьте этот метод в Активность: `public void onToggleClicked(View v) { if (((ToggleButton) v).isChecked()) { Toast.makeText(this, "Включено", Toast.LENGTH_SHORT).show(); } else { Toast.makeText(this, "Выключено", Toast.LENGTH_SHORT).show(); } }` 3. Запустите приложение и проверьте его функционирование

Практическая работа № 5. Лабораторная работа «Изменение элементов дизайна»

Цель практической работы:

Задание:

Использование виджета `RadioButton`. Радиокнопки используются для выбора между различными взаимоисключающими вариантами. Для создания группы радиокнопок используется элемент `RadioGroup`, внутри которого располагаются элементы `RadioButton`.

Эталон ответа:

```
<RadioGroup
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical" >
    <RadioButton
        android:id="@+id/radio_dog"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Собака" />
    <RadioButton
        android:id="@+id/radio_cat"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="onRadioButtonClicked"
        android:text="Кошечка" />
    <RadioButton
        android:id="@+id/radio_rabbit"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="onRadioButtonClicked"
        android:text="Кролик" />
</RadioGroup>
```

Варианты:

Использование виджета `EditText`

Использование виджета `ToggleButton`

Использование виджета `RadioButton`

Практическая работа № 6. Обработка событий: подсказки»

Цель практической работы: Создание обработчиков событий. Изучить работу с потоками. Научиться работать с мультимедиа файлами. Изучить работу с классом `AsyncTask`

Задание:

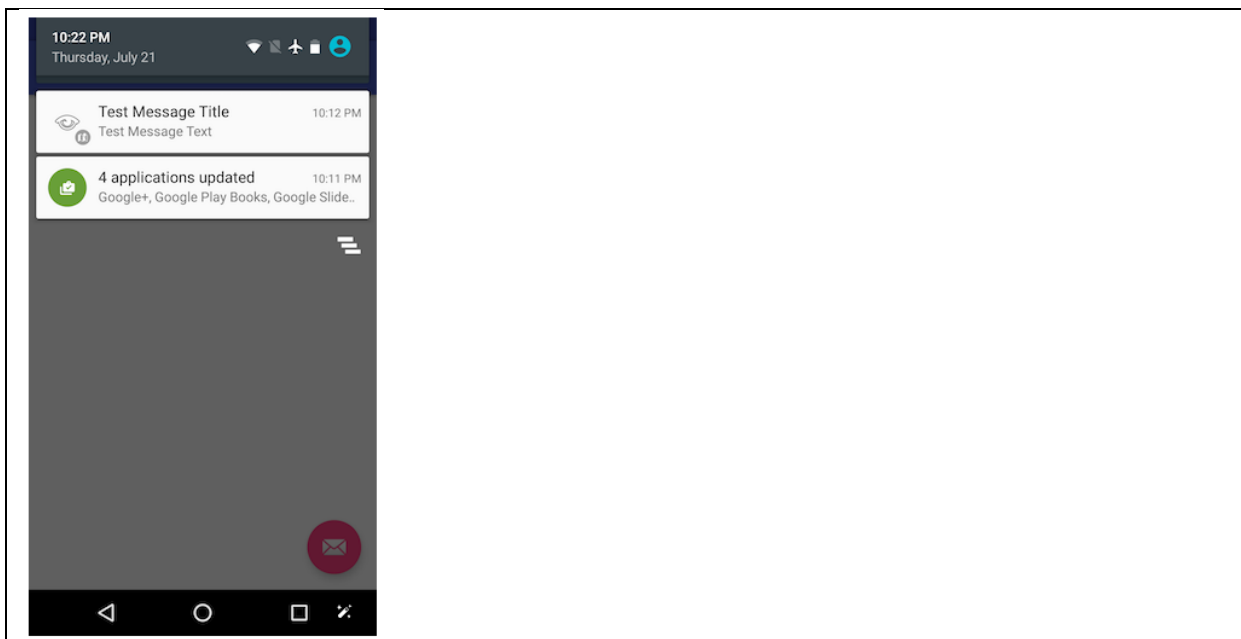
Разработать мобильное приложение, состоящее из четырех `activity`.

После запуска приложения пользователь должен попадать на экран с `activity`. На данном

экране должно быть представлено меню, состоящее из четырех кнопок. Высота кнопок должна составлять 20% от высоты экрана. Расстояние между кнопками - 2%. Первая и последняя кнопка должны быть на равном расстоянии от краев экрана. Ширина кнопок 75%, выравнивание посередине.

После нажатия на первую кнопку пользователь должен переходить к activity2, его внешний вид представлен на рисунке 64. Верстка должна осуществляться с использованием LinearLayout, ширина кнопок должна задаваться в процентах от ширины экрана.

Эталон ответа:



Варианты:

Задание. На основании изученных примеров разработать приложение сохраняющее статистику проигрываемых песен на радио Мегабайт. Для сохранения песни и названия необходимо создать базу данных, содержащую таблицу со следующими полями:

- ID
- Исполнитель
- Название трека
- Время внесения записи

Практическая работа № 7. Обработка событий: цветовая индикация

Цель практической работы: Создание обработчиков событий. Изучить работу с потоками.

Научиться работать с мультимедиа файлами.

Задание:

Для выполнения практической работы по обработке событий и цветовой индикации на Java в среде Android Studio, можно создать простую программу, которая меняет цвет фона окна при нажатии на кнопку.

Эталон ответа:



Варианты:

Реализуйте мобильное приложение различными вариациями цвета

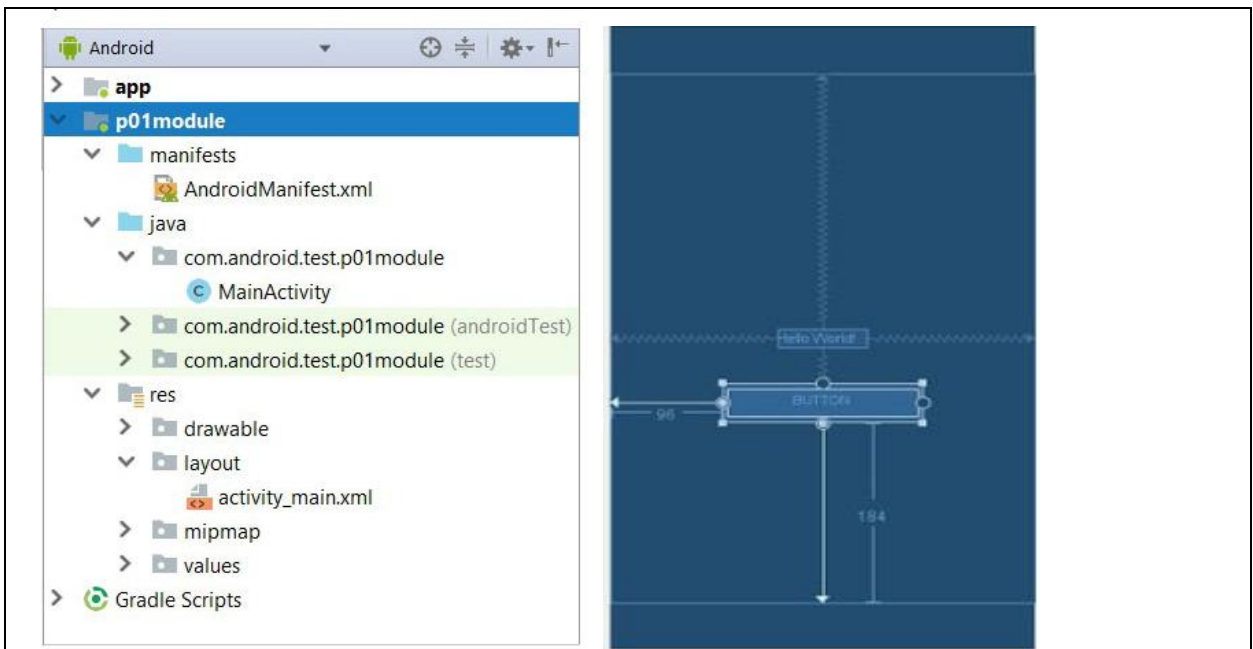
Практическая работа № 8. Подготовка стандартных модулей

Цель практической работы: получить навыки разработки модулей мобильного приложения

Задание:

Создайте новый модуль (приложения), добавьте кнопку и реализуйте обработки события на нажатие кнопки

Эталон ответа:



Варианты:

Создайте дополнительный библиотечный модуль

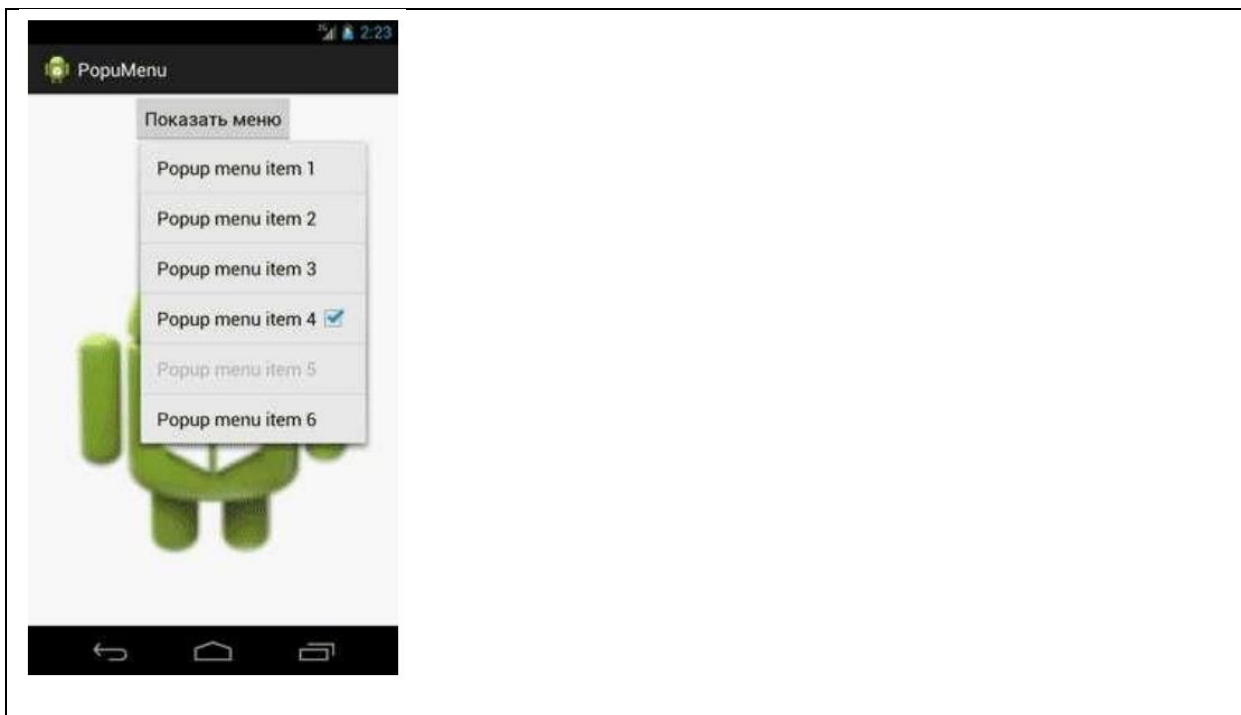
Практическая работа № 9. Обработка событий: переключение между экранами

Цель практической работы: получить навыки создания обработчиков событий для переключения между экранами

Задание:

Разработать PopUpMenu

Эталон ответа:



Варианты:

Необходимо подготовить меню, которое реализует сценарии для администраторов и пеших курьеров:

- Курьер
 - мои заказы
 - моя статистика
 - чат с поддержкой
 - данные заказа
 - маршрут и карта
- Администратор
 - аналогично страницам курьера, но имеет возможность просматривать данные всех курьеров

Необходимо подготовить меню, которое реализует сценарии для администраторов и пользователей.

- Пользователь
 - Мой аккаунт, баланс, личные данные
 - Мои поездки
 - Поддержка
- Администратор
 - Общая статистика
 - Автопарк (список машин, карточки, данные про тех обслуживание, бензин и тд)

Необходимо подготовить меню, которое реализует сценарии для администраторов и пользователей.

- Пользователь
 - Мой аккаунт, баланс, личные данные
 - Мои поездки

- Поддержка
- Администратор
- Общая статистика
- Автопарк (список машин, карточки, данные про тех обслуживание, бензин и тд)

Идея - сделать приложение, где пользователь под успокаивающую музыку собирает мозаики (+- как пазл, но только все кусочки квадратные).

- отображение сетки на поле и бледной подсветки будущего изображения)
- drag n drop кусочков на поле
- счетчик правильно уложенных кусочков
- Несколько предустановленных изображений для сбора, разного уровня сложности
- Несколько предустановленных расслабляющих мелодий -

Идея - пользователь ложится, кладет устройство себе на живот, приложение анализирует паттерны дыхания и в игровой форме направляет пользователя на корректное выполнение упражнений.

- Примитивная обработка данных акселерометра и гироскопа
- Обратная связь от приложения в виде мелодии / вибрации
- Кастомизируемый список упражнений
- Система баллов за выполнение тренировок, уровни и ачивки
- Напоминания
- Экран достижений / истории тренировок

Практическая работа № 10. Передача данных между модулями

Цель практической работы: получить навык применения метода передачи данных между модулями

Задание:

Провести тестирование, выполнить оптимизацию системы и составить отчет

Эталон ответа:

```
package com.example.viewapp;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
    public void onClick(View v) {

        EditText nameText = findViewById(R.id.name);
        EditText companyText = findViewById(R.id.company);
```



```

EditText ageText = findViewById(R.id.age);

String name = nameText.getText().toString();
String company = companyText.getText().toString();
int age = Integer.parseInt(ageText.getText().toString());

Intent intent = new Intent(this, SecondActivity.class);
intent.putExtra("name", name);
intent.putExtra("company", company);
intent.putExtra("age", age);
startActivity(intent);
}}

```

Варианты:

Необходимо подготовить меню, которое реализует сценарии для администраторов и пеших курьеров:

- Курьер
- мои заказы
- моя статистика
- чат с поддержкой
- данные заказа
- маршрут и карта
- Администратор
- аналогично страницам курьера, но имеет возможность просматривать данные всех курьеров

Необходимо подготовить меню, которое реализует сценарии для администраторов и пользователей.

- Пользователь
- Мой аккаунт, баланс, личные данные
- Мои поездки
- Поддержка
- Администратор
- Общая статистика
- Автопарк (список машин, карточки, данные про тех обслуживание, бензин и тд)

Необходимо подготовить меню, которое реализует сценарии для администраторов и пользователей.

- Пользователь
- Мой аккаунт, баланс, личные данные
- Мои поездки
- Поддержка
- Администратор
- Общая статистика
- Автопарк (список машин, карточки, данные про тех обслуживание, бензин и тд)

Идея - сделать приложение, где пользователь под успокаивающую музыку собирает мозаики (+- как пазл, но только все кусочки квадратные).

1. отображение сетки на поле и бледной подсветки будущего изображения)
2. drag n drop кусочков на поле
3. счетчик правильно уложенных кусочков
4. Несколько предустановленных изображений для сбора, разного уровня

сложности

5. Несколько предустановленных расслабляющих мелодий -

Идея - пользователь ложиться, кладет устройство себе на живот, приложение анализирует паттерны дыхания и в игровой форме направляет пользователя на корректное выполнение упражнений.

- Примитивная обработка данных акселерометра и гироскопа
- Обратная связь от приложения в виде мелодии / вибрации
- Кастомизируемый список упражнений
- Система баллов за выполнение тренировок, уровни и ачивки
- Напоминания
- Экран достижений / истории тренировок

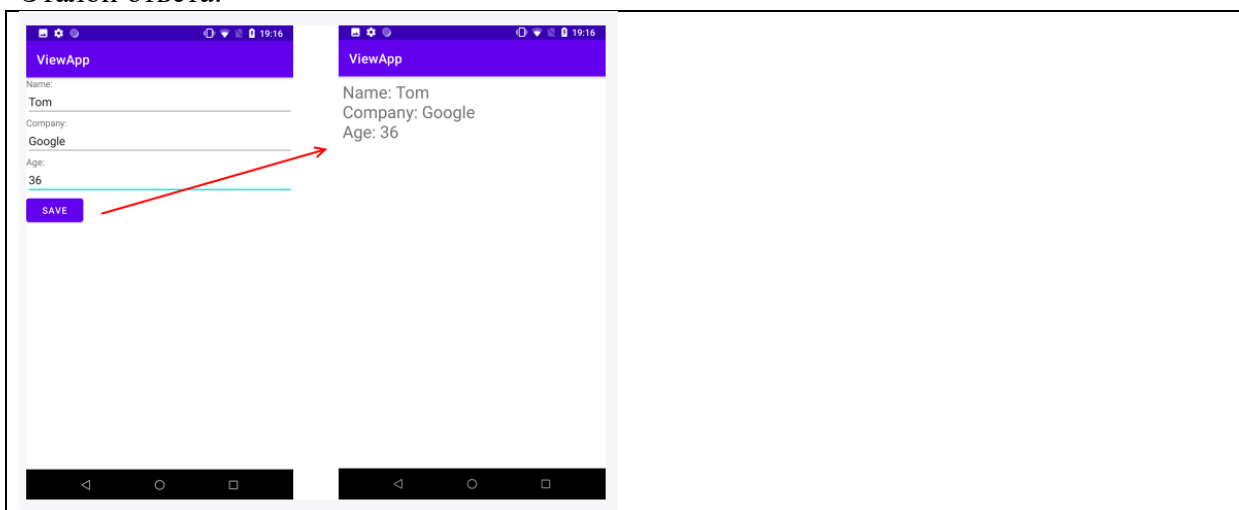
Практическая работа № 11. Тестирование и оптимизация мобильного приложения

Цель практической работы: Провести тестирование созданного приложения и выполнить его оптимизацию

Задание:

Провести тестирование, выполнить оптимизацию системы и составить отчет

Эталон ответа:



Варианты:

Необходимо подготовить меню, которое реализует сценарии для администраторов и пеших курьеров:

- Курьер
- мои заказы
- моя статистика
- чат с поддержкой
- данные заказа
- маршрут и карта
- Администратор
- аналогично страницам курьера, но имеет возможность просматривать данные всех курьеров

Необходимо подготовить меню, которое реализует сценарии для администраторов и пользователей.

- Пользователь

<ul style="list-style-type: none"> ▪ Мой аккаунт, баланс, личные данные ▪ Мои поездки ▪ Поддержка ▪ Администратор ▪ Общая статистика ▪ Автопарк (список машин, карточки, данные про тех обслуживание, бензин и тд)
<p>Необходимо подготовить меню, которое реализует сценарии для администраторов и пользователей.</p> <ul style="list-style-type: none"> ▪ Пользователь ▪ Мой аккаунт, баланс, личные данные ▪ Мои поездки ▪ Поддержка ▪ Администратор ▪ Общая статистика ▪ Автопарк (список машин, карточки, данные про тех обслуживание, бензин и тд)
<p>Идея - сделать приложение, где пользователь под успокаивающую музыку собирает мозаики (+- как пазл, но только все кусочки квадратные).</p> <ol style="list-style-type: none"> 6. отображение сетки на поле и бледной подсветки будущего изображения) 7. drag n drop кусочков на поле 8. счетчик правильно уложенных кусочков 9. Несколько предустановленных изображений для сбора, разного уровня сложности 10. Несколько предустановленных расслабляющих мелодий -
<p>Идея - пользователь ложиться, кладет устройство себе на живот, приложение анализирует паттерны дыхания и в игровой форме направляет пользователя на корректное выполнение упражнений.</p> <ul style="list-style-type: none"> ▪ Примитивная обработка данных акселерометра и гироскопа ▪ Обратная связь от приложения в виде мелодии / вибрации ▪ Кастомизируемый список упражнений ▪ Система баллов за выполнение тренировок, уровни и ачивки ▪ Напоминания ▪ Экран достижений / истории тренировок

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ ДАГЕСТАН
Частное профессиональное образовательное учреждение
«Республиканский полипрофессиональный колледж»

Темы контрольных работ
МДК 01.03 Разработка мобильных приложений

1.Форма предоставления контрольной работы

Требования к структуре и оформлению контрольной работы:

Контрольная работа оформляется на электронном и бумажном носителях.

– текстовые документы представляются в формате Word 2003 (doc.);

– параметры текстового редактора: поля – верхнее, нижнее – 2.0 см, левое – 3.0 см, правое

– 1 см, шрифт Times New Roman, высота 14, межстрочный интервал полуторный, выравнивание по ширине, красная строка 1,25 см;

- документы, содержащие подписи и печати, сканируются в формате JPG или PDF, отсканированный текст, подписи и печати должны читаться без затруднений в масштабе 1:1;
- в текстах не допускается сокращение названий и наименований;
- все страницы нумеруются (нумерация начинается с титульного листа, номер на первой странице не ставится), номер страницы ставится в центре нижней части страницы, в колонтитуле на каждой странице указываются фамилия, имя и отчество студента.

Структура работы

1. Титульный лист
2. Содержание
3. Пункт 1
4. Подпункт 1.1
5.
6. Пункт 2
7. Подпункт 2.1
8.
9. Список использованных источников

Тематика контрольных работ

1. Акселерометр и служба определения местоположения, вторичные потоки выполнения, обработка асинхронных операций и доступ к Веб-сервисам.
2. Особенности использования pivot и ranogama.
3. Краткая история ОС Android.
4. Intel для Android: партнерство и инструментарий разработчика.
5. Архитектура приложений для Android. Ресурсы приложения. Пользовательский интерфейс. Инструментарий разработки приложений для Android.
6. Обзор шагов разработки типичного приложения под Android. Особенности разработки с использованием эмулятора. Отладка кода в эмуляторе и на реальных приложениях. Пример простейших программ Android-приложения. Запуск приложения на эмуляторе.
7. Тестирование приложения с помощью Dalvik Debug Monitor Server (DDMS).
8. Планирование покадровой анимации, анимирование, анимация шаблонов, видов, использование класса Camera.
9. Проверка безопасности, работа со службами, основанными на местоположении, использование HTTP-служб, службы AIDL.
10. Подготовка AndroidManifest.xml для загрузки, локализация приложения, подготовка ярлыка приложения, подготовка APK-файла для загрузки, работа пользователя с Android Market.

Критерии оценки:

- отметка «5»: Задание выполнено в полном объеме с соблюдением необходимой последовательности. Студент работал полностью самостоятельно.
- отметка «4»: Практическое задание выполнено студентом в полном объеме и самостоятельно. Допускается отклонение от необходимой последовательности выполнения, не влияющее на правильность конечного результата. Допускаются неточности и небрежность в оформлении результатов задания.
- отметка «3»: Практическое задание выполнено и оформлено студентом с помощью преподавателя или хорошо подготовленных и уже выполнивших на «отлично» данную работу студентов. На выполнение задания затрачено много времени.

отметка «2»: Выставляется в том случае, когда студент оказался неподготовленным к выполнению задания. Полученные результаты не позволяют сделать правильных выводов и полностью расходятся с поставленной целью. Обнаружено плохое знание теоретического материала и отсутствие необходимых умений. Руководство и помощь со стороны преподавателя неэффективны из-за плохой подготовки студента.

Составители:

_____ Ж.А. Кадрышева

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ ДАГЕСТАН
Частное профессиональное образовательное учреждение
«Республиканский полипрофессиональный колледж»

Вопросы к экзамену МДК 01.03 Разработка мобильных приложений

Теоретические вопросы:

1. Основные платформы мобильных приложений, сравнительная характеристика
2. Нативные приложения, веб -приложения, гибридные и кроссплатформенные приложения, их области применения
3. Нативный iOS
4. Нативный Android
5. Нативный Windows UWP
6. Основные языки для разработки мобильных приложений (Java, Objective -C и др.)
7. Инструменты разработки мобильных приложений (JDK/ AndroidStudio/ WebView/ Phonegap и др.)
8. Инструментарий среды разработки мобильных приложений
9. Структура типичного мобильного приложения
10. Элементы управления и контейнеры
11. Работа со списками
12. Способы хранения данных
13. Архитектуры кроссплатформенных фреймворков. Qt
14. Архитектуры кроссплатформенных фреймворков. Flutter
15. Архитектуры кроссплатформенных фреймворков. Xamarin
16. Архитектуры кроссплатформенных фреймворков. Xamarin.Forms
17. Процесс разработки и документация. Первичная документация
18. Экраны, данные и логика мобильного приложения
19. Архитектура приложения. Многослойный MVVM
20. Введение в программирование на Java. Основные сведения о языке программирования
21. Реализация принципов ООП в Java.
22. Правила именования и лексика языка Java.
23. Реализация наследования в программах на Java.
24. Паттерны проектирования программ.

Задачи:

1. Разработать приложение «Домашняя библиотека» с пользовательским графическим интерфейсом по предложенному заданию. В разработанных приложениях интерфейс пользователя должен использовать таблицы GridView для представления данных на

экране, а также другие необходимы виджеты. Для хранения данных и запросов использовать СУБД SQLite.

2. Разработать приложение «Домашняя бухгалтерия» с пользовательским графическим интерфейсом по предложенному заданию. В разработанных приложениях интерфейс пользователя должен использовать таблицы GridView для представления данных на экране, а также другие необходимы виджеты. Для хранения данных и запросов использовать СУБД SQLite.

3. Разработать приложение «Участники спортивных соревнований» с пользовательским графическим интерфейсом по предложенному заданию. В разработанных приложениях интерфейс пользователя должен использовать таблицы GridView для представления данных на экране, а также другие необходимы виджеты. Для хранения данных и запросов использовать СУБД SQLite.

4. Разработать приложение «Результаты аттестации студенческой группы» с пользовательским графическим интерфейсом по предложенному заданию. В разработанных приложениях интерфейс пользователя должен использовать таблицы GridView для представления данных на экране, а также другие необходимы виджеты. Для хранения данных и запросов использовать СУБД SQLite.

5. Разработать приложение «Информационная система салона магазина по продаже мобильных устройств» с пользовательским графическим интерфейсом по предложенному заданию. В разработанных приложениях интерфейс пользователя должен использовать таблицы GridView для представления данных на экране, а также другие необходимы виджеты. Для хранения данных и запросов использовать СУБД SQLite.

6. Разработать приложение «Информационная система учета услуг в фитнес-центре» с пользовательским графическим интерфейсом по предложенному заданию. В разработанных приложениях интерфейс пользователя должен использовать таблицы GridView для представления данных на экране, а также другие необходимы виджеты. Для хранения данных и запросов использовать СУБД SQLite.

7. Разработать приложение «Информационная система фестиваля художественной самодеятельности в университете» с пользовательским графическим интерфейсом по предложенному заданию. В разработанных приложениях интерфейс пользователя должен использовать таблицы GridView для представления данных на экране, а также другие необходимы виджеты. Для хранения данных и запросов использовать СУБД SQLite.

8. Разработать приложение «Информационная система учета работ в автомобильной мастерской» с пользовательским графическим интерфейсом по предложенному заданию. В разработанных приложениях интерфейс пользователя должен использовать таблицы GridView для представления данных на экране, а также другие необходимы виджеты. Для хранения данных и запросов использовать СУБД SQLite.

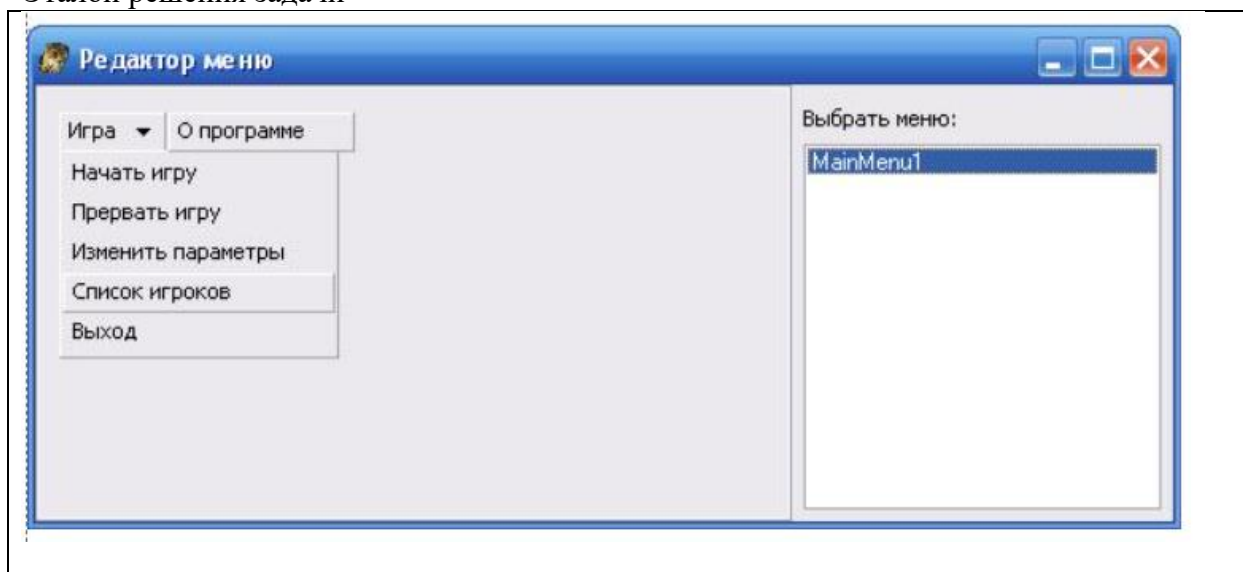
9. Разработать приложение «Информационная система подведения итогов спортивных соревнований по легкой атлетике» с пользовательским графическим интерфейсом по предложенному заданию. В разработанных приложениях интерфейс пользователя должен использовать таблицы GridView для представления данных на экране, а также другие необходимы виджеты. Для хранения данных и запросов использовать СУБД SQLite.

10. Разработать приложение «Информационная система агентства по продаже недвижимости» с пользовательским графическим интерфейсом по предложенному заданию. В разработанных приложениях интерфейс пользователя должен использовать таблицы GridView для представления данных на экране, а также другие необходимые виджеты. Для хранения данных и запросов использовать СУБД SQLite.

Эталонные ответы:

Основные платформы мобильных приложений, сравнительная характеристика
Нативные приложения, веб-приложения, гибридные и кроссплатформенные приложения, их области применения
Нативный iOS
Нативный Android
Нативный Windows UWP
Основные языки для разработки мобильных приложений (Java, Objective-C и др.)
Инструменты разработки мобильных приложений (JDK/ AndroidStudio/ WebView/ Phonegap и др.)
Инструментарий среды разработки мобильных приложений
Структура типичного мобильного приложения
Элементы управления и контейнеры
Работа со списками
Способы хранения данных
Архитектуры кроссплатформенных фреймворков. Qt
Архитектуры кроссплатформенных фреймворков. Flutter
Архитектуры кроссплатформенных фреймворков. Xamarin
Архитектуры кроссплатформенных фреймворков. Xamarin.Forms
Процесс разработки и документация. Первичная документация
Экраны, данные и логика мобильного приложения
Архитектура приложения. Многослойный MVVM
Введение в программирование на Java. Основные сведения о языке программирования
Реализация принципов ООП в Java.
Правила именования и лексика языка Java.
Реализация наследования в программах на Java.
Паттерны проектирования программ.

Эталонное решение задачи



Самостоятельная работа при изучении МДК 01.03 Разработка мобильных приложений

Самостоятельная работа 3.1 Создать приложение, по нажатию кнопки в котором проигрывается какой-то звук.

1. Порядок выполнения

1. Изучить рекомендуемую литературу. 2. Разработать приложение, которое предоставляет пользователю возможность выбора рода деятельности: работа с камерой для создания снимков; воспроизведение аудио и видео; просмотр изображений. 3. Разработать приложение, в котором реализовать четыре активности: главная активность, предназначена для выбора рода деятельности, содержит три кнопки, нажатие на каждую кнопку вызывает к жизни соответствующую активность; активность для работы с камерой и создания снимков; активность для воспроизведения аудио и видео; активность для просмотра изображений. 4. Разработать приложения, получающего координаты устройства и отслеживающего их изменение 5. Ответить на контрольные вопросы. 6. Оформить отчет.

2. Контрольные вопросы

1. Как настроить интерфейс и реализовать логику активности для работы с камерой. 2. Как настроить интерфейс и реализовать логику активности для воспроизведения аудио и видео. 3. Как настроить интерфейс и реализовать логику активности для просмотра изображений. 4. Как настроить интерфейс и реализовать логику главной активности приложения. 5. Технология разработки приложения, получающего координаты устройства и отслеживающего их изменение 6. Каким образом имитировать передачу данных о местоположении на эмулятор.

3. Варианты заданий

Создать приложение, которое рисует движение солнца на фоне неба. Снизу изобразить зеленую поляну. 2. Создать приложение, которое рисует движение ядра, которое выстреливается из пушки под углом α и скоростью V_0 . Изобразить стену, в которую может попасть ядро. Траектория движения ядра должна быть построена с учетом силы тяготения Земли. 3. Создать приложение, которое рисует движение автомобиля по дороге на фоне неба. 4. Создать приложение, которое рисует движение человека по дороге на фоне домов. 5. Создать приложение, которое рисует движение автомобиля через перекресток. 6. Создать приложение, которое рисует движение двух автомобилей через перекресток вдоль одного направления. 7. Создать приложение, которое рисует движение двух автомобилей через перекресток в перпендикулярном направлении. 8. Создать приложение, которое рисует движение Колобка по зеленой поляне к домику. 9. Создать приложение, которое рисует полет самолета на фоне неба с облаком. 10. Создать приложение, которое выполняет вывод на экран изображения с камеры и сохраняет его в файле 11. Создать приложение, которое выполняет вывод на экран изображения с камеры и сохраняет в файле выделенную область. 12. Создать приложение, которое выполняет геолокацию с помощью сервисов Google, используя объект Location

4. Содержание отчета

1.Название и цель работы. 2.Листинг созданного приложения в Android Studio с использованием элементов графики и анимации. 3.Результаты работы приложения в эмуляторе телефона. 4.Выводы по проделанной работе.

Самостоятельная работа 3.2 Создать приложение, при запуске которого проигрывается какое-то видео.

1. Порядок выполнения

1. Изучить рекомендуемую литературу.
2. Разработать приложение, которое предоставляет пользователю возможность выбора рода деятельности: работа с камерой для создания снимков; воспроизведение аудио и видео; просмотр изображений.
3. Разработать приложение, в котором реализовать четыре активности: главная активность, предназначена для выбора рода деятельности, содержит три кнопки, нажатие на каждую кнопку вызывает к жизни соответствующую активность; активность для работы с камерой и создания снимков; активность для воспроизведения аудио и видео; активность для просмотра изображений.
4. Разработать приложения, получающего координаты устройства и отслеживающего их изменение
5. Ответить на контрольные вопросы.
6. Оформить отчет.

2. Контрольные вопросы

1. Как настроить интерфейс и реализовать логику активности для работы с камерой.
2. Как настроить интерфейс и реализовать логику активности для воспроизведения аудио и видео.
3. Как настроить интерфейс и реализовать логику активности для просмотра изображений.
4. Как настроить интерфейс и реализовать логику главной активности приложения.
5. Технология разработки приложения, получающего координаты устройства и отслеживающего их изменение
6. Каким образом имитировать передачу данных о местоположении на эмулятор.

3. Варианты заданий

Создать приложение, которое рисует движение солнца на фоне неба. Снизу изобразить зеленую поляну.

2. Создать приложение, которое рисует движение ядра, которое выстреливается из пушки под углом α и скоростью V_0 . Изобразить стену, в которую может попасть ядро. Траектория движения ядра должна быть построена с учетом силы тяготения Земли.
3. Создать приложение, которое рисует движение автомобиля по дороге на фоне неба.
4. Создать приложение, которое рисует движение человека по дороге на фоне домов.
5. Создать приложение, которое рисует движение автомобиля через перекресток.
6. Создать приложение, которое рисует движение двух автомобилей через перекресток вдоль одного направления. 7. Создать приложение, которое рисует движение двух автомобилей через перекресток в перпендикулярном направлении. 8. Создать приложение, которое рисует движение Колобка по зеленой поляне к домику. 9. Создать приложение, которое рисует полет самолета на фоне неба с облаком. 10. Создать приложение, которое выполняет вывод на экран изображения с камеры и сохраняет его в файле 11. Создать приложение, которое выполняет вывод на экран изображения с камеры и сохраняет в файле выделенную область. 12. Создать приложение, которое выполнят геолокацию с помощью сервисов Google, используя объект Location

4. Содержание отчета

1. Название и цель работы. 2. Листинг созданного приложения в Android Studio с использованием элементов графики и анимации. 3. Результаты работы приложения в эмуляторе телефона. 4. Выводы по проделанной работе.

Самостоятельная работа 3.3 Создать приложение, при запуске которого активируется фотокамера телефона, производится снимок, и этот снимок помещается в ImageView интерфейса приложения.

1. Порядок выполнения

- 1.Изучить методические указания к практическому занятию.
 - 2.Запустить на выполнение Android Studio.
 - 3.Если будет использоваться внешний эмулятор для запуска Android-приложений (например, BlueStacks или Genymotion), загрузить его.
 - 4.Создать в среде Android Studio проект, разработанный на практическом занятии.
 - 5.Создать в соответствующих директориях проекта файлы grass.xml, sky.xml, sun.xml.
 - 6.Добавить в проект графические изображения неподвижных и движущихся изображений.
 - 7.Создать новую папку res/anim и добавить в нее файл sunrise.xml.
 - 8.В окне java-кода проекта добавить строки для получения анимированного изображения.
 - 9.Запустить созданное приложение в эмуляторе Android и наблюдать за появлением этого приложения и результатов его работы в окне приложений эмулятора.
- Результаты выполнения практического задания запишите в отчет.

2. Контрольные вопросы

1. Каковы этапы жизненного цикла служб?
2. Как создать службу в Android-приложении?
3. Какие существуют способы вызова службы?
4. Как зарегистрировать службу в файле манифеста приложения?
5. Приведите пример создания анимации в XML и в коде.

3. Варианты заданий

- Создать приложение, которое рисует движение солнца на фоне неба. Снизу изобразить зеленую поляну.
2. Создать приложение, которое рисует движение ядра, которое выстреливается из пушки под углом α и скоростью V_0 . Изобразить стену, в которую может попасть ядро. Траектория движения ядра должна быть построена с учетом силы тяготения Земли.
 3. Создать приложение, которое рисует движение автомобиля по дороге на фоне неба.
 4. Создать приложение, которое рисует движение человека по дороге на фоне домов.
 5. Создать приложение, которое рисует движение автомобиля через перекресток.
 6. Создать приложение, которое рисует движение двух автомобилей через перекресток вдоль одного направления.
 7. Создать приложение, которое рисует движение двух автомобилей через перекресток в перпендикулярном направлении.
 8. Создать приложение, которое рисует движение Колобка по зеленой поляне к домику.
 9. Создать приложение, которое рисует полет самолета на фоне неба с облаком.
 10. Создать приложение, которое выполняет вывод на экран изображения с камеры и сохраняет его в файле
 11. Создать приложение, которое выполняет вывод на экран изображения с камеры и сохраняет в файле выделенную область.
 12. Создать приложение, которое выполняет геолокацию с помощью сервисов Google, используя объект Location

4. Содержание отчета

- 1.Название и цель работы.
- 2.Листинг созданного приложения в Android Studio с использованием элементов графики и анимации.
- 3.Результаты работы приложения в эмуляторе телефона.
- 4.Выводы по проделанной работе.

Критерии оценки:

оценки «хорошо» заслуживает студент, обнаруживший полные знания учебно-программного материала, успешно выполняющий предусмотренные в программе задания, усвоивший основную литературу, рекомендованную в программе. Как правило, оценка «хорошо» выставляется студентам, показавшим систематический характер знаний по дисциплине и способным к их самостоятельному пополнению и обновлению в ходе дальнейшей учебной работы и профессиональной деятельности;

оценки «удовлетворительно» заслуживает студент, обнаруживший знание учебно-программного материала в объеме, необходимом для дальнейшей учебы и предстоящей работы по профессии, справляющийся с выполнением заданий, предусмотренных программой, знакомый с основной литературой, рекомендованной программой. Как правило, оценка «удовлетворительно» выставляется студентам, допустившим погрешность в ответе на экзамене и при выполнении экзаменационных заданий, но обладающим необходимыми знаниями для их устранения под руководством преподавателя;

оценка «неудовлетворительно» выставляется студенту, обнаружившему пробелы в знаниях основного учебно-программного материала, допустившему принципиальные ошибки в выполнении предусмотренных программой заданий. Как правило, оценка «неудовлетворительно» ставится студентам, которые не могут продолжить обучение или приступить к профессиональной деятельности по окончании вуза без дополнительных занятий по соответствующей дисциплине».

Составители:

_____ Ж.А. Кадрышева

3.4 Задания для оценки освоения МДК 01.04 Системное программирование

Практические и лабораторные работы Тема 4.1 Программирование на языке
низкого уровня

Практическая работа № 1. Использование потоков.

Цель практической работы: : изучить основы создания и управления потоками в ОС Windows.

Задание:

Разработать многопоточное Win32-приложение, которое использует диалоговое окно для управления потоками процессов. Дизайн диалогового окна задается вариантом, преподавателем или самостоятельно. Для визуализации работы потоков использовать соответствующие элементы управления диалога, графику. Приложение должно содержать три потока. Предусмотреть вывод системной информации о потоках (например полей CONTEXT , временны параметры и т.д.). В отчете привести диаграмму состояния потоков, копии окон

Эталон ответа:

```
Создание потока:  
hThread2 = CreateThread(NULL, 0, Second_Thread, hDlg, CREATE_SUSPENDED, &Id2 );  
Назначение потоку относительного приоритета:  
SetThreadPriority(hThread2, prir);  
Возобновление потока: ResumeThread(hThread1);  
Приостановка потока: SuspendThread(hThread1);  
Получение времени создания, окончания потока и др.:  
GetThreadTimes(hThread1, &pftCreationTime1,  
&pftExitTime1, &pftKernelTime1, &pftUserTime1);  
Получение контекста потока:  
GetThreadContext(hThread, &ct).
```

Завершение потока: `ExitThread(0);` .
 Завершение потока извне: `TerminateThread(hThread1, 0);`.
 Окончание потока
 Существует несколько способов завершения потока:
 1) функция потока возвращает управление
 2) вызов функции `ExitThread` из потока
 3) вызов функции `TerminateThread`
 4) завершение процесса, содержащего данный поток
 Время выполнения потока
 Временные показатели работы потока определяются функцией `GetThreadTimes`, которая возвращает четыре временных параметра: время создания, время завершения, время выполнения, время выполнения. В программе определение и вывод времени создания потоков можно выполнить следующим образом:
`GetThreadTimes(hThread[i],&creationTime,&exitTime,&kernelTime,&userTime);`
 // Дата и время создания процесса по Гринвичу
`FileTimeToSystemTime(&creationTime, &stUTC);`
 // Конвертируем время создания процесса в местное время
`SystemTimeToTzSpecificLocalTime(NULL, &stUTC, &stLocal);`
`j+=sprintf(buf+j, "Время создания %d-го потока:%02d/%02d/%d %02d:%02d:%02d\r\n",`
`i+1, stLocal.wDay,`
`stLocal.wMonth, stLocal.wYear, stLocal.wHour,`
`stLocal.wMinute, stLocal.wSecond);`
 // Дата и время завершения процесса по Гринвичу
`FileTimeToSystemTime(&exitTime, &stUTC);`
 // Конвертируем время завершения процесса в местное время
`SystemTimeToTzSpecificLocalTime(NULL, &stUTC, &stLocal);`
`j+=sprintf(buf+j, "Время завершения %d-го потока: %02d/%02d/%d`
`%02d:%02d:%02d\r\n", i+1, stLocal.wDay,`
`stLocal.wMonth, stLocal.wYear, stLocal.wHour,`
`stLocal.wMinute, stLocal.wSecond);`
 Результаты работы программного обеспечения

Варианты:

Должны быть реализованы три потока, каждый из которых осуществляет передвижение собственной надписи по главному окну. Все надписи должны быть различны и двигаться с разной скоростью.

Главное окно должно быть разделено на четыре части. Также должны быть созданы четыре потока, каждый из которых раз в секунду изменяет цвет фона в своей части окна.

На главном окне должны быть созданы три элемента управления `Edit`. Необходимо создать три потока, каждый из которых раз в две секунды установит случайное число в соответствующий `Edit`.

При помощи трех потоков необходимо реализовать движение трех квадратиков по главному окну в случайных направлениях. Каждый поток должен двигать собственный квадратик.

При помощи пяти потоков реализовать падение пяти букв с различной скоростью по главному окну.

При помощи двух потоков необходимо реализовать движение двух шариков друг за другом по синусоиде.

При помощи двух потоков реализовать вращение двух палочек по экрану с различной скоростью

Главное окно должно быть разделено диагональю на две части. Необходимо создать два потока, каждый из которых раз в полсекунды закрасивает соответствующую часть

главного окна в случайный цвет.

Реализовать восемь потоков, каждый из которых рисует постепенно удлиняющийся луч. Все лучи должны исходить из одной точки и быть направлены под разными углами.

При помощи трех потоков реализовать механические часы: первый поток должен двигать часовую стрелку, второй минутную и третий секундную.

При помощи трех потоков реализовать электронные часы: первый поток должен выводить значение часа, второй – минуты, а третий – секунды.

Практическая работа № 2. Обмен данными.

Цель практической работы: изучить основы взаимодействия локальных процессов на базе файлов, проецируемых в память.

Задание:

1. Осуществить обмен текстовыми и графическими данными между двумя приложениями P1 и P2 с выводом информации в заданную область окна программы. 2. Реализовать обработку процессом файлов больших размеров. 3. Осуществить обмен данными между двумя приложениями P1 и P2 через Page файл с выводом информации в заданную область окна программы.

Эталон ответа:

```
1.exe case WM_INITDIALOG: hFileMapping =  
CreateFileMapping(INVALID_HANDLE_VALUE, NULL, PAGE_READWRITE, 0, 1000,  
TEXT("katarina")); if(hFileMapping == NULL) MessageBox(hDlg, TEXT("Невозможно  
создать проекцию файла в Page-файле..."), TEXT("Error"), MB_ICONERROR); else {  
sharedBuffer = (LPTSTR) MapViewOfFile(hFileMapping,  
FILE_MAP_ALL_ACCESS, 0, 0, 0); }
```

Варианты:

Написать две программы, каждая из которых создает окно. На первом из них должны быть созданы две группы RadioButton. В первой из них имеется выбор из трех цветов: красный, синий, зеленый. Во второй – из четырех типов примитивов: ромб, квадрат, круг, звезда. Также на первом окне должен быть создан Checkbox с надписью «Draw». Информация об изменениях состояния данных Checkbox и RadioButtons должна передаваться во второе окно. При щелчке мышкой по второму окну проверяется переданная информация о состоянии CheckBox. Если он не выбран, ничего не происходит; если он выбран, то в точке щелчка мышки рисуется выбранный во второй группе RadioButtons примитив цветом, выбранным в первой из групп.

Практическая работа № 3. Сетевое программирование сокетов.

Цель практической работы: получить навык программирования сокетов

Задание:

Создать приложение на потоковом соquete TCP. Сервер построен синхронно, выполнение потока блокируется, пока сервер не даст согласие на соединение с клиентом. Клиент завершает соединение с сервером, отправляя серверу сообщение .

Эталон ответа:

```

string theMassege = "Test socket";
byte[] msg = Encoding.ASCII.GetBytes(theMassege + "<TheEnd>");
//отправляем данные через сокет
int bytesSend = sender.Send(msg);
//получаем ответ от удаленного устройства
int bytesRec = sender.Receive(bytes);
Console.WriteLine("сервер отвечает :{0}", Encoding.ASCII.GetString
(bytes,0,bytesRec));
//освобождаем сокет
        sender.Shutdown(SocketShutdown.Both);
        sender.Close();
    }
    catch (Exception e)
    {
        Console.WriteLine("исключение {0}",e.ToString());
    }
}

```

Варианты:

Создать программу сканирования портов. Программа пытается соединиться с localhost по каждому порту, сообщается об успешности соединения, если соединение установить не удается перехватывается исключение SocketException.

Создать клиент – серверное приложение, которое переданную информацию со стороны клиента считывает в файл.

Добавить в приложение подтверждения от сервера в ответ на переданные ему пакеты.

Практическая работа № 4. Работы с буфером экрана.

Цель практической работы: изучить основы работы с объектом – буфер обмена

Задание:

Создать приложение, состоящее из двух процессов:

- первый процесс записывает текстовый файл и растровый рисунок в буфер обмена;
- второй процесс считывает информацию из буфера обмена и отображает в окне процесса.

Эталон ответа:

```

case IDC_BUTTON1: { if (OpenClipboard(hDlg)) { UINT format =
RegisterClipboardFormat(L"MyClipboardData");
//вызываем второй раз, чтобы просто получить формат if
(IsClipboardFormatAvailable(format))
{ MyClipboardData MCD; //извлекаем данные из буфера HANDLE hData =
GetClipboardData(format);
MyClipboardData* buffer = (MyClipboardData *) GlobalLock(hData);
//заполняем нашу структуру полученными данными MCD = *buffer;
GlobalUnlock(hData) SetDlgItemTextA(hDlg, IDC_EDIT1, MCD.In fData);
CloseClipboard();
} } CloseClipboard();
return TRUE;
}

```

Варианты:

Реализуйте горизонтальную бегущую строку.

Реализуйте вертикальную бегущую строку.

Реализуйте строку, бегущую по периметру дисплея.

Реализуйте счёт в двоичной позиционной системе счисления.

Реализуйте счёт в шестнадцатеричной позиционной системе счисления.

Добавьте в файл graphics.c функцию вывода переменной типа unsigned int, добавьте

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ ДАГЕСТАН
Частное профессиональное образовательное учреждение
«Республиканский полипрофессиональный колледж»

Варианты контрольных работ МДК 01.04 Системное программирование

1. Создание простых программ на языке Ассемблер. Разбор ошибок и отладка программ.
2. Создание программ с разветвлением на Ассемблере. Решение проблем компиляции.
4. Создание программ с использованием циклов. Решение проблем отладки и компиляции.
5. Создание программ с использованием логических операций. Решение проблем логичной структуры алгоритмов программы.
6. Создание программ с использованием циклов и логических операций.
7. Создание программ с использованием команд сдвигов. Разбор ошибок логических операций.
8. Составление программ с использованием ввода-вывода на экран. Разбор ошибок ввода-вывода.

Вариант 1

1. Изучить программную модель микроконтроллера, используя раздел «Теоретическое введение».
2. Изучить инструкцию пользователя симулятора микроконтроллера, используя соответствующий подраздел «Теоретического введения».
3. Изучить команды пересылки. Изучение каждой команды проводить следующим образом:
 - Открыть окно «Дисассемблер».
 - В пункте меню «Правка» выбрать опцию «Вставка команды».
 - В открывшемся окне «Вставить инструкцию» ввести мнемокод соответствующей команды пересылки, описание которой представлено в меню «Справка», и нажать на кнопку ОК. Записать в отчет строку о введенной в окно «Дисассемблер» команде.
 - Открыть окна «Процессор», «Регистры», «Дамп памяти данных». Просмотреть содержимое этих окон.
 - Выполнить команду пересылки, выбрав в меню «Запуск команды» опцию «Шаг».
 - Записать в отчет изменения в окнах «Процессор», «Регистры», «Дамп памяти данных».
 - Для повторного выполнения команды в меню «Запуск команды» выбрать опцию «Сброс».
4. Написать и исследовать работу программы пересылки байта из ячейки памяти программ с адресом 71H в регистр R1 банка 1.
 1. Функциональная модель микроконтроллера.
 2. Содержимое регистров и ячеек памяти окон «Процессор», «Регистры», «Дамп памяти данных» до и после выполнения каждой команды пересылки в соответствии с таблицей 1.3.
 3. Содержимое регистров и ячеек памяти окон «Процессор», «Регистры», «Дамп памяти данных» и памяти программ до и после выполнения каждой команды разработанной программы в соответствии с таблицей 1.4.

Таблица 1 - Результаты выполнения команд

№	Команда	Код	Выполняемая операция	Содержимое используемых регистров и ячеек памяти до и после выполнения		Пояснение
				До	После	
1	MOV A,R0	E8	Пересылка байта данных из регистра R0 в аккумулятор A	A/00 PC/00 PSW/00	A/F2 PC/01 PSW/01	
2
...
...

Вариант 2

1. Исследовать команды записи в стек и считывания из стека: PUSH P1; PUSH 01; PUSH PSW; POP PSW; POP 01; POP P1.

2. Исследовать команды входа и выхода из подпрограмм.

3. Написать и исследовать работу подпрограммы возведения в квадрат числа в аккумуляторе.

4. Написать и исследовать программу опроса содержимого двух младших разрядов порта P1 и перехода в зависимости от их состояния к одной из четырех подпрограмм, начальные адреса которых находятся в ячейках 21H, 25H, 3AH и 2FH.

1. Загрузить последовательность команд: MOV P1, #3FH; MOV R1, #C3H; MOV A, R1; PUSH P1; PUSH 01; PUSH PSW; SETB RS1; MOV P1, #1H; MOV 01H, #02H; POP PSW; POP 01; POP P1.

2. Выполнить последовательность команд в пошаговом режиме.

3. Записать в отчет в соответствии с таблицей 2.2 выполняемые команды и содержимое изменяемых регистров и ячеек памяти.

4. Загрузить, начиная с адреса 0000, последовательность команд: ACALL 05; LJMP 20; NOP; NOP; NOP; RET.

5. Выполнить программу, состоящую из данной последовательности команд в пошаговом режиме.

6. Записать в отчет в соответствии с таблицей 2.2 выполняемые команды и содержимое изменяемых регистров и ячеек памяти.

7. Разработать и ввести подпрограмму возведения в квадрат числа с использованием команды MUL AB. Для обращения к подпрограмме использовать команду ACALL.

8. Убедиться в работоспособности подпрограммы.

9. Записать в отчет в соответствии с таблицей 2.2 выполняемые команды и содержимое изменяемых регистров. Определить время выполнения подпрограммы.

10. Разработать программу опроса содержимого двух младших разрядов порта P1 и перехода в зависимости от их состояния к одной из четырех подпрограмм, начальные адреса которых находятся в ячейках 21H, 25H, 3AH и 2FH

11. Задавая в порт различные кодовые комбинации в используемых разрядах, проверить работоспособность программы.

Таблица 2 - Результаты выполнения команд

№	Команда	Код	Выполняемая операция	Содержимое регистров и памяти до и после выполнения		Пояснение
				До	После	
1	MOV A,R0	E8	Пересылка байта данных из регистра R0 в аккумулятор A	A/00 R0/F2 0000/F2	A/F2 R0/F2 0000/F2	
2

Вариант 3

1. Разработать и ввести в окно «Дисассемблер» программу сложения двух произвольно выбранных однобайтовых чисел.
2. Осуществляя запуск программы в режиме «Шаг», убедиться в правильности ее выполнения, используя окна «Дисассемблер». «Процессор», «Регистры». Результаты наблюдений свести в табл.3.3. Определить время выполнения программы.
3. Разработать и ввести в окно «Дисассемблер» программу умножения двух произвольно выбранных однобайтовых чисел.
4. Осуществляя запуск программы в режиме «Шаг», убедиться в правильности ее выполнения, используя окна «Дисассемблер». «Процессор», «Регистры». Обратит внимание на формируемые флаги. Результаты наблюдений свести в таблице 3.3. Определить время выполнения программы.

Таблица 3 - Результаты выполнения команд

№	Команда	Код	Выполняемая операция	Содержимое регистров и памяти до и после выполнения		Пояснение
				До	После	
1	MOV A,R0	E8	Пересылка байта данных из регистра R0 в аккумулятор A	A/00 R0/F2 0000/F2	A/F2 R0/F2 0000/F2	
2

5. Разработать и ввести в окно «Дисассемблер» программу деления произвольно выбранных однобайтовых чисел.
 6. Осуществляя запуск программы в режиме «Шаг», убедиться в правильности ее выполнения, используя окна «Дисассемблер». «Процессор», «Регистры». Результаты наблюдений свести в таблицу. Определить время выполнения программы.
 7. Исследовать команды сдвига вправо и влево. Результаты наблюдений свести в таблицу.
 8. Загрузить из каталога *Examples* программу «Временная задержка.txt».
 9. Используя меню *Запуск*, установить точку останова программы на команде с адресом 0007.
 10. Запустить программу в непрерывном режиме (опция *Запуск*).
 11. В окне «Процессор» считать информацию о времени выполнения и числе циклов исполняемой программы.
 12. Используя длительности исполнения команд программы, рассчитать время исполнения программы.
 13. Ввести изменения в исполняемой программе для получения временной задержки, заданной преподавателем.
 14. Запустить модернизированную программу и определить время ее выполнения в соответствии с п.п. 8 – 10. Сравнить полученные данные с расчетным значением.
- Система оценки знаний

Отлично: выполнен вариант 3. Ошибки отсутствуют (допущена незначительная ошибка).
Хорошо: выполнен вариант 3, допущено не более 3-х неточностей/ошибок; выполнен вариант 2, ошибки отсутствуют (допущены незначительные недочеты/ошибки, но не более 3-х).

Удовлетворительно: выполнен вариант 3, допущено 4-5 неточностей/ошибок; выполнен вариант 2, допущено не более 4 неточностей/ошибок; выполнен вариант 1, ошибки отсутствуют (допущено не более 3-х неточностей/ошибок).

Неудовлетворительно: допущено большее количество ошибок; задания выполнены частично или не выполнены.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ ДАГЕСТАН

Частное профессиональное образовательное учреждение

«Республиканский полипрофессиональный колледж»

Самостоятельная работа при изучении МДК.01.04 Системное программирование

Задание 1. Подготовить доклады по темам:

«Принцип программного управления»

«Классическая схема ЭВМ»

«Основы автоматизации вычислительного процесса»

Задание 2. Ответить на вопросы.

1. Что такое программный принцип управления компьютером?

2. На какие типы классифицируют программное обеспечение?

3. Расскажите классификацию системных программ.

4. Приведите примеры различного системного программного обеспечения согласно классификации.

5. Расскажите классификацию прикладных программ.

6. Приведите примеры различного прикладного программного обеспечения согласно классификации.

7. Расскажите классификацию инструментальных программ.

8. Приведите примеры различного инструментального программного обеспечения согласно классификации.

9. Подумайте, с каким программным обеспечением вам чаще всего придется сталкиваться, обоснуйте свой ответ.

Форма контроля:

подготовка сообщения

опросы, собеседование;

3.5 Задания для оценки освоения МДК 01.05 Разработка веб-приложений

Практические и лабораторные работы Тема 5.1 Веб-дизайн и разработка

Практическая работа № 1. «Установка программного комплекса. Связка Apache+PHP+MySQL»

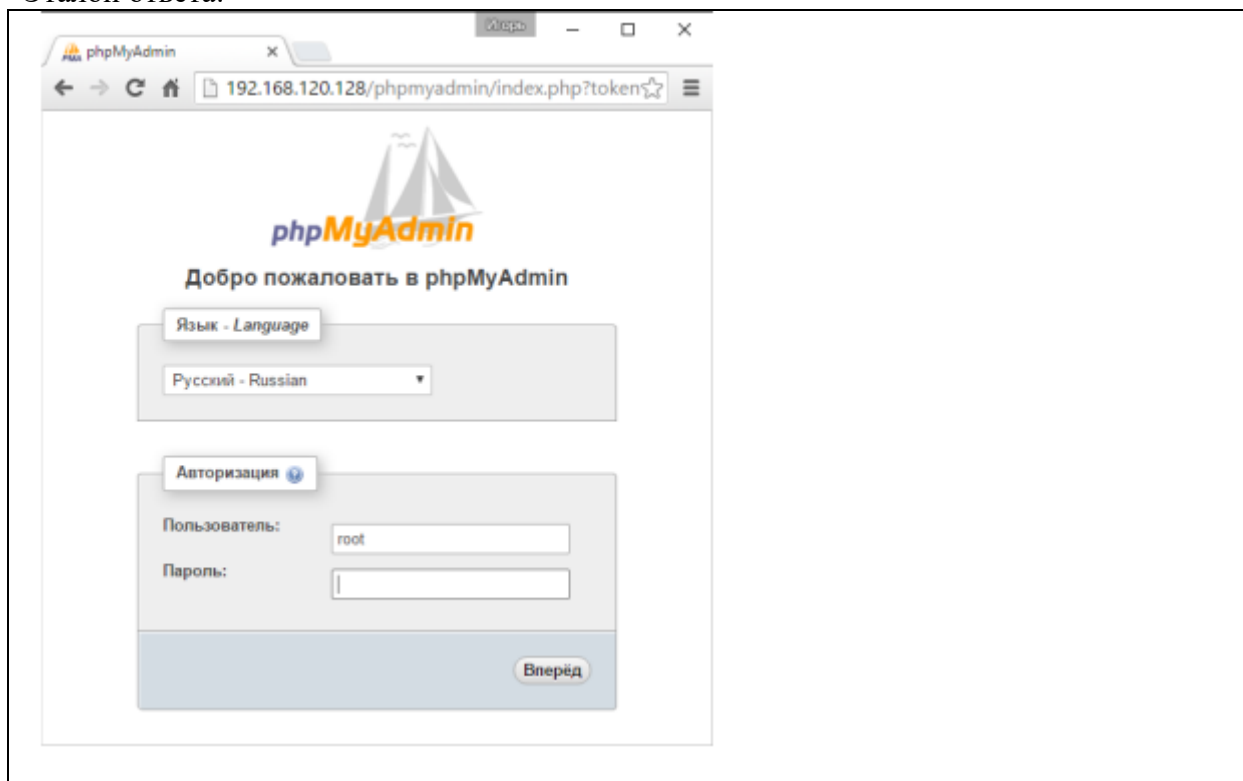
Цель практической работы: Настройка веб-сервера на операционной системе, состоящая из свободного программного обеспечения с открытым исходным кодом – Debian.

Изучение основных команд системы Debian, необходимых для настройки системы.

Задание:

В ходе лабораторной работы настройка веб-сервера будет осуществляться на виртуальной машине с операционной системой Debian, которая была создана с использованием VMPlayer. Инструкция по запуску виртуальной машины приведена в разделе 9. Вход в операционную систему необходимо выполнять под пользователем student, для выполнения команд требующих особых привилегий необходимо использовать учетную запись суперпользователя root. Пароли к учетным записям приведены в разделе 8.

Эталон ответа:



Практическая работа № 2. «Обработка данных веб-форм методом GET»

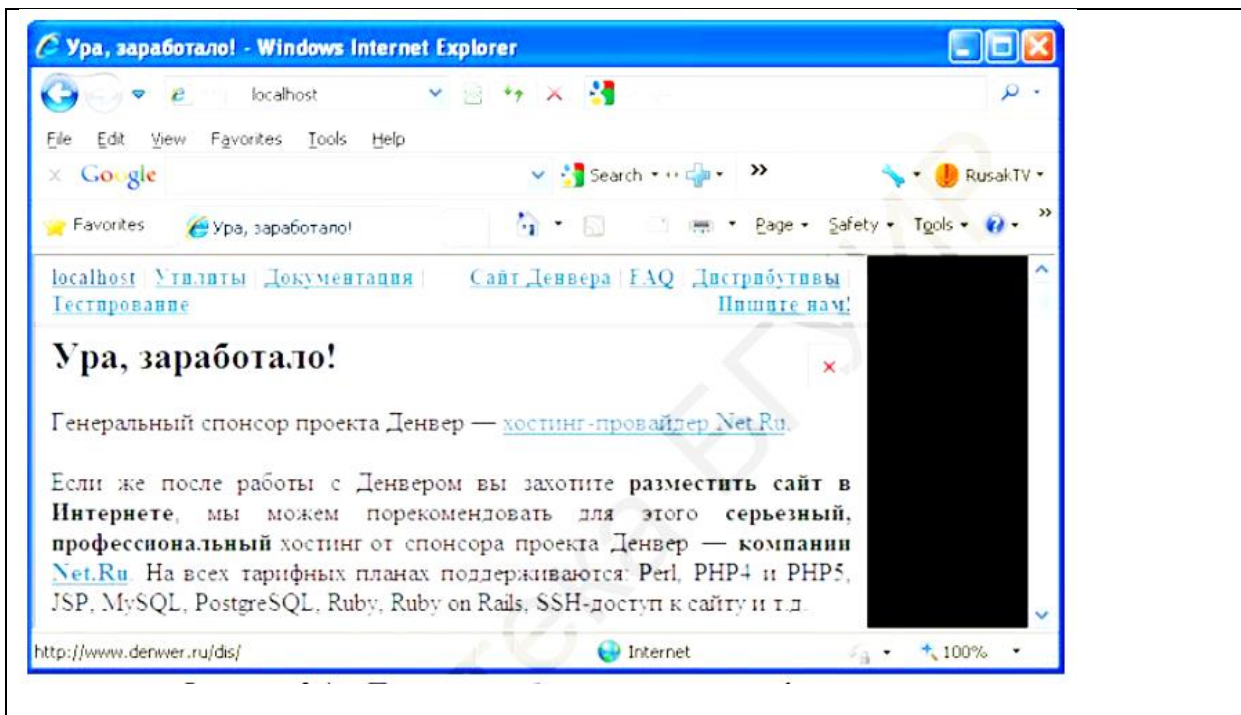
Цель практической работы: учить основные элементы, которые могут быть размещены в HTML-форме и научиться обрабатывать данные, введенные в форму с помощью языка программирования PHP

Задание:

Обработка данных, вводимых пользователем на странице form1.html, при помощи скрипта reader.php, расположенного в том же каталоге.

```
<HTML>
  <HEAD>
    <TITLE> Пример HTML-формы </TITLE>
  </HEAD>
  <BODY>
    <H1> Пример HTML-формы </H1>
    <FORM METHOD= "POST" ACTION= "reader.php" >
      Здесь должны быть элементы формы.
    </FORM>
  </BODY>
</HTML>
```

Эталон ответа:



Варианты:

Добавить в созданную форму текстовые поля для ввода ФИО пользователя, логина, пароля и электронного адреса. Все элементы формы разместить в таблице. Таблица располагается по центру, и границы таблицы не отображаются

Создать файл reader.php для вывода на экран в табличной форме пользовательских данных, введенных в форму, которая хранится в файле form1.html.

Добавить в форму флажки для выбора пользователем иностранных языков, которыми он свободно владеет. Вывести на экран выбранные значения вместе с остальными данными

Практическая работа № 3. «Обработка данных веб-форм методом POST»

Цель практической работы: научиться обрабатывать формы на Web-страницах с помощью сценариев на языке PHP.

Задание:

1. Реализовать на практике пример обработчика формы для ввода логина и пароля.
2. Разработать сценарий для вывода приветствия пользователя, который вводит свое имя с помощью формы. Для передачи данных использовать метод GET.
3. Разработать сценарий вычисления суммы, разности, произведения, частного и остатка для введенных двух чисел. Проверить его выполнения. Для передачи данных использовать метод POST.
4. Разработать сценарий для проведения тестирования (4-5 вопросов) и вывод результатов анкетирования пользователя.

Эталон ответа:

Форма для регистрации участников

Имя

Фамилия

E-mail

Выберите курс, который вы бы хотели посещать:

- РНР
- Лер
- Ред
- Уик

Что вы хотите, чтобы мы знали о вас?

Подтвердить получение

Варианты:

Создать форму следующего вида

Фамилия

Инициалы

Пароль

Адрес получателя

Количество

Тип

Доставка Почтой Электронно Курьером

Требуется накладная

Дополнительная информация

Создать форму следующего вида

Username

Email

Password

Confirm password

Already a member? [Sign in](#)

Создать форму следующего вида

Tambah Data Siswa

Nama

Kelas

Alamat

Создать форму следующего вида

Tambah Data Siswa

Nama

Kelas

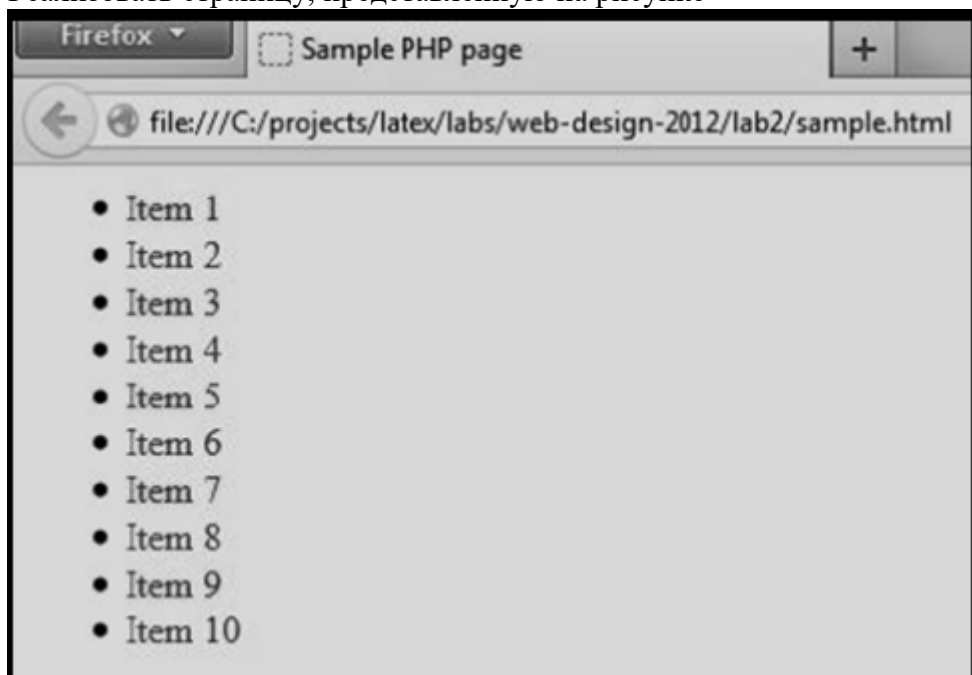
Alamat

Практическая работа № 4. «Программирование базовых конструкций на PHP, обработка строк»

Цель практической работы: Освоение базовых конструкций языка PHP: средства вывода; организация циклов; операторы ветвления.

Задание:

Реализовать страницу, представленную на рисунке

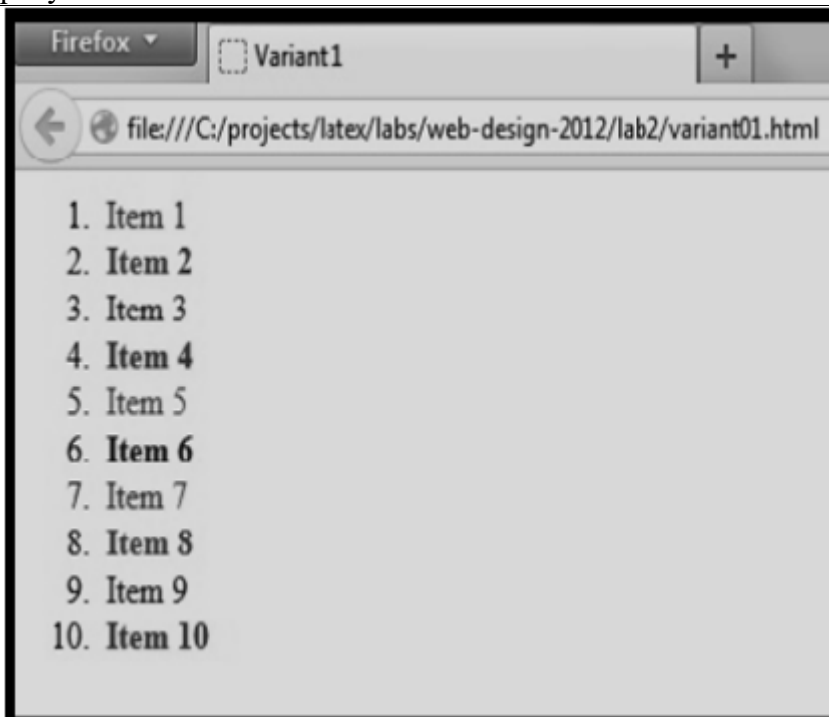


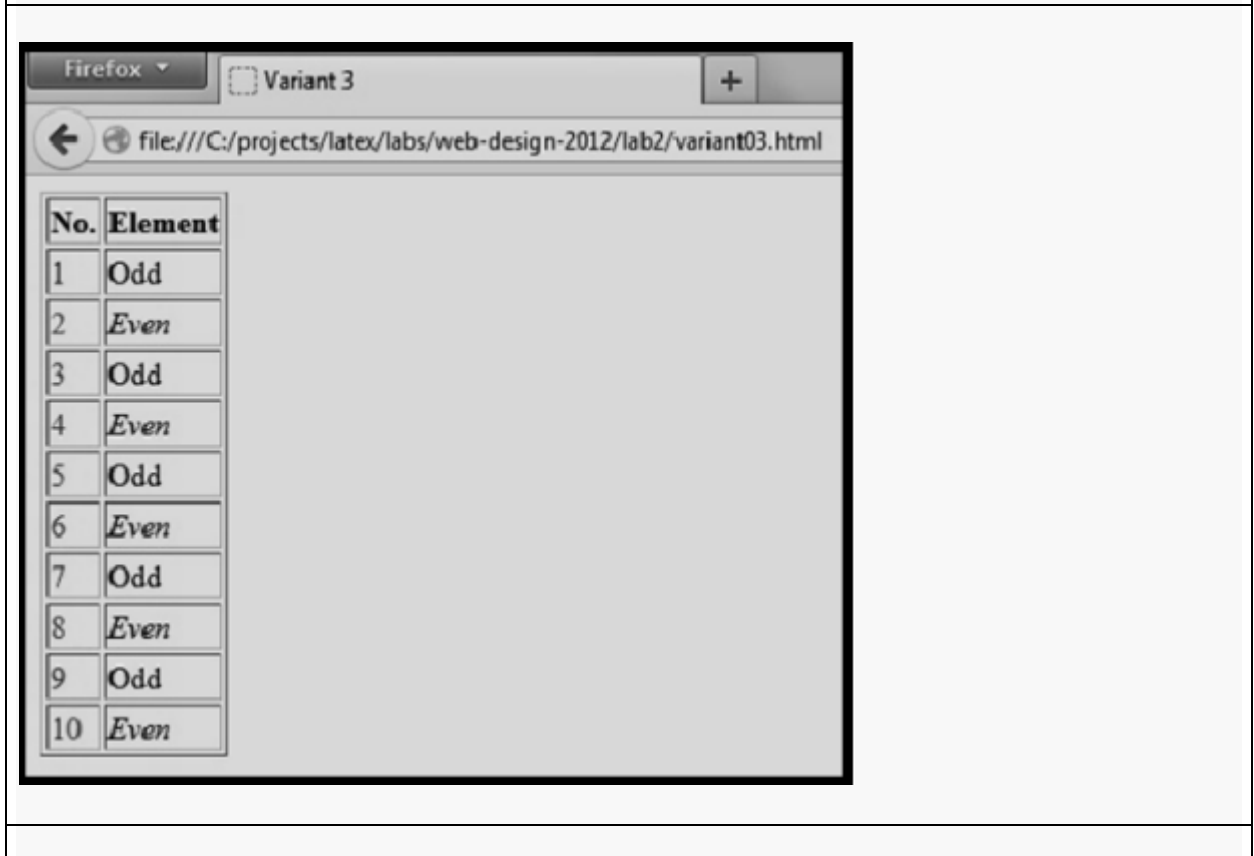
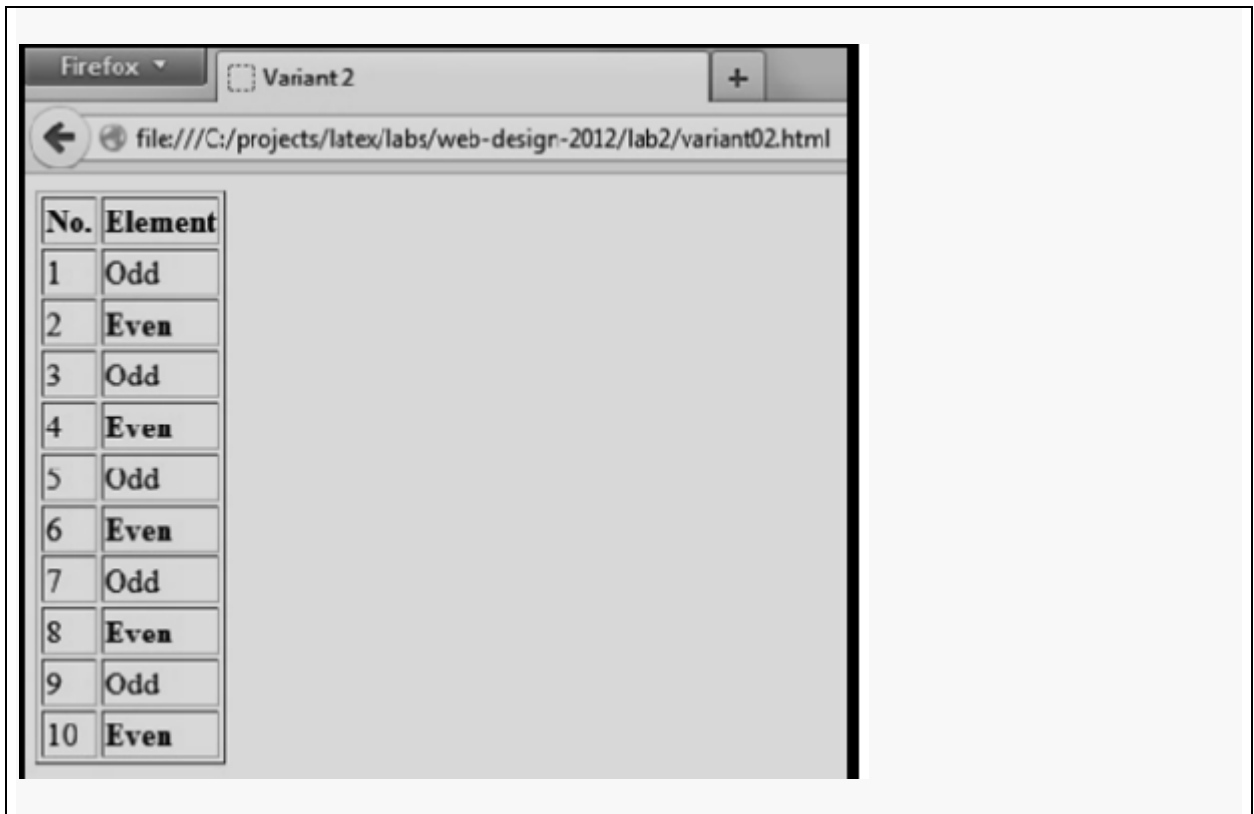
Эталон ответа:

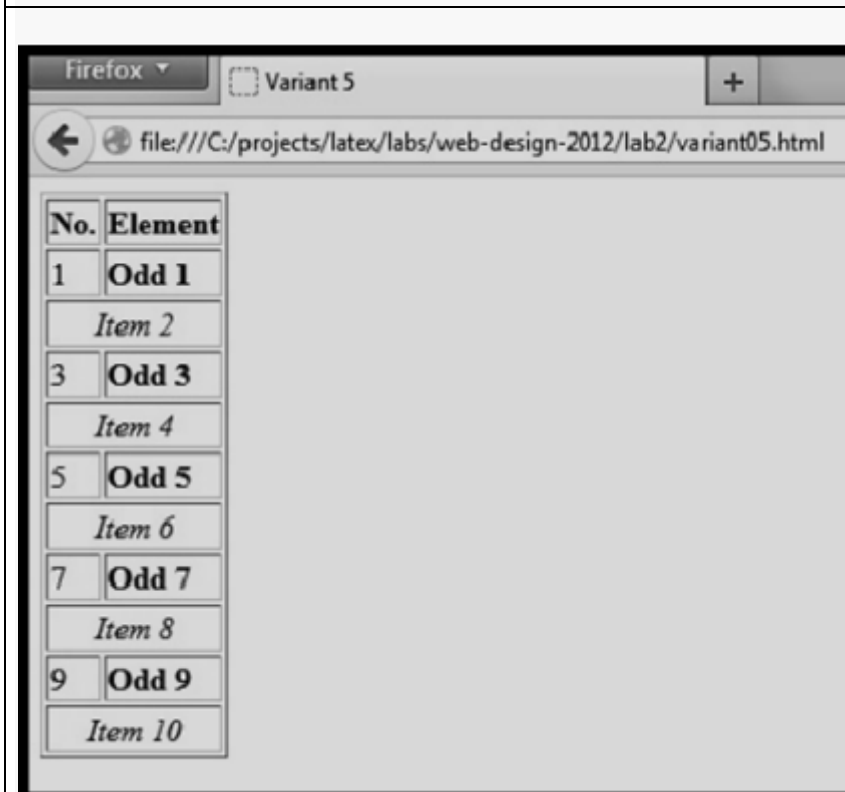
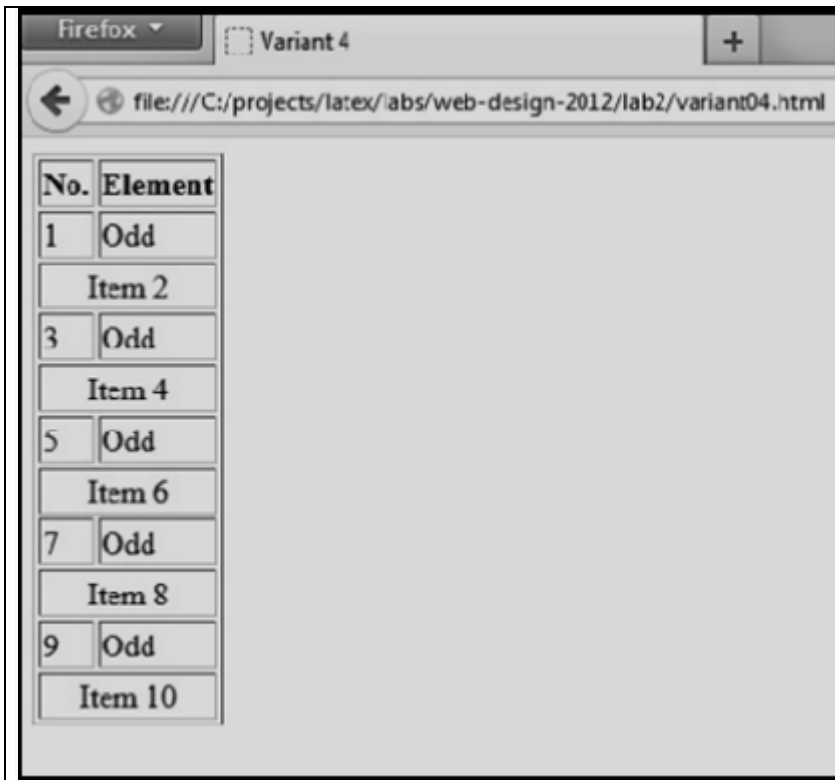
```
<?php
for ($i = 1; $i <= 10; $i++) {
    // Вывод тэга элемента
    echo "<li>";
    // Вывод тэга шрифта в зависимости от четности
    if ($i%2 == 0) echo "<font color = \"green\">";
    else echo "<font color = \"red\">";
    // Вывод содержимого элемента
    echo "Item ".$i;
    // Вывод закрывающего элемента шрифта
    echo "</font>";
    // Вывод закрывающего элемента списка с переводом на новую строку
    echo "</li>\n";
}
?>
```

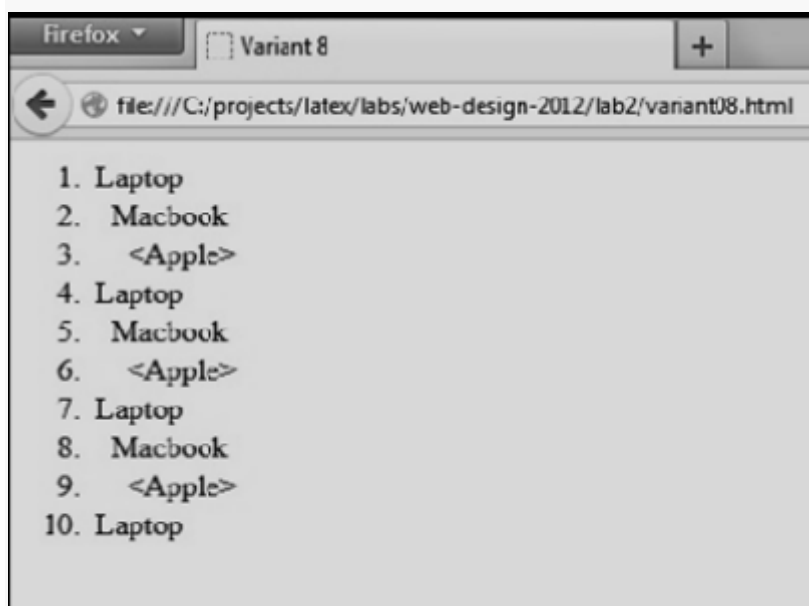
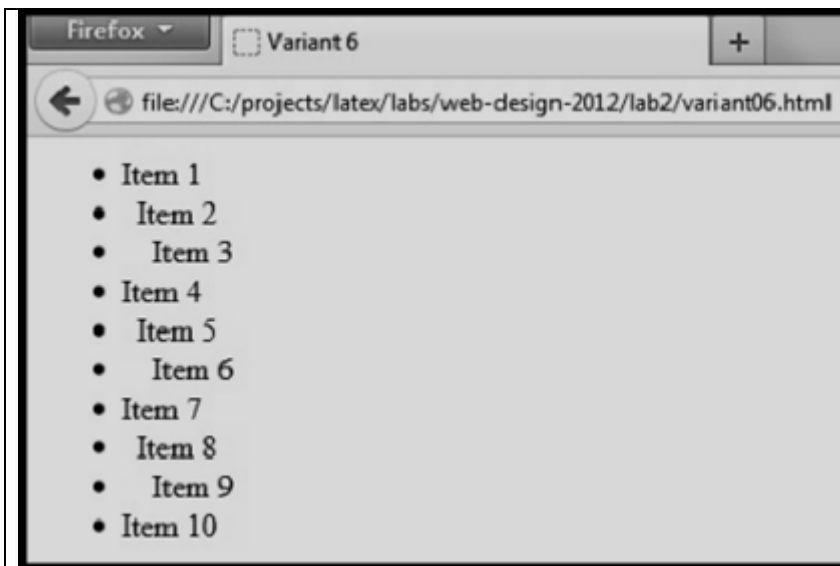
Варианты:

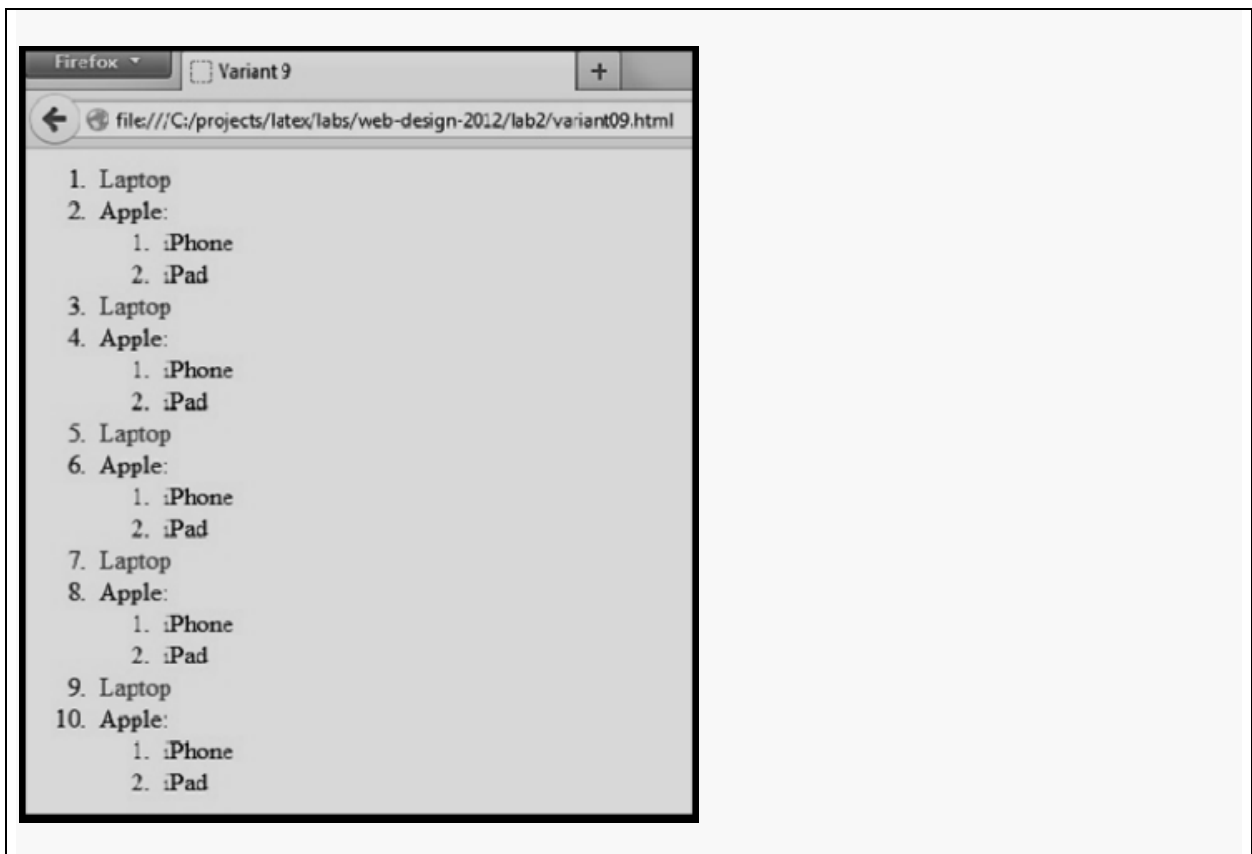
Необходимо проанализировать каждую страницу, выявить, какие тэги требуются для ее реализации, и написать исходный код этой страницы, после чего убедиться, что браузер отображает эту страницу именно так, как она выглядела изначально. Для каждого варианта следует проставить заголовок страницы, который отображает браузер на рисунке











Практическая работа №5. «Программирование ветвлений и циклов на PHP»

Цель практической работы: Установка и настройка компонентов платформы AMP и реализация базового примера на PHP для ознакомления с синтаксисом языка и особенностями библиотек

Задание:

Написать сценарий для вывода на экран

Андрей Иванов

Борис Петров
Сергей Волков
Федор Макаров

Эталон ответа:

```
<?php
$names["Иванов"] = "Андрей";
$names["Петров"] = "Борис";
$names["Волков"] = "Сергей";
$names["Макаров"] = "Федор";
foreach ($names as $key => $value) {
echo "<b>$value $key</b><br>";
}
?>
```

Варианты:

1. Необходимо сгенерировать таблицу. Заголовок и строки генерируются с помощью PHP, а не прописываются вручную. Заголовок содержит 5 ячеек, i -я ячейка – название "Header i ". Помимо заголовка в таблице 10 строк. Каждая четная строка выделяется цветом Aqua. В первом столбце каждой строки отображается номер ячейки, остальные ячейки строки хранят только знак решетки. В каждой третьей строке текст должен быть выделен курсивом.

2. Необходимо сгенерировать двухуровневый пронумерованный список. Первый уровень содержит 10 элементов. i -й элемент – это текст "Odd i " или "Even i " в зависимости от того, является ли номер элемента четным числом. Каждый элемент первого уровня содержит 5 дочерних элементов второго уровня. i -й дочерний элемент – это текст "Child i ". Нечетные дочерние элементы нужно выделить жирным. Каждый третий элемент первого уровня должен быть красного цвета.

3. Необходимо сгенерировать маркированный список из первых 10 элементов последовательности Фибоначчи. Каждый элемент с нечетным значением выделить курсивом, каждый третий элемент – цветом Aqua. i -й элемент должен содержать текст вида "F(i) = [Значение i -го элемента]"

4. Вывести список-"лесенку" из 10 элементов, где каждый следующий элемент располагается строкой ниже предыдущего и на 3 отступа правее предыдущего. Каждый нечетный элемент выделить жирным, каждый третий элемент – красным цветом. i -й элемент – это текст вида "Item i ".

Практическая работа № 6. «Программирование массивов на PHP»

Цель практической работы: изучить способы получения случайных чисел. Написать программу для работы с одномерными массивами.

Задание:

Создайте массив, заполненный числами от 1 до 100. Найдите сумму элементов данного массива.

Эталон ответа:

для начала создадим массив с числами от 1 до 100. Вручную это сделать будет очень сложно, поэтому воспользуемся функцией range:

```
<?php
    $arr = range(1, 100);
?>
```

Сумму элементов полученного массива найдем с помощью функции array_sum (можно использовать цикл foreach, как мы это делали раньше, но array_sum в данном случае

гораздо удобнее и проще):

```
<?php
```

```
//Найдем сумму элементов нашего массива:
```

```
echo array_sum(range(1, 100));
```

```
?>
```

Варианты:

1. В массиве из 20 целых чисел найти наибольший элемент и поменять его местами с первым элементом.

2. В массиве из 10 целых чисел найти наименьший элемент и поменять его местами с последним элементом.

3. В массиве из 15 вещественных чисел найти наибольший элемент и поменять его местами с последним элементом.

4. В массиве из 25 вещественных чисел найти наименьший элемент и поменять его местами с первым элементом.

5. Дан массив X, содержащий 27 элементов. Вычислить и вывести элементы нового массива Y, где $y_i = 6.85x_i^2 - 1.52$. Если $y_i < 0$, то вычислить и вывести $a = x_i^3 - 0.62$ и продолжить вычисления; если $y_i \geq 0$, то вычислить и вывести $b = 1/x_i^2$ и продолжить вычисления.

6. Дан массив X, содержащий 16 элементов. Вычислить и вывести значения d_i , где
$$d_i = \frac{e^{x_i} + 2e^{-x_i}}{\sqrt{5 + \sin x_i}}$$
 и значения $d_i > 0.1$.

7. Дан массив Y, содержащий 25 элементов. Записать в массив R и вывести значения элементов, вычисляемые по формуле
$$r_i = \frac{5y_i + \cos^2 y_i}{2.35}, \quad i=1,2,\dots,25.$$

8. Дан массив F, содержащий 18 элементов. Вычислить и вывести элементы нового массива $p_i = 0.13f_i^3 - 2.5f_i + 8$. Вывести отрицательные элементы массива P.

9. Вычислить и вывести элементы массива Z, где $z_i = i^2 + 1$, если i – нечетное, и $z_i = 2i - 1$, если i – четное. Сформировать и вывести массив D: $d_i = 2.5z_i$, если $z_i < 2.5$ и $d_i = z_i/2.5$, если $z_i \geq 2.5$.

10. Заданы массивы D и E. Вычислить и вывести значения $f_i = (2d_i + \sin e_i)/d_i$, где $i=1,2,\dots,16$; вывести $1 < f_i < 3$.

11. В массиве R, содержащем 25 элементов, заменить значения отрицательных элементов квадратами значений, значения положительных увеличить на 7, а нулевые значения оставить без изменения. Вывести массив R.

12. Дан массив A целых чисел, содержащий 30 элементов. Вычислить и вывести сумму тех элементов, которые кратны 5.

13. Дан массив A целых чисел, содержащий 30 элементов. Вычислить и вывести сумму тех элементов, которые нечетны и отрицательны.

14. Дан массив A целых чисел, содержащий 30 элементов. Вычислить и вывести сумму тех элементов, которые удовлетворяют условию $|a_i| < i^2$.

15. Дан массив A целых чисел, содержащий 30 элементов. Вычислить и вывести количество и сумму тех элементов, которые делятся на 5 и не делятся на 7.

Практическая работа № 7. «Обработка результатов множественного выбора данных средствами PHP»

Цель практической работы: Изучить механизм обработки HTML-формы PHP-скриптом.

Задание:

Создать HTML-документ, который будет содержать практически все элементы HTML-форм.

Эталон ответа:

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
<title>Test Form</title>
</head>
<body>
<h3>Тестовая форма</h3>
<form name="form1" method="post" action="script.php">
<p><span>Текстовое поле: </span>
<input type="text" name="textfield">
</p>
<p>Поле ввода пароля:
<input type="password" name="pswfield">
</p>
<p>Скрытое поле hidden
<input name="hidden" type="hidden" id="hidden" value="Скрытое_значение">
</p>
<hr size="1">
<p>Независимые переключатели (checkbox):</p>
<p>
<input type="checkbox" name="checkbox1" value="1">
Вариант первый
<input type="checkbox" name="checkbox2" value="1">
Вариант второй
<input type="checkbox" name="checkbox3" value="1" checked>
Вариант третий (по умолчанию)</p>
<hr size="1">
<p>Зависимые переключатели (radio):</p>
<p>
<input name="radiobutton" type="radio" value="yes">
Да
<input name="radiobutton" type="radio" value="no">
Нет</p>
<hr size="1">
<p>Многострочное текстовое поле (textarea):</p>
<p>
<textarea name="textarea" cols="40" rows="10">Текст по умолчанию</textarea>
</p>
<hr size="1">
<p>Список с единственным выбором:</p>
<p>
<select name=day_s size=1>
<option value=1>Понедельник</option>
<option value=2>Вторник</option>
<option value=3 selected>Среда</option>
<option value=4>Четверг</option>
<option value=5>Пятница</option>
<option value=6>Суббота</option>
<option value=7>Воскресенье</option>
```


случае параметры передаются так:

```
day_m=01&day_m=03&day_m=07...
```

Множественный список можно представить в виде массива, а обработать его элементы с помощью цикла foreach. Нам даже не обязательно знать количество элементов множественного списка. Нам нужно лишь предварительно дать понять PHP, что мы будем передавать массив:

```
<select name="day_m[]" size=7 mutiple>
```

Квадратные скобки [] - это признак массива. Циклическая обработка массива осуществляется так:

```
foreach ($_POST['day_m'] as $key=>$value) echo "$key = $value <br>";
```

А теперь приведем окончательный листинг PHP-скрипта, обрабатывающего тестовую форму:

```
<?php
```

```
// Выводим HTML-заголовки:
```

```
echo '<html>';
```

```
echo '<head>';
```

```
echo '<meta http-equiv="Content-Type" content="text/html; charset=windows-1251">';
```

```
echo '<title>Test Form</title>';
```

```
echo '</head>';
```

```
echo '<body>';
```

```
echo '<h3>Тестовая форма</h3>';
```

```
echo "<p>Переданное значение текстового поля: <b>".$_POST['textfield']."</b></p>";
```

```
echo "<p>Переданное значение поля пароля: <b>".$_POST['pswfield']."</b></p>";
```

```
echo "<p>Переданное значение скрытого поля hidden: <b>".$_POST['hidden']."</b></p>";
```

```
echo '<hr size="1">';
```

```
echo '<p>Были включены следующие независимые переключатели: </p>';
```

```
if (isset($_POST['checkbox1'])) echo "<p><b>Первый</b></p>";
```

```
if (isset($_POST['checkbox2'])) echo "<p><b>Второй</b></p>";
```

```
if (isset($_POST['checkbox3'])) echo "<p><b>Третий</b></p>";
```

```
echo '<hr size="1">';
```

```
if (isset($_POST['radiobutton']))
```

```
{
```

```
echo '<p>Был выбран независимый переключатель со следующим значением: ';
```

```
if ($_POST['radiobutton']=="yes") echo "<b>Yes</b>";
```

```
if ($_POST['radiobutton']=="no") echo "<b>No</b>";
```

```
echo '</p>';
```

```
}
```

```
else echo '<p>Ни один из независимых переключателей не был выбран</p>';
```

```
echo '<hr size="1">';
```

```
echo '<p>Значение многострочного текстового поля :</p>';
```

```
echo "<p><b>".$_POST['textarea']."</b></p>";
```

```
echo '<hr size="1">';
```

```
echo "<p>Значение списка с единственным выбором: <b>".$_POST['day_s']."</b></p>";
```

```
echo '<hr size="1">';
```

```
echo '<p>Значения списка с множественным выбором: </p>';
```

```
foreach ($_POST['day_m'] as $keys=>$values) echo "<b>$values</b><br>";
```

```
echo '</body>';
```

```
echo '</html>';
```

```
?>
```

Варианты:

Задание 1.

Запустить Denwer.

- Задание 2.** Открыть HTML-документ практической работы №2.
- Задание 3.** Создать новый документ *.php (* - имя документа, прописанное в теге формы),
- Задание 4.** Например, <form name="form1" method="post" action="script.php">
- Задание 5.** В данном случае имя документа – script.php.
- Задание 6.** Написать PHP-скрипт вывода данных в окно браузера.
- Задание 7.** Получить результат обработки HTML-формы.
- Задание 8.** Результаты выполнения практического задания запишите в отчет.

Практическая работа № 8. «Обработка результатов фильтрации данных, полученных с помощью переключателей средствами PHP»

Цель практической работы: Изучить механизм обработки форм в PHP

Задание: Нужно написать обработчики формы (см. выше) для регистрации участников заочной школы программирования и после регистрации отправить участнику сообщение.

Эталон ответа:

```
<h2>форма для регистрации студентов</h2>
<form action="1.php" method=POST>
Имя <br><input type=text name="first_name"
    value="Введите Ваше имя"><br>
Фамилия <br><input type=text name="last_name"><br>
E-mail <br><input type=text name="email"><br>
<p> Выберите курс, который вы бы хотели посещать:<br>
<input type=checkbox name='kurs[]' value='PHP'>PHP<br>
<input type=checkbox name='kurs[]' value='Lisp'>Lisp<br>
<input type=checkbox name='kurs[]' value='Perl'>Perl<br>
<input type=checkbox name='kurs[]' value='Unix'>Unix<br>
<p>Что вы хотите, чтобы мы знали о вас? <BR>
<textarea name="comment" cols=32 rows=5></textarea>
<input type=submit value="Отправить">
<input type=reset value="Отменить">
</form>
```

Варианты:

1. Написать программу-вычислитель, которая позволяет пользователю передать два числа и указать операцию, выполняемую над ними.
2. Составить программу, благодаря которой пользователь может выбрать цвет и размер шрифта из предложенного списка.
3. Написать программу, обрабатывающую ввод анкетных данных клиента и реагирующую соответствующим образом. Например, программа может выводить строку «Здравствуйте, Иванов Петр Федорович! Спасибо за заказ! Мы рады приветствовать в вашем лице клиента нашей фирмы» или «Здравствуйте, Симонова Инна Леонидовна! Спасибо за заказ! Так как сумма вашего заказа превысила 5000 рублей, Вы получаете 10 %-ную скидку». Анкетные данные: ФИО, возраст, сумма заказа.
4. Программа-загадка. Пользователь должен отгадать число, загаданное программой. Если введенное пользователем число не равно загаданному, то ему сообщается об этом и программа советует выбрать другое число, большее или меньшее предыдущего. Если пользователь отгадал число, то выводится строка приветствия.
5. Составить программу, благодаря которой пользователь может выбрать цвет фона из предложенного списка.
6. Создать форму для приема сообщения посетителя и выводить его сообщение на другой странице.
7. Создать форму со списком товаров на продажу и выводить информацию о заказе

<i>посетителя на другой странице.</i>
<i>8. Составить программу, благодаря которой пользователь может выбрать язык (русский или английский, например), на котором отображается содержимое страницы.</i>
<i>9. Создать форму со списком предоставляемых услуг вашей туристической фирмой. После выбора пользователем какой-нибудь услуги выводится более полная информация о ней.</i>
<i>10. Составить программу, благодаря которой пользователь может выбрать цвет шрифта из предложенного списка</i>

Практическая работа № 9. «Проверка вводимых пользователем данных на веб-странице средствами PHP»

Цель практической работы: научиться обрабатывать формы на Web-страницах с помощью сценариев на языке PHP

Задание:

Реализовать форму

PHP Проверка каждой переменной

Имя:

E-mail:

Веб Сайт:

Комментарий:

Пол: Женский Мужской

Ваш ввод:

Эталон ответа:

```
<!DOCTYPE HTML>
<html>
<head>
</head>
<body>
<h1>PHP Проверка каждой переменной</h1>

<?php
// определите переменные и задайте пустые значения
$name = $email = $gender = $comment = $website = "";

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $name = test_input($_POST["name"]);
    $email = test_input($_POST["email"]);
    $website = test_input($_POST["website"]);
    $comment = test_input($_POST["comment"]);
```

```

$gender = test_input($_POST["gender"]);
}

function test_input($data) {
    $data = trim($data);
    $data = stripslashes($data);
    $data = htmlspecialchars($data);
    return $data;
}
?>
<h2>Пример проверки формы PHP</h2>
<form method="post" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);?>">
Имя: <input type="text" name="name">
<br><br>
E-mail: <input type="text" name="email">
<br><br>
Веб Сайт: <input type="text" name="website">
<br><br>
Комментарий: <textarea name="comment" rows="5" cols="40"></textarea>
<br><br>
Пол:
<input type="radio" name="gender" value="Женский">Женский
<input type="radio" name="gender" value="Мужской">Мужской
<br><br>
<input type="submit" name="submit" value="Отправить">
</form>
<?php
echo "<h2>Ваш ввод:</h2>";
echo $name;
echo "<br>";
echo $email;
echo "<br>";
echo $website;
echo "<br>";
echo $gender;
?>

</body>
</html>

```

Варианты:

PHP Contact Form

***Name**

***Email**

***Subject**

***Message**

***Human Verification**

2 + 5 =

Everest Forms

Just another WordPress site

Simple Contact Form Classic Contact Form **Advanced Contact Form**

Advanced Contact Form

First Name *

Last Name

Email *

Subject *

Website

Checkboxes

- First Choice
- Second Choice
- Third Choice

Date

Message

Contact form

Name:

E-mail address: *

Phone:

Recipient:

Message: *

Hi there!

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Attachments:

test.zip

No file selected.

(Allowed file types: PDF, ZIP - Maximum allowed size: 2M)

Send a copy to yourself

(* Required fields.)

Практическая работа № 10. «Загрузка файла на сервер из веб-формы средствами PHP»

Цель практической работы: Изучить некоторые приемы взаимодействия с сервером
Задание:

1. Создать форму регистрации пользователя, позволяющую передать на сервер (метод GET) следующие данные:
 - Имя пользователя (обязат.)
 - Логин (обязат.)
 - Пароль и подтверждение пароля (обязат.)
 - Адрес электронной почты (обязат.)
 - Пол (не обязат.)
 - Дата рождения (не обязат.)
 - Страна и город (не обязат.)
2. Написать javascript, выполняющий проверку правильности заполнения полей регистрации перед отправкой на сервер.
3. Написать серверный php-скрипт, принимающий регистрационные данные и отображающий их на веб-странице.
4. Изменить метод отправки данных из формы на POST, внести соответствующие изменения в скрипты и сравнить результаты выполнения.
5. Создать форму загрузки от 1-го до 5-ти файлов на сервер и написать php-скрипт, принимающий эти файлы. Скрипт должен выводить сообщение об успешной загрузке или об ошибке в случае неудачи.

Эталон ответа:

```
<?php
if ($_GET) {
foreach ($_GET as $k=>$v) echo "$k=$v<br>";
}
?>
```

Для формы из [листинга 1](#) этот скрипт выведет нечто такое:

```
username=User
passwd=verysecure
sex=male
email=e@mail
subscribe=1
invisible=hiddendata
```

Листинг 7. Получение значения конкретного элемента массива \$_GET

```
<?php
$user = "anonymous";
// isset — проверить, установлена ли переменная
// strip_tags — удалить все HTML-теги из полученной строки
if (isset($_GET["username"])) $user = strip_tags($_GET["username"]); echo $user; ?>
```

Приведенных примеров достаточно, чтобы понять, что вся работа с массивом \$_GET (или \$_POST) аналогична использованию обычных массивов и переменных в PHP.

Дополнительно: Обработку данных, переданных методом GET, можно также выполнять путем разбора строки запроса, которая доступна из [переменной окружения](#) \$_SERVER["QUERY_STRING"].

Информация о файлах, переданных на сервер, сохраняется в массиве \$_FILES. Работа с ним несколько отличается от обработки \$_GET или \$_POST, поскольку \$_FILES - двумерный. Т.е. каждый его элемент сам является массивом, содержащим ряд свойств загруженного файла. Содержимое массива \$_FILES для некоторого файла может выглядеть примерно так (получено 2 файла):

```
// Вывод команды print_r($_FILES) Array (
```

```

[file1] => Array (
[name] => asoiu.gif // исходное имя файла [type] => image/gif // тип содержимого
[tmp_name] => /tmp/phpQJBrgA // временное имя в каталоге tmp
[error] => 0 // код ошибки (0 - если нет ошибок)
[size] => 34081 // размер в байтах
)
[file2] => Array (
[name] => web-service.gif
[type] => image/gif
[tmp_name] => /tmp/phpSBq1D23
[error] => 0
[size] => 3007
)
)
)

```

Варианты:

1. Написать форму для загрузки файла на сервер, в свою рабочую папку.
2. Разработать форму ввода адреса web-страницы. При нажатии кнопки вывести в окно браузера заголовок этой страницы.
3. Написать скрипт для получения списка файлов в заданной директории. Для этого разработать форму ввода полного пути к каталогу. При нажатии кнопки вывести содержимое указанного каталога в окне браузера

Практическая работа № 11. «Создание счетчика посещений веб-страницы средствами PHP»

Цель практической работы:

Задание:

На стороне клиента в поля формы заносятся имя и возраст, в зависимости от возраста возвращаются разные приветствия (рассматриваются два варианта: для методов get и post)

Эталон ответа:

HTML-документ, содержащий форму:

```

<html>
<form action=http://localhost/cgi/hello.exe method=get>
<p>ИМЯ<input type=text name=name>
<p>ВОЗРАСТ<input type=text name=age>
<p><input type=submit>
</form>

```

CGI-приложение (файл hello.cpp)

```

#include <iostream.h>
void main()
{
int age;
char *name;
char *query_string=getenv("QUERY_STRING");

//query_string="name=Maria&age=18"
//из строки извлекаются подстроки "Maria" и "18"
//и присваиваются переменным name и age соответственно

cout<<"Content-type: text/html\n\n";
cout<<"<html>";

```

```
if(age<=16) cout<<"Привет, ";
if(age>16) cout<<"Здравствуйте, ";
cout<<name<<"</html>";
}
```

Варианты:

Вариант 1

CGI.

Проверка знания правил дорожного движения и текстовый счетчик посещения страницы.

RНР

.

Проверка знаний правил дорожного движения с записью результатов прохождения теста разными пользователями в файл и вывода результатов в виде диаграммы.

Вариант 2

CGI.

Проверка знания таблицы умножения и текстовый счетчик посещения страницы.

RНР

.

Проверка знания таблицы умножения с записью результатов прохождения теста разными пользователями в файл и вывода результатов в виде диаграммы.

Вариант 3

CGI.

Любой психологический тест и текстовый счетчик посещения страницы.

RНР

.

Любой психологический тест и вывод результатов в виде таблицы с сортировкой по убыванию или возрастанию.

При повторном входе пользователя вывод его предыдущего результата

Вариант 4

CGI.

Проверка знания языка

HTML

и текстовый счетчик посещения страницы.

RНР

.

Проверка знания языка

HTML

с тематическими вопросами и вывод результатов с процентами успешных ответов по каждой теме.

Вариант 5

CGI

.

<p>Экзамен по дисциплине и текстовый счетчик посещения страницы. PHP</p> <p>.</p> <p>Экзамен по дисциплине и в результате вывод не только оценки, но и перечня тем с большинством неправильных ответов. Вывод результата по темам в виде диаграммы</p>
<p>Вариант 6 CGI</p> <p>Проверка знания словарного запаса (англ. или нем. или франц.) и текстовый счетчик посещения страницы. PHP</p> <p>.</p> <p>Проверка знания словарного запаса (англ. или нем. или франц.) с записью результатов прохождения теста разными пользователями в файл и вывода результатов в виде диаграммы</p>
<p>Вариант 7 CGI</p> <p>.</p> <p>Проверка знания таблицы умножения и текстовый счетчик посещения страницы. PHP</p> <p>.</p> <p>Выбор учебных курсов из предлагаемого перечня и формирование расписания занятий. Первоначально предлагается перечень всех курсов и все свободные дни недели. Затем выбранный курс из списка удаляется, и день недели помечается как занятый выбранным курсом. Скрипт прекращает работу, когда все дни недели заняты и расписание сохраняется в файле, который можно скачать по ссылке.</p>
<p>Вариант 8 CGI. Любой психологический тест и текстовый счетчик посещения страницы. PHP</p> <p>.</p> <p>Проверка таблицы умножения с ограничением по времени на каждый ответ</p> <p>.</p> <p>Вывод итогового результата для разных пользователей в виде диаграммы</p>
<p>Вариант 9 CGI</p>

.
Проверка знания языка
HTML
и текстовый счетчик посещения страницы.
PHP

.
Вывод отдельных слов иностранного языка, из которых нужно построить фразу по правилам. Рядом с каждым словом поле, в котором можно определить порядковый номер слова во фразе. Предусмотреть кнопку помощи для решения задачи. Предусмотреть вывод итогового о результата

Практическая работа № 12. «Создание индивидуального счетчика посещений средствами»
Цель практической работы: Изучить возможности языка PHP для генерации динамических страниц. Разработать интерфейс доступа к базе данных на языке PHP.

Задание:

Установить WEB-сервер Apache2 с поддержкой модуля PHP.

Разработать WEB интерфейс взаимодействия сайта с базой данных.

В качестве СУБД можно использовать POSTGRES или MySQL.

Эталон ответа:

Счетчик будет учитывать количество обновлений страницы. При каждом обновлении страницы он будет увеличиваться на 1, а при закрытии браузер – обнуляется.

```
session_start();
```

```
/*
```

Переменная \$_SESSION['counter'] будет нашим счетчиком. Если скрипт запускается первый раз — она будет пуста, присвоим ей единицу. Если не первый раз — тогда прибавим единицу.

```
*/
```

```
if ( !isset($_SESSION['counter']) )
```

```
{
```

```
$_SESSION['counter'] = 1;
```

```
}
```

```
else
```

```
{
```

```
$_SESSION['counter'] = $_SESSION['counter'] + 1;
```

```
}
```

```
//Выведем значение счетчика:
```

```
echo 'Вы обновили эту страницу ' . $_SESSION['counter'] . ' раз!';
```

```
/*
```

Обновите страницу несколько раз, посмотрите на то, как увеличивается значение переменной. Затем закройте браузер и откройте снова — убедитесь в том, что переменная обнулилась!

```
*/
```

Варианты:

Создать счетчик посещений сайта.
Счетчик должен быть графическим. На нем должна быть отображена информация о количестве посещений за день, за неделю и за месяц.

Создать счетчик посещений сайта.
Счетчик должен быть графическим. На нем должна быть отображена информация о количестве посещений минимум с 3-х различных браузеров.

Создать счетчик посещений сайта.
Счетчик должен быть графическим. На нем должна быть отображена информация о количестве уникальных и неуникальных посещений.
Под неуникальными посещениями подразумеваются посещения с того IP-адреса, с которого уже заходили.

Создать счетчик посещений сайта зарегистрированными пользователями.
Данные о зарегистрированных пользователях хранятся в базе данных (логин для каждого пользователя уникальный).
Счетчик должен быть графическим. На нем должна быть отображена информация о количестве посещений зарегистрированными и незарегистрированными пользователями.
Пользователь получает сообщение об успешной/не успешной регистрации и для него отображается счетчик.

Практическая работа № 13. «Создание базы данных программными средствами рНрМуAdmin»

Цель практической работы: Ознакомиться с инструментом создания баз данных рНрМуadmin

Задание:

Проектирование базы данных Имя базы данных “test” По умолчанию у нас имя сервера localhost , имя пользователя root , пароль- пустая строка. В базе данных содержится одна таблица workers

Эталон ответа:

i d	name	ag e	salary
1	Дима	23	400
2	Петя	25	500
3	Вася	23	500
4	Коля	30	1000
5	Иван	27	500
6	Кирил л	28	1000

Варианты:

1. Необходимо создать две таблицы. Первая хранит в себе сведения о фирмах, а именно название фирмы (до 250 символов). Вторая – сведения об автомобилях, производимых фирмами: название автомобиля (до 150 символов), его стоимость (целое число) и ссылка на фирму. Далее вывести сведения о фирмах и автомобилях в виде двухуровневого списка. Первый уровень содержит названия фирм в алфавитном порядке. На втором расположены сведения об автомобилях для данной фирмы, отсортированные по названиям автомобилей в алфавитном порядке. Все данные

страница должна извлечь из базы одним запросом, т. е. нельзя сделать запрос чисто для фирм, а потом для каждой фирмы извлекать дополнительным запросом сведения об автомобилях. Выглядеть страница должна так

```
1. Ford.
  1. Name - extended. Cost - 2500.
  2. Name - T-model. Cost - 2700.
2. Mercedes.
  1. Name - advanced. Cost - 2800.
  2. Name - Basic. Cost - 2900.
```

2. Необходимо создать две таблицы. Первая хранит в себе сведения о фирмах, а именно название фирмы (до 100 символов). Вторая – сведения об автомобилях, производимых фирмами: название автомобиля (до 150 символов), его стоимость (целое число) и ссылка на фирму. Далее вывести сведения о фирмах и автомобилях в виде двухуровневого списка. Первый уровень содержит названия фирм в алфавитном порядке. На втором расположены сведения об автомобилях для данной фирмы, отсортированные по стоимости автомобилей в убывающем порядке. Все данные страница должна извлечь из базы одним запросом, т. е. нельзя сделать запрос чисто для фирм, а потом для каждой фирмы извлекать дополнительным запросом сведения об автомобилях. Выглядеть страница должна так:

```
* Ford.
  1. Name - T-model. Cost - 2700.
  2. Name - extended. Cost - 2500.
* Mercedes.
  1. Name - Basic. Cost - 2900.
  2. Name - advanced. Cost - 2800.
```

3. Необходимо создать две таблицы. Первая хранит в себе сведения о фирмах, а именно название фирмы (до 150 символов). Вторая – сведения об автомобилях, производимых фирмами: название автомобиля (до 200 символов), его стоимость (целое число) и ссылка на фирму. Далее вывести сведения о фирмах и автомобилях в виде одноуровневого списка. Каждый элемент списка должен содержать сведения о фирме, названии модели и стоимости. Отсортировывают элементы по названию фирмы и автомобиля. Выглядеть страница должна так:

```
1. Firm - Ford. Name - extended. Cost - 2500.
2. Firm - Ford. Name - T-model. Cost - 2700.
3. Firm - Mercedes. Name - advanced. Cost - 2800.
4. Firm - Mercedes. Name - Basic. Cost - 2900.
```

4. Необходимо создать две таблицы. Первая хранит в себе сведения о фирмах, а именно название фирмы (до 250 символов). Вторая – сведения об автомобилях, производимых фирмами: название автомобиля (до 150 символов), его стоимость (целое число) и ссылка на фирму. Далее вывести сведения о фирмах и автомобилях в виде таблицы из двух столбцов. Сведения о каждой фирме даны в отдельных строках, объединяющих сразу два столбца. Эти строки отсортировывают в алфавитном порядке. Далее для каждой фирмы следуют сведения об автомобилях, отсортированные по названиям автомобилей. Все данные страница должна извлечь из базы одним запросом, т.е. нельзя сделать запрос чисто для фирм, а потом для каждой фирмы извлекать дополнительным

запросом сведения об автомобилях. Выглядеть страница должна так:

```

-----
|Model   |Cost |
-----
|  Ford  |     |
-----
|extended|2500 |
-----
|T-model |2700 |
-----
| Mercedes|     |
-----
|advanced|2800 |
-----
|Basic   |2900 |
-----

```

5. Необходимо создать две таблицы. Первая хранит в себе сведения о фирмах, а именно название фирмы (до 250 символов), вторая – сведения об автомобилях, производимых фирмами: название автомобиля (до 150 символов), его стоимость (целое число) и ссылка на фирму. Необходимо вывести сведения о фирмах и автомобилях в виде таблицы из трех столбцов. Сведения отсортировывают по названию фирмы (в алфавитном порядке) и стоимости автомобиля (в порядке убывания). Выглядеть страница должна так:

```

-----
|Firm    |Model  |Cost |
-----
|Ford    |T-model|2700 |
-----
|Ford    |extended|2500 |
-----
|Mercedes|Basic  |2900 |
-----
|Mercedes|advanced|2800 |
-----

```

6. Необходимо создать две таблицы для хранения сведений об отелях. Первая хранит сведения о фирмах – название фирмы (до 150 символов). Вторая – сведения об отелях: название отеля (до 150 символов), стоимость проживания в отеле и название страны, где расположен отель (до 150 символов), а также ссылка на фирму. Сведения об отелях выводятся в виде двухуровневого списка. На первом уровне должны располагаться названия стран в алфавитном порядке. На втором – сведения об отелях для данной страны, отсортированные по названию фирмы и отеля в алфавитном порядке. Все данные страница должна извлечь из базы одним запросом, т. е. нельзя сделать запрос чисто для стран, а потом для каждой страны извлекать дополнительным запросом сведения об отелях. Выглядеть страница должна так:

```

1. Egypt.
  1. Firm - Arch. Hotel - Awesome. Cost - 1000.
  2. Firm - Arch. Hotel - Best. Cost - 800.
  3. Firm - Tour. Hotel - Epic. Cost - 100.
2. France.
  1. Firm - Arch. Hotel - Ritz. Cost - 1200.
  2. Firm - Arch. Hotel - Zimmer. Cost - 700.
  3. Firm - Tour. Hotel - Fail. Cost - 50.

```

7. Необходимо создать две таблицы для хранения сведений об отелях. Первая хранит сведения о фирмах – название фирмы (до 150 символов). Вторая – сведения об отелях: название отеля (до 150 символов), стоимость проживания в отеле и название страны, где расположен отель (до 150 символов), а также ссылка на фирму. Сведения об отелях выводятся в виде двухуровневого списка. На первом уровне должны располагаться

названия фирм в алфавитном порядке. На втором – сведения об отелях для данной фирмы, отсортированные по названию страны и отеля в алфавитном порядке. Все данные страница должна извлечь из базы одним запросом, т. е. нельзя сделать запрос чисто для фирм, а потом для каждой фирмы извлекать дополнительным запросом сведения об отелях. Выглядеть страница должна так:

```
* Arch.
  1. Country - Egypt. Hotel - Awesome. Cost - 1000.
  2. Country - Egypt. Hotel - Best. Cost - 800.
  3. Country - France. Hotel - Ritz. Cost - 1200.
  4. Country - France. Hotel - Zimmer. Cost - 700.
* Tour.
  1. Country - Egypt. Hotel - Epic. Cost - 100.
  2. Country - France. Hotel - Fail. Cost - 50.
```

8. Необходимо создать две таблицы для хранения сведений об отелях. Первая хранит сведения о фирмах – название фирмы (до 150 символов). Вторая – сведения об отелях: название отеля (до 150 символов), стоимость проживания в отеле и название страны, где расположен отель (до 150 символов), а также ссылка на фирму. Необходимо вывести сведения об отелях в виде таблицы из трех столбцов. Сведения о фирмах должны идти в алфавитном порядке в строках, объединяющих все три столбца. Для каждой фирмы далее идут строки со сведениями об отелях, отсортированные по названию страны (в алфавитном порядке), и стоимости отеля (в порядке возрастания). Все данные страница должна извлечь из базы одним запросом, т. е. нельзя сделать запрос чисто для фирм, а потом для каждой фирмы извлекать дополнительным запросом сведения об отелях. Выглядеть страница должна так:

Country	Hotel	Cost
Arch		
Egypt	Best	800
Egypt	Awesome	1000
France	Zimmer	700
France	Ritz	1200
Tour		
Egypt	Epic	100
France	Fail	50

9. Необходимо создать две таблицы для хранения сведений об отелях. Первая хранит сведения о фирмах – название фирмы (до 150 символов). Вторая – сведения об отелях: название отеля (до 150 символов), стоимость проживания в отеле и название страны, где расположен отель (до 150 символов), а также ссылка на фирму. Сведения об отелях выводятся в виде таблицы из трех столбцов. Сведения о странах должны идти в алфавитном порядке в строках, объединяющих все три столбца. Для каждой страны далее идут строки со сведениями об отелях, отсортированные по названию фирмы (в алфавитном порядке) и стоимости отеля (в порядке возрастания). Все данные страница должна извлечь из базы одним запросом, т. е. нельзя сделать запрос чисто для стран, а потом для каждой страны извлекать дополнительным запросом сведения об отелях. Выглядеть страница должна так:

Firm	Hotel	Cost
Egypt		
Arch	Best	800
Arch	Awesome	1000
Tour	Epic	100
France		
Arch	Zimmer	700
Arch	Ritz	1200
Tour	Fail	50

10. Необходимо создать две таблицы для хранения сведений об отелях. Первая хранит сведения о фирмах – название фирмы (до 150 символов). Вторая – сведения об отелях: название отеля (до 150 символов), стоимость проживания в отеле и название страны, где расположен отель (до 150 символов), а также ссылка на фирму. Сведения об отелях выводятся в виде таблицы из четырех столбцов. Сведения отсортировывают по названиям стран, фирм и отелей. Выглядеть страница должна так:

Country	Firm	Hotel	Cost
Egypt	Arch	Awesome	1000
Egypt	Arch	Best	800
Egypt	Tour	Epic	100
France	Arch	Ritz	1200
France	Arch	Zimmer	1700
France	Tour	Fail	150

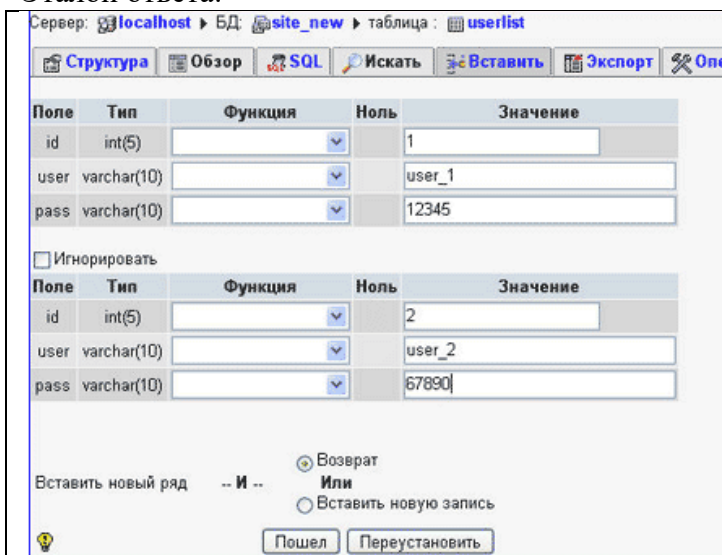
Практическая работа № 14. «Организация связи между клиентской частью веб-приложения и базой данных средствами PHP»

Цель практической работы: научиться создавать базы данных на PHP

Задание:

- 1) Построить физическую схему БД.
- 2) Создать скрипт для подключения и создания одной из таблиц, а также добавления тестовых записей в таблицу.
- 3) Создать страницу и php-скрипт для выполнения одного из действий в соответствии с номером варианта (реализовать можно это на одной странице):
 - добавления новой записи в таблицу;
 - редактирования записи в таблице;
 - удаления записи из таблицы.

Эталон ответа:



Скрипт обработки загружаемых данных.

```

Перед подключение к БД нужно получить переменные посланные методом GET
$stud=$_GET['student'];
$name=$_GET['username'];

```

```

<?php
/* Определяем значения переменным */
$hostname="localhost";
$username="Ваш логин";

```

```

$password="Ваш_пароль";
/* Имя базы данных */
$dbName="Имя_вашей_базы_данных";
/* Имя таблицы */
$table="Имя_таблицы";
/* Создать соединение с MySQL*/
MYSQL_CONNECT($hostname,$username,$password) OR DIE("Не могу
подсоединиться");
/* Выбор БД */
MYSQL_SELECT_DB($dbName) or die("Не могу выбрать БД");
/* Введение информации в БД */
$query="INSERT INTO $table VALUES ('$user_name','$phone','$img')";
$result=MYSQL_QUERY($query);
/* Закрыть соединение */
MYSQL_CLOSE();
print "Запись введена в БД! <br>";
?>

```

Скрипт обработки поискового запроса

```

<?php
$hostname="localhost";
$username="Ваш_логин";
$password="Ваш_пароль";
$dbName="Имя_вашей_базы_данных";
$table="Имя_таблицы";
/* Создать соединение с MySQL*/
mysql_connect($hostname,$username,$password) or die("No connect");
mysql_select_db("$dbName") or die("No select DB");
/* Выбрать людей, имена которых начинаются на $usr_name */
$query="select * from $table where user_name like '$user_name%'";
$result=mysql_query($query);
/* Сколько нашлось таких сотрудников? */
$number=mysql_numrows($result);
/* Распечатка выбранных сотрудников */
$i=0;
if ($number == 0)
{
print "Нет данных о $user_name в БД <br>";
}
elseif ($number > 0)
{
while ($i < $number)
{
/* Присвоение переменным значений полей user_name, phone, img для i-й
строки */
$user_name=mysql_result($result,$i,"user_name");
$phone=mysql_result($result,$i,"phone");
$img=mysql_result($result,$i,"img");
/* Вывод найденных значений */
print "<br><img src='foto/$img' height=130><br>";
print "<u> Имя:</u> <br>";
print "$user_name <br><br>";
}
}

```

```

print "<u> Телефон: </u><br>";
print "$phone <br><br>";
$i++;
print "<hr>";
}
}
?>

```

Скрипт для удаления данных

```

<?php
$hostname="localhost";
$username="Ваш_логин";
$password="Ваш_пароль";
$dbName="Имя_вашей_базы_данных";
$userTable="Имя_таблицы";
mysql_connect($hostname,$username,$password) or die("No connect");
mysql_select_db("$dbName") or die("No select DB");
/* Удаление записи для поля user_name */
$query="delete from $userTable where user_name='$user_name'";
$result=mysql_query($query);
print "Запись удалена <br>";
?>

```

Скрипт редактирования данных

```

<?php
$hostname="localhost";
$username="Ваш_логин";
$password="Ваш_пароль";
$dbName="Имя_вашей_базы_данных";
$userTable="Имя_таблицы";
mysql_connect($hostname,$username,$password) or die("No connect");
mysql_select_db("$dbName") or die("No select DB");
/* Выбор записи по полю user_name */
$query="select * from $userTable where user_name='$user_name'";
$result=mysql_query($query);
/* Определение существования выбранной записи */
$number=mysql_numrows($result);
if ($number==0)
{
print "Нет такой записи!<br>";
}
else
{
/* Если запись выбрана из базы данных, вывести все ее поля в форму для
редактирования и удалить ее */
$phone=mysql_result($result,0,"phone");
$img=mysql_result($result,0,"img");
print "<img src='foto/$img' height=130><br>";
/* Удаление записи */
$query="delete from $userTable where user_name='$user_name'";
$result=mysql_query($query);
/* Создание формы для редактирования */

```



```

print "<form method=post action='form5.php'>";
print "<br><b>Введите Ваше имя:</b>";
print "<br><input name='user_name' value='$user_name' size=30>";
print "<br><br><b>Введите номер вашего телефона:</b>";
print "<br><input name='phone' value='$phone' size=10>";
print "<br><br><b>Введите имя фото лат.буквами:</b>";
print "<br><input name='img' value='$img' size=20>";
print "<br><br><input type='submit' value='Зарегистрировать'>";
print "</form>";
}
?>

```

Скрипт для загрузки отредактированных данных

```

<?php
hostname="localhost";
$username="Ваш_логин";
$password="Ваш_пароль";
$dbName="Имя_вашей_базы_данных";
$userTable="Имя_таблицы";
/* Создать соединение */
MYSQL_CONNECT($hostname,$username,$password) OR DIE("Не могу
подсоединиться");
/* Выбор БД */
MYSQL_SELECT_DB($dbName) or die("Не могу выбрать БД");
/* Введение информации в БД */
$query="INSERT INTO $userTable VALUES ('$user_name','$phone','$img)";
$result=MYSQL_QUERY($query);
/* Закрывает соединение */
MYSQL_CLOSE();
print "Запись введена в БД! <br>";
?>

```

Варианты:

создайте базу данных «Поликлиника», состоящую из 2-3 таблиц;
 организуйте связи между таблицами БД;
 заполните таблицы информацией;
 отобразите данные таблицы на странице;
 осуществите выборку данных из БД.

создайте базу данных «Деканат», состоящую из 2-3 таблиц;
 организуйте связи между таблицами БД;
 заполните таблицы информацией;
 отобразите данные таблицы на странице;
 осуществите выборку данных из БД.

создайте базу данных «Школа», состоящую из 2-3 таблиц;
 организуйте связи между таблицами БД;
 заполните таблицы информацией;
 отобразите данные таблицы на странице;
 осуществите выборку данных из БД.

создайте базу данных «Ателье», состоящую из 2-3 таблиц;
 организуйте связи между таблицами БД;
 заполните таблицы информацией;
 отобразите данные таблицы на странице;
 осуществите выборку данных из БД.

создайте базу данных «Мастерская», состоящую из 2-3 таблиц;

организовать связи между таблицами БД;
заполните таблицы информацией;
отобразите данные таблицы на странице;
осуществите выборку данных из БД.

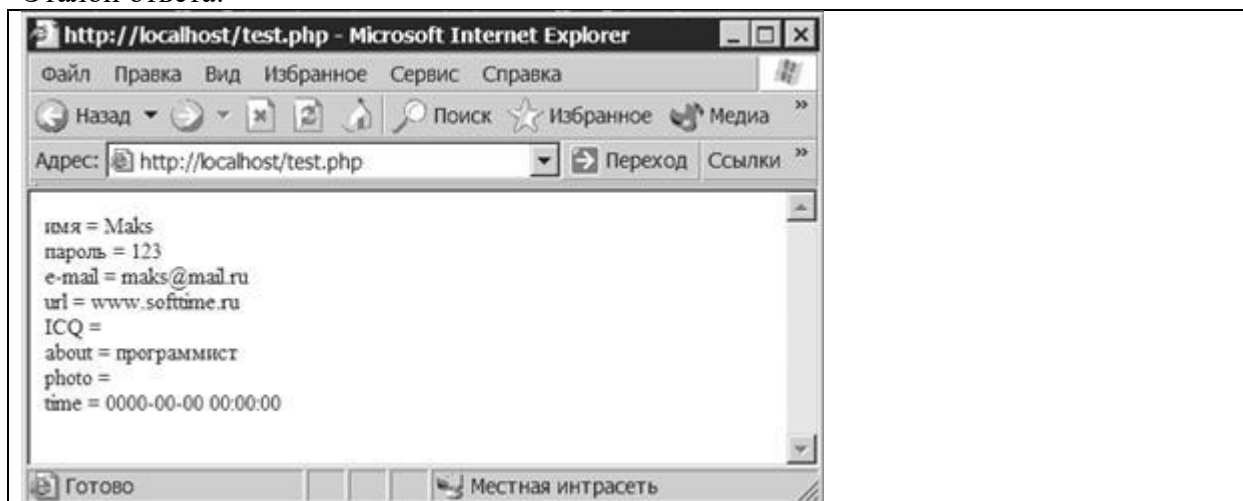
Практическая работа № 15 «Реализация запроса на выборку информации из базы данных по заданному критерию средствами PHP и MySQL»

Цель практической работы: Изучить теоретические основы разработки PHP-сценариев с использованием сервера баз данных MySQL

Задание:

1. создать БД согласно варианту задания. заполнить таблицу данными
2. создать HTML-документ с формой, служащей для ввода исходных данных и передачи их PHP-скрипту
3. создать PHP-скрипт для получения из БД информации согласно указанным в задании запросам.
4. Организовать в скрипте вывод результата в Web-браузер пользователя.
5. Предусмотреть операторы диагностики и обработки исключительных ситуаций

Эталон ответа:



Варианты:

1. *Базы данных:* Автозаправочные станции = {Название АЗС, объем бензина (л), город }.
В PHP – сценариях необходимо реализовать следующие запросы к таблице из этой базы данных:

- Найти суммарный объем бензина на бензоколонках города, название которого ввел пользователь;
- Найти бензоколонки, в названии которых присутствует строка, введенная пользователем;
- Найти бензоколонки, в названии которых количество бензина больше введенного пользователем значения.

2. *Базы данных:* Детские сады = {Название детского сада, количество детей, количество работников, средняя зарплата }.
Необходимо реализовать в PHP – сценариях следующие запросы к таблице из этой базе данных:

- Найти детские сады, в которых средняя зарплата меньше введенного пользователем значения
- Найти детские сады, в названии которых присутствует строка, введенная пользователем;

<ul style="list-style-type: none"> · Найти количество работников и количество детей в детском саду, название которого ввел пользователь.
<p>3. <i>Базы данных:</i> Программные Продукты (ПП) = {Название ПП, стоимость, Назначение фирмы-разработки }.</p> <p>В РНР – сценариях необходимо реализовать следующие запросы к таблице из этой базы данных:</p> <ul style="list-style-type: none"> · Найти самый дорогой ПП фирмы, название которого ввел пользователь; · Найти все программные продукты фирмы, название которой ввел пользователь; · Найти программные продукты, в названии фирмы-разработчика которых присутствует строка, введенная пользователем.
<p>4. <i>Базы данных:</i> Продуктовые магазины (ПМ) = {Название ПМ, город, Год, Объем продаж (тыс. руб)}.</p> <p>В РНР – сценариях необходимо реализовать следующие запросы к таблице из этой базы данных:</p> <ul style="list-style-type: none"> · Найти магазины, у которых объем продаж в 2009 г. превысил указанное пользователем значение; · Найти магазины города, название города которого ввел пользователь; · найти суммарный объем продаж в магазинах, в названии которых присутствует строка, введенная пользователем.
<p>5. <i>Базы данных:</i> Компьютеры = {Номер компьютера, Объем свободного места на жестких дисках, Номера дисплейного класса}.</p> <p>В РНР – сценариях необходимо реализовать следующие запросы к таблице из этой базы данных:</p> <ul style="list-style-type: none"> · найти компьютеры, у которых свободное место на жестких дисках превышает введенное пользователем значение · найти дисплейные классы, в которых количество компьютеров больше или равно введенному пользователем числу; · найти компьютеры дисплейного класса, номер которого задал пользователь.
<p>6. <i>Базы данных:</i> Книги = {Название книги, Типография, Количество страниц, Год издания }</p> <p>В РНР – сценариях необходимо реализовать следующие запросы к таблице из этой базы данных:</p> <ul style="list-style-type: none"> · найти названия книг, напечатанных в том году, который ввел пользователь; · найти типографии, которые выпустили n и более различных книг, где n – это число, введенное пользователем; · найти книги, имеющие количество страниц от A до B, где A и B – введенные пользователем числа.
<p>7. <i>Базы данных:</i> Документы = {Наименование документа, количество страниц, Название кафедры}</p> <p>В РНР – сценариях необходимо реализовать следующие запросы к таблице из этой базы данных:</p> <ul style="list-style-type: none"> · найти общее количество документов кафедры, название которой ввел пользователь · найти документы, имеющие количество страниц от A до B, введенные пользователем числа. · Найти кафедры, на которых разработаны документы, в названии которых содержится строка, введенная пользователем.
<p>8. <i>Базы данных:</i> Изделия = {Наименование изделия, Количество входящих в него</p>

деталей, Общая трудоемкость изготовления изделия (час), Стоимость}
В РНР – сценариях необходимо реализовать следующие запросы к таблице из этой базы данных:

- Найти изделия, у которых количество деталей больше или равно введенному пользователем числу;
- Найти названия и трудоемкость изготовления изделий, стоимость которых меньше введенного пользователем значения;
- Найти информацию по изделиям, в названии которых присутствует строка, введенная пользователем.

9. *Базы данных:* Кабинеты = {Номер кабинета, Номер корпуса, Количество компьютеров, Количество парт}

В РНР – сценариях необходимо реализовать следующие запросы к таблице из этой базы данных:

- Найти кабинет, в котором количество парт и количество компьютеров превышают введенные пользователем значения;
- Найти количество компьютеров в корпусе, номер которого ввел пользователь;
- Найти корпус, в котором суммарное количество компьютеров не меньше, чем значение, введенное пользователем.

10. *Базы данных:* Участники конференции = {ФИО, название, доклада, город, Название секции}

В РНР – сценариях необходимо реализовать следующие запросы к таблице из этой базы данных:

- Найти участников, которые приехали из города, название которого ввел пользователь;
- Найти секции с количеством докладчиков большим или равными числу, введенному пользователем;
- Найти количество участников секции, для которой пользователь ввел название или его часть.

11. *Базы данных:* Автомобили = {Модель автомобиля, цена, название завода-изготовителя}

В РНР – сценариях необходимо реализовать следующие запросы к таблице из этой базы данных:

- Найти автомобили, цена которых не превышает введенное пользователем значение;
- Найти заводы-изготовители, количество выпускаемых моделей которых не меньше, чем число, введенное пользователем;
- Найти информацию по автомобилям, в названии которых присутствует строка, введенная пользователем.

Практическая работа № 16 «Применение технологии AJAX»

Цель практической работы: изучить основы технологии AJAX. Написать простейшие клиентское и серверное приложение обменивающиеся асинхронными данными.

Задание:

Сконфигурировать метод ajax

Эталон ответа:

```
$.ajax( '/data/people.json', {  
  type: 'GET',  
  dataType: 'json',  
  success: function( resp ) {
```

```

    console.log( resp.people );
  },
  error: function( req, status, err ) {
    console.log( 'что-то пошло не так', status, err );
  }
});

```

Варианты:

Задание 1. Передайте на сервер два числа методом POST. Имена чисел – num1 и num2. Цель — файл a1.php. Если все выполнено верно, то будет возвращена строка с суммой чисел, если нет – ошибка. Способ передачи – AJAX.

Задание 2. Передайте на сервер свой год рождения в формате XXXX. Передачу осуществляйте методом POST, имя переменной year, имя целевого файла – a2.php. Если все выполнено верно, то сервер вернет строку, в которой будет количество прожитых лет, если нет – ошибку. Способ передачи – AJAX.

Задание 3. Создайте форму с radiobutton для выбора пола (мужчина или женщина). Добавьте поле для ввода роста (в сантиметрах). По нажатию кнопки “Идеальный вес”, пошлите запрос на файл a3.php. На сервер передается параметр пол – имя параметра sex, значения man или woman и рост (имя height). Передача осуществляется методом AJAX, с помощью POST.

Если все выполнено правильно, то будет возвращена строка с расчетом идеального веса под заданный пол. В противном случае – ошибка.

Задание 4. Измените предыдущее значение, чтобы рост задавался с помощью ползунка range.

Задание 5. Создайте форму с полями:

fio – поле ввода для имени и фамилии;

email – поле ввода для email адреса

phone – поле ввода для телефона

По нажатию кнопки передайте данные из полей ввода на сервер (mail.php) посредством POST (AJAX). Имена переменных – такие, как указано в списке. Если все сделано верно, то по адресу, указанному в файле mail.php в переменной \$to будет отправлено письмо, если нет – выведена ошибка.

Возвращаемый формат – строка.

Для отправления сообщения в PHP есть функция mail.

```
$to = 'alexlu@mail.ru';
```

```
$subject = 'Вам отправили запрос';
```

```
$message = "Пользователь $fio с email $email и телефоном $phone отправил вам запрос";
```

```
$res = mail ($to, $subject, $message);
```

Задание 6. Пошлите на сервер (goods.php) методом GET артикул товара (art). Сервер реагирует только на 2 артикула (1 или 2). Если все выполнено верно, то сервер пришлет вам JSON строку с описанием товара. После раскодировки (JSON.parse) вы сможете вывести поля:

- name – имя товара
- weight – вес товара
- cost – стоимость товара
- img – ссылка на картинку.

Для работы с массивами стоит использовать функцию array.

Для кодирования данных json можно использовать json_encode()

Практическая работа № 17 «Использование библиотеки jQuery»

Цель практической работы:

научиться использовать библиотеку jQuery

Задание:

Библиотеку jQuery можно скачать с сайта <http://jquery.com>, а можно вставить в документ, используя известные интернет-адреса:

- По адресу <http://code.jquery.com/jquery-latest.js> – доступна всегда последняя версия.
- С Google: <https://developers.google.com/speed/libraries/devguide?hl=ru#jquery> можно загрузить любую из не слишком старых версий. Синтаксис такой: `src="http://ajax.googleapis.com/ajax/libs/jquery/1.8.3/jquery.min.js"`, где 1.8.3 — версия, причём можно указать её приблизительно: 1.8 означает последнюю версию вида 1.8.*, а 1 — последнюю версию вида 1.*. Файл `jquery.min.js` обозначает сжатый код, а `jquery.js` — несжатый, для удобства отладки.
- Либо с Microsoft CDN: `src="http://ajax.aspnetcdn.com/ajax/jQuery/jquery-1.9.1.min.js"`

Эталон ответа:

```
<!DOCTYPE html>
<html>
<head>
<!--если выбираем библиотеку от Microsoft CDN -->
<script src="http://ajax.aspnetcdn.com/ajax/jQuery/jquery-1.9.1.min.js"></script>
<!--если выбираем библиотеку от Google -->
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.8.3/jquery.min.js"></script>
<!--третий вариант -->
<script src="http://code.jquery.com/jquery-latest.js"></script>
</head>
<body>
</body>
</html>
```

Варианты:

Выполнить пример, проверить работу метода `hide()`. Попробовать заменить библиотеку `jquery-latest.js` на другие две, описанные выше, сравнить результаты работы.

Пояснения к скрипту:

```
<script>
//в строке ниже в качестве селектора использован весь объект -документ,
//при этом будет срабатывать jquery-запрос, как только наступит событие полной готовности
//страницы к работе с пользователем и запустится метод ready(),
// при срабатывании которого будет создана следующая функция

$(document).ready(function () {

//созданная функция в свою очередь будет jquery-запросом, который ищет все элементы типа
//button - кнопка, и с их методами click связывает (ассоциирует) действие в виде функции,

$("button").click(function () {

// которая выполнит jquery-запрос, который для всех найденных элементов внутри тега <p>
выполнит метод hide(), т.е. скроет их со страницы

$("p").hide();

//далее закрываем внутреннюю функцию

});

//далее закрываем внешнюю функцию

});
</script>
```

Выполнить пример, проанализировать отличие его от примера в задании 1.

```

<!DOCTYPE html>
<html>
<head>
<script src="http://code.jquery.com/jquery-latest.js">
</script>

<script>
  $(document).ready(function () {
    $("#but1").click(function () {
      $("#test").hide();
    });
    $("#but2").click(function () {
      $("#test").show();
    });
  });
</script>
</head>

<body>
<h2>This is a heading 2</h2>
<p>This is a paragraph.</p>
<p id="test">This is another paragraph.</p>
<button id="but1">Click me</button>
<button id="but2">Click me for show</button>
</body>

</html>

```

Выполнить пример, проанализировать отличие его от примера в задании 2. .class селектор
 Селектор jQuery class находит элементы определенного класса. Для поиска элементов определенного класса указывается перед названием точка, например: `$(".test")`

```

<!DOCTYPE html>
<html>
<head>
<script src="http://code.jquery.com/jquery-latest.js">
</script>

<script>
  $(document).ready(function () {
    $("button").click(function () {
      $(".test").hide();
    });
  });
</script>
</head>
<body>

<h2 class="test">This is a heading</h2>
<p class="test">This is a paragraph.</p>
<p>This is another paragraph.</p>
<button>Click me</button>
</body>
</html>

```

Практическая работа № 19 «Создание сайта на CMS»

Цель практической работы:

Научиться устанавливать CMS;

4) Научиться наполнять сайты контентом;

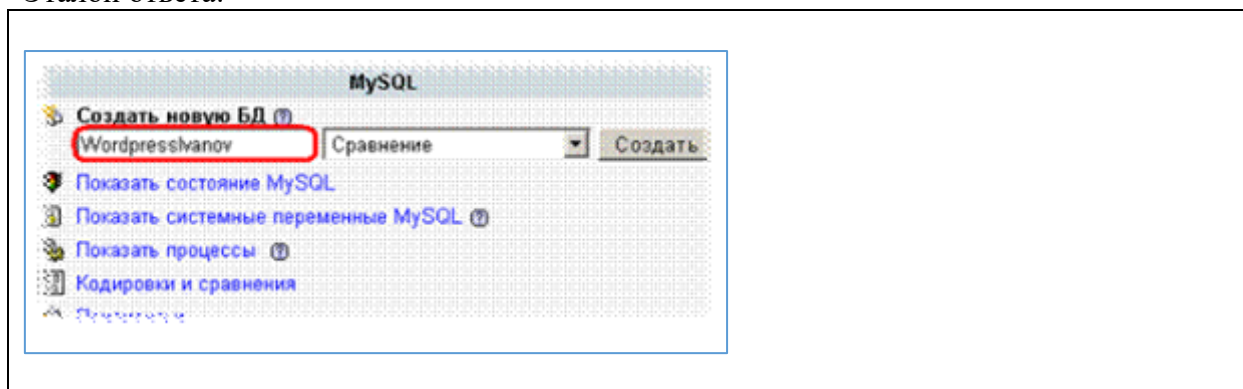
5) Научиться расширять возможности CMS путем установки плагинов;

6) Научиться менять дизайн сайта организованного на CMS.

Задание:

1. Создаем на Денвере собственный новый домен например WordpressIvanov а в ней папку www
2. Качаем инсталляцию WordPress http://ru.wordpress.org/wordpress-2.8.6-ru_RU.zip
3. Распаковываем содержимое папки в архиве в папку C:\WebServers\home\WordPressIvanov\www
4. Перезапускаем Денвер
5. Подготовительный этап: запускаем phpMyAdmin и создаем в нем новую пустую базу данных WordpressIvanov

Эталон ответа:



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ ДАГЕСТАН
Частное профессиональное образовательное учреждение
«Республиканский полипрофессиональный колледж»

Вопросы для дифференцированного зачета
МДК 01.05 Разработка веб-приложений

1. Современные технологии разработки веб-приложений
2. Основные требования, предъявляемые к дизайну графических интерфейсов, способам передачи информации в сети Интернет.
3. Основы PHP: базовый синтаксис, циклы, массивы.
4. Формы. Организация форм средствами HTML. Организация форм средствами веб-программирования. Методы GET, POST.
5. Cookie. HTTP-заголовки ответа сервера. Сессии
6. Обработка файлов PHP: чтение, запись, загрузка на сервер.
7. Работа с базой данных: MySQL, phpMyAdmin
8. Извлечение, редактирование и обновление данных из базы данных средствами PHP.
9. Объектно-ориентированное программирование на PHP
10. PHP + XML, стилевые таблицы XSLT
11. Язык сценариев JavaScript. Объекты в JavaScript
12. Основы jQuery и основы выборки элементов
13. Технология AJAX

Эталон ответа к дифференцированному зачету

1. Современные технологии разработки веб-приложений
Фреймворки: такие как React, Angular, Vue.js, Node.js, Django и т.д. облегчают создание веб-приложений, предоставляя готовые решения для работы с интерфейсом пользователя, обработки входящих запросов
Single Page Applications (SPA): SPA-приложения позволяют обновлять содержимое страницы без перезагрузки браузера, что обеспечивает более плавный и динамический пользовательский опыт.
Progressive Web Apps (PWA): PWA - это веб-приложения, которые объединяют передовые возможности веб-сайтов и мобильных приложений для создания быстрых, надежных и удобных в использовании приложений.
RESTful API: REST (Representational State Transfer) - это популярный стиль архитектуры для разработки веб-сервисов с использованием HTTP протокола.
Serverless Computing: Serverless архитектура позволяет разрабатывать и развертывать веб-приложения без необходимости управления инфраструктурой сервера.

2. Основные требования, предъявляемые к дизайну графических интерфейсов, способам передачи информации в сети Интернет.

Дизайн графических интерфейсов должен удовлетворять следующие основные требования:

1. Понятность и интуитивность: пользователь должен легко понимать, как взаимодействовать с интерфейсом и какие действия нужно предпринять для достижения конкретной цели.
2. Согласованность и целостность: элементы интерфейса должны быть согласованы между собой и образовывать цельное и логичное визуальное решение.
3. Удобство использования: интерфейс должен быть удобным для пользователей всех категорий и уровня навыков.
4. Визуальная привлекательность: дизайн интерфейса должен быть эстетически приятным и привлекательным для пользователя.
5. Соблюдение стандартов и технических требований: дизайн должен быть реализован с учетом технических возможностей и требований платформы, на которой будет работать веб-приложение или сайт.

3. Основы PHP: базовый синтаксис, циклы, массивы.

PHP (Hypertext Preprocessor) - это широко используемый язык программирования на стороне сервера, который часто используется для создания динамических веб-страниц. Вот основные элементы PHP, которые вам нужно знать:

Базовый синтаксис PHP:

Теги PHP: PHP-код обычно заключается в теги `<?php` и `?>`. Например:

```
<?php
    // Код PHP здесь
?>
```

Циклы в PHP:

```
<?php
    for ($i = 0; $i < 5; $i++) {
        echo $i;
    }
    $j = 0;
    while ($j < 5) {
        echo $j;
        $j++;
    }
?>
```

Массивы в PHP позволяют хранить и обрабатывать наборы данных.

```
<?php
    $массив = array(значение1, значение2, значение3);
    $ассоциативный_массив = array("ключ1" => "значение1", "ключ2" => "значение2");

    echo $массив[0]; // Выводит значение1
    echo $ассоциативный_массив["ключ1"]; // Выводит значение1
?>
```

4. Формы. Организация форм средствами HTML.

Формы.

HTML формы позволяют пользователям вводить данные, которые затем могут быть отправлены на сервер для обработки. Элемент `<form>` используется для создания формы, а внутри него могут быть различные элементы для ввода данных, такие как `<input>`, `<textarea>` и `<select>`.

Организация форм средствами веб-программирования

При организации форм средствами веб-программирования обычно используются

серверные технологии, такие как PHP, Python, Ruby, Node.js и другие. Когда пользователь отправляет данные с формы, эти данные обрабатываются на сервере с помощью специальных скриптов, которые обычно выполняются после отправки формы.

Пример:

```
```php
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
 $username = $_POST["username"];
 echo "Привет, $username!";
}
?>
```
```

Методы GET и POST

HTTP протокол поддерживает два основных метода для отправки данных: GET и POST. Метод GET отправляет данные в URL запроса. Данные видны в строке адреса браузера и ограничены по длине. GET обычно используется для получения данных от сервера. Метод POST отправляет данные в теле HTTP запроса. Данные не видны в URL и не имеют ограничений по длине. POST обычно используется для отправки данных на сервер для обработки форм и других действий.

5. Cookie. HTTP-заголовки ответа сервера. Сессии

Cookie - это небольшой фрагмент данных, который веб-сервер отправляет браузеру, а в свою очередь браузер сохраняет его на стороне клиента. Cookie обычно используются для хранения информации о состоянии сеанса, сессионной информации, предпочтениях пользователя и др.

HTTP-заголовки ответа сервера - это часть ответа, который сервер отправляет обратно клиенту после обработки запроса. HTTP-заголовки могут содержать метаданные, управлять кэшированием, устанавливать тип содержимого, управлять перенаправлениями, устанавливать коды состояния и многое другое.

Сессии - это механизм на стороне сервера для отслеживания состояния взаимодействия между клиентом и сервером. Сессия представляет собой уникальный идентификатор, который ассоциируется с каждым пользователем.

6. Обработка файлов PHP: чтение, запись, загрузка на сервер.

1. Чтение файлов:

Для чтения содержимого файла в PHP можно воспользоваться функцией

`file_get_contents()`. Например:

```
```php
$file_content = file_get_contents('file.txt');
echo $file_content;
```
```

2. Запись в файлы:

Для записи данных в файл в PHP используется функция `file_put_contents()`. Например, чтобы записать текст в файл:

```
```php
$data = "Hello, World!";
file_put_contents('output.txt', $data);
```
```

3. Загрузка файлов на сервер:

Для загрузки файла на сервер в PHP используется форма с элементом `

```
```html
<form action="upload.php" method="post" enctype="multipart/form-data">
 <input type="file" name="file">
```
```

```
<input type="submit" value="Upload">
</form>
...

```

Пример скрипта загрузки файла `upload.php`:

```
```php
if ($_FILES['file']['error'] === UPLOAD_ERR_OK) {
 $tmp_name = $_FILES['file']['tmp_name'];
 $upload_path = 'uploads/' . $_FILES['file']['name'];
 move_uploaded_file($tmp_name, $upload_path);
 echo 'File uploaded successfully.';
} else {
 echo 'Error uploading file.';
}
```

```

7. Работа с базой данных: MySQL, phpMyAdmin

Основные задачи, которые можно выполнять с помощью phpMyAdmin:

1. Создание базы данных: Для создания новой базы данных в phpMyAdmin перейдите на вкладку "Базы данных", введите имя новой базы данных и выберите нужную кодировку символов.

2. Создание таблиц: В разделе выбранной базы данных выберите "Создать новую таблицу". Задайте название таблицы, определите количество и типы столбцов, а также их свойства (например, PRIMARY KEY, AUTO_INCREMENT).

3. Импорт и экспорт данных: С помощью phpMyAdmin можно импортировать данные из файла SQL или CSV в базу данных, а также экспортировать данные из таблицы в нужный формат для дальнейшей работы.

4. Запросы SQL: В phpMyAdmin можно выполнять SQL-запросы непосредственно во встроенной консоли. Это дает возможность выполнять различные операции с данными, фильтровать их и получать нужную информацию.

5. Управление пользователями и привилегиями: PhpMyAdmin позволяет создавать новых пользователей, устанавливать им различные привилегии (например, SELECT, INSERT, UPDATE, DELETE) и управлять доступом к базам данных.

6. Изменение структуры таблицы: Вы можете изменять структуру существующей таблицы, добавлять новые столбцы, изменять и удалять существующие.

7. Оптимизация и анализ базы данных: PhpMyAdmin предоставляет инструменты для оптимизации таблиц и баз данных, а также для анализа и проверки их целостности.

8. Резервное копирование и восстановление: Безопасность данных - это один из важнейших аспектов работы с базой данных. PhpMyAdmin позволяет создавать резервные копии баз данных и восстанавливать их в случае необходимости.

8. Извлечение, редактирование и обновление данных из базы данных средствами PHP. Для работы с базами данных в PHP используется расширение PDO (PHP Data Objects) или расширение MySQLi. В данном примере рассмотрим использование PDO для работы с базой данных.

1. Подключение к базе данных:

Для начала необходимо создать подключение к базе данных. Для этого используется следующий код:

```
```php
<?php
$host = 'localhost';
$dbname = 'my_database';
$username = 'my_username';
$password = 'my_password';

```

```

try {
 $pdo = new PDO("mysql:host=$host;dbname=$dbname", $username, $password);
 $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
 echo "Соединение установлено";
} catch(PDOException $e) {
 echo "Ошибка соединения: " . $e->getMessage();
}
?>
``

```

## 2. Извлечение данных из базы данных:

Для извлечения данных из таблицы базы данных используется запрос SQL. Пример извлечения всех записей из таблицы "users":

```

``php
<?php
$stmt = $pdo->query('SELECT * FROM users');
while ($row = $stmt->fetch()) {
 echo "ID: " . $row['id'] . ", Name: " . $row['name'] . "
";
}
?>

```

## 3. Редактирование данных в базе данных:

Для редактирования данных в таблице используется запрос SQL с оператором UPDATE. Пример изменения имени пользователя с ID=1:

```

``php
<?php
$name = 'Новое имя';
$id = 1;
$stmt = $pdo->prepare('UPDATE users SET name = :name WHERE id = :id');
$stmt->execute(array('name' => $name, 'id' => $id));
echo "Данные успешно обновлены";
?>
``

```

## 4. Обновление данных в базе данных:

Для добавления новой записи в таблицу используется запрос SQL с оператором INSERT. Пример добавления нового пользователя:

```

``php
<?php
$name = 'Новый пользователь';
$email = 'new_user@example.com';
$stmt = $pdo->prepare('INSERT INTO users (name, email) VALUES (:name, :email)');
$stmt->execute(array('name' => $name, 'email' => $email));
echo "Новый пользователь добавлен";
?>

```

## 9. Объектно-ориентированное программирование на PHP

Объектно-ориентированное программирование (ООП) в PHP является мощным подходом, который позволяет создавать модульный и структурированный код. Ключевыми понятиями ООП в PHP являются классы и объекты. Класс - это шаблон для создания объектов, определяющий свойства (переменные) и методы (функции) этого объекта. Объект - экземпляр класса, обладающий свойствами и методами, определенными в классе.

Основные принципы ООП на PHP

### 1. Инкапсуляция:

Инкапсуляция позволяет скрыть детали реализации от внешнего мира и предоставить доступ только к необходимым методам и свойствам объекта. В PHP это достигается с помощью модификаторов доступа: `public`, `protected` и `private`.

#### 2. Наследование:

Наследование позволяет создавать новые классы на основе существующих, заимствуя их свойства и методы. Подкласс (наследник) может расширять или переопределять функциональность родительского класса. Наследование в PHP реализуется с помощью ключевого слова `extends`.

#### 3. Полиморфизм:

Полиморфизм позволяет объектам разных классов реагировать на одинаковые вызовы методов по-разному. В PHP полиморфизм достигается через использование интерфейсов и абстрактных классов, а также методов `__call` и `__callStatic`.

#### 4. Абстракция:

Абстракция заключается в определении общего интерфейса для классов, скрывая детали реализации.

#### 10. PHP + XML, стилевые таблицы XSLT

XSLT (Extensible Stylesheet Language Transformations) – это язык преобразования XML-данных в другой формат, часто используемый для создания HTML-страниц из XML-документов.

Для использования XSLT в PHP, сначала нужно создать XSLT-шаблон, описывающий, как XML-данные должны быть преобразованы. Затем можно использовать встроенные функции PHP для применения XSLT-шаблона к XML-данным.

#### 11. Язык сценариев JavaScript. Объекты в JavaScript

Объекты являются основным строительным блоком в JavaScript. Они представляют собой структуру данных, содержащую ключи (свойства) и значения, где значения могут быть какими угодно типами данных, включая другие объекты.

##### Прототипы в JavaScript:

Каждый объект в JavaScript имеет свой прототип, который является другим объектом или `null`.

##### Ключевое слово "this" в JavaScript:

Ключевое слово `**this**` в JavaScript используется для обращения к текущему объекту внутри метода объекта.

#### 12. Основы jQuery и основы выборки элементов

jQuery — это быстрая и компактная библиотека JavaScript, которая облегчает работу с документом HTML, обработку событий, анимацию и работу с AJAX. Одним из ключевых преимуществ jQuery является возможность легко выбирать элементы на веб-странице и манипулировать ими.

Для выборки элементов jQuery использует селекторы, которые аналогичны селекторам CSS

##### 1. Элементы HTML:

- `$( 'p' )` - выбрать все `<p>` элементы на странице.

##### 2. Классы CSS:

- `$( '.my-class' )` - выбрать все элементы с классом `'my-class'`.

##### 3. Идентификаторы:

- `$( '#my-id' )` - выбрать элемент с идентификатором `'my-id'`.

##### 4. Дочерние элементы:

- `$( 'div p' )` - выбрать все `<p>` элементы, которые являются потомками `<div>` элементов.

##### 5. Соседние элементы:

- `\$('h2 + p')` - выбрать `

` элементы, которые следуют непосредственно за `

## ` элементами.

После выборки элементов с помощью селекторов, можно применять к ним различные методы jQuery для изменения их содержимого, стилей или обработки событий.

Например:

```
````javascript
// Скрыть все элементы <p>, которые имеют класс 'my-class'
$('p.my-class').hide();
// Изменить текст элемента с идентификатором 'my-id'
$('#my-id').text('Новый текст');
```

13. Технология AJAX

AJAX (Asynchronous JavaScript and XML) - это набор технологий, который позволяет отправлять и получать данные с сервера без необходимости перезагрузки всей веб-страницы. Основные компоненты AJAX включают в себя:

1. JavaScript: используется для отправки асинхронных запросов на сервер и обработки полученных данных.
2. XMLHttpRequest (XHR): объект, который используется для взаимодействия с сервером и отправки/получения данных в формате JSON, XML или других.
3. DOM (Document Object Model): позволяет динамически обновлять содержимое веб-страницы без перезагрузки.

Преимущества использования технологии AJAX:

- Повышение производительности: за счет передачи только необходимых данных и обновления только определенных частей страницы.
- Улучшенный пользовательский опыт: пользователь может взаимодействовать с веб-приложением без задержек и перезагрузок.
- Экономия трафика: не нужно загружать всю страницу снова, а только необходимые данные.
- Асинхронные запросы: позволяют делать несколько запросов одновременно, не блокируя интерфейс.

Критерии оценки:

Отметка «5»: ответ полный и правильный на основании изученных теорий; материал изложен в определенной логической последовательности, литературным языком. Ответ самостоятельный.

Отметка «4»: ответ полный и правильный на основании изученных теорий; материал изложен в определенной логической последовательности, при этом допущены две-три несущественные ошибки, исправленные по требованию преподавателя.

Отметка «3»: ответ полный, но при этом допущена существенная ошибка, или неполный, несвязный.

Отметка «2»: при ответе обнаружено непонимание обучающимся основного содержания учебного материала или допущены существенные ошибки, которые он не смог исправить при наводящих вопросах преподавателя.

Составители:

_____ Ж.А. Кадышева

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ ДАГЕСТАН

Частное профессиональное образовательное учреждение

«Республиканский полипрофессиональный колледж»

Самостоятельная работа при изучении МДК.01.05 Разработка веб-приложений

Цель работы: изучение и освоение принципов разработки сайта.

Задачи практической работы

1. Изучить общие принципы разработки сайта.
2. Определить основную тему разрабатываемого сайта.
3. Разработать структуру и содержание сайта.
4. Написать сценарий сайта.
5. Разработать макет сайта.
6. Собрать и подготовить информацию и материалы для сайта.
7. Определить аппаратно-программные потребности сайта.

Перечень обеспечивающих средств для выполнения практической работы:

- индивидуальное задание на выполнение работы для 2 человек;
- персональный компьютер;
- программное обеспечение (ПО) Microsoft Office Word;

Задание

Для выполнения практической работы № 1 необходимо изучить общие принципы разработки сайта. Студент должен выбрать из таблицы 1 один из вариантов названия будущего разрабатываемого сайта. Для определения основной темы разрабатываемого сайта необходимо сформулировать, что должно присутствовать на сайте, представляющем компанию (или проект). Разработать структуру, содержание, дизайн, макет сайта. Написать техническое задание (сценарий), собрать, подготовить информацию и материалы для разрабатываемого сайта. Определить аппаратно-программные требования сайта.

Тематика сайтов:

1. «Создание сайта Каталог продукции»
2. «Создание сайта магазин игрушек»
3. «Создание сайта резюме преподавателя»

3.6. Рекомендуемая литература для разработки оценочных средств и подготовки обучающихся к аттестации.

Таблица 1 – Обеспечение дисциплины учебными изданиями

Библиографическое описание издания (автор, заглавие, вид, место и год издания, кол. стр.)	Основная/ дополнительная литература	Книгообеспеченность	
		Кол-во. экз. в библ.	Электронные ресурсы
Подбельский, В. В. Программирование. Базовый курс C#: учебник для среднего профессионального образования / В. В. Подбельский. — Москва : Издательство Юрайт, 2023. — 369 с.	Основная	-	https://urait.ru/bcode/517893
Кудрина, Е. В. Основы алгоритмизации и программирования на языке C#: учебное пособие для среднего профессионального образования / Е. В. Кудрина, М. В. Огнева. — Москва : Издательство Юрайт, 2023. — 322 с.	Основная	-	https://urait.ru/bcode/517324
Черткова, Е. А. Программная инженерия. Визуальное моделирование программных систем: учебник для среднего профессионального образования / Е. А. Черткова. — 2-е изд.,	Основная	-	https://urait.ru/bcode/515393

испр. и доп. — Москва : Издательство Юрайт, 2023. — 147 с.			
Трофимов, В. В. Основы алгоритмизации и программирования : учебник для среднего профессионального образования / В. В. Трофимов, Т. А. Павловская ; под редакцией В. В. Трофимова. — Москва : Издательство Юрайт, 2023. — 137 с	Дополнительная	-	https://urait.ru/bcode/515434
Казанский, А. А. Программирование на Visual C# : учебное пособие для среднего профессионального образования / А. А. Казанский. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2023. — 192 с.	Дополнительная	-	https://urait.ru/bcode/513400

Таблица 2 – Перечень современных профессиональных баз данных (СПБД)

№	Наименование СПБД
1	Научная электронная библиотека eLIBRARY - www.elibrary.ru
2	Научная электронная библиотека КиберЛениНка - www.cyberleninka.ru

Таблица 3 – Перечень информационных справочных систем (ИСС)

№	Наименование ИСС
1	Справочная правовая система КонсультантПлюс www.consultant.ru
2	Электронная библиотечная система ЭБС ЮРАИТ - www.urait.ru

3.7 перечень материалов, оборудования и информационных источников

Оборудование учебного кабинета и рабочих мест кабинета:

Кабинет № 31 информатики (для проведения занятий лекционного типа и занятий семинарского типа, курсового проектирования (выполнения курсовых работ) групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации) оборудован мультимедийным комплексом. Специализированная мебель: Учебная мебель на 39 посадочных места (столов трехместных 13 шт., скамеек 13 шт.), рабочее место преподавателя (стол 1 шт., стул 1 шт.), кафедра 1 шт. доска меловая 3х секционная 1шт. Компьютер Intel Pentium Dual CPU E2160 1,8 GHz ОЗУ- 2 Gb, HDD-500Gb, DVD RW-ROM, Клавиатура, Мышь. ОС windows 7 Максимальная. Локальный сеть с выходом в Интернет. Видеопроектор потолочный Epson EB-S82, проекционный экран Clasic Solition 266x149, акустические колонки Genius.

Наборы демонстрационного оборудования и учебно-наглядных пособий: мультимедийные приложения к лекционным курсам и практическим занятиям, интерактивные учебно-наглядные пособия.

Лаборатория № 2.2 программного обеспечения и сопровождения компьютерных систем. Компьютерный класс (для проведения практических занятий, с применением вычислительной техники) оборудован мультимедийным комплексом. Специализированная мебель и оборудование:

Учебная мебель на 16 посадочных мест, рабочее место преподавателя (стол – 1 шт., стул – 1 шт.). Компьютер Intel i5 7400/1Tb/8Gb/Philips 243V5Q 23' – 16 шт. Компьютер Intel i3 - 2100 2.4 Ghz/4/500Gb/Acer V193 19» – 1 шт. Мультимедийный проектор Тип 1 Optoma x 400 – 1 шт. Консультант + (Договор поставки и сопровождения экземпляров системы № 124 от 28.08.2020), 7-Zip (freeware), Acrobat Reader DC (freeware), Adobe Acrobat Reader

DC (freeware), FireFox 77.0.1 (freeware), Google Chrome 83.0.4103.97 (freeware), VLC media player (freeware), K-Lite Codec Pack Full (freeware). Программное обеспечение общего и профессионального назначения бесплатное (с открытой лицензией): EclipseIDEforJavaEEDevelopers, .NETFrameworkJDK 8, MicrosoftSQLServerExpressEdition, RAD Studio, NetBeans, ARIS Inkscape, MySQLInstallerforWindows, SQLServerManagementStudio, MicrosoftSQLServerJavaConnector, AndroidStudio, IntelliJIDEA.

Наборы демонстрационного оборудования и учебно-наглядных пособий: мультимедийные приложения к лекционным курсам и практическим занятиям, интерактивные учебнонаглядные пособия.

Студия 3.1 инженерной и компьютерной графики. Компьютерный класс (для проведения практических занятий, курсового проектирования (выполнения курсовых работ) с применением вычислительной техники).

Специализированная мебель и оборудование: Учебная мебель на 15 посадочных мест, рабочее место преподавателя (стол - 1 шт., стул - 1 шт.). Компьютер Intel i5 4460/1Тб/8Гб/монитор Samsung 23" - 1 шт. Мультимедийный проектор Тип 1 Optoma x 400 - 1 шт. Компьютер Intel i5 4460/1Тб/8Гб/ монитор Samsung 23" - 15 шт. Графический планшет. Перечень лицензионного программного обеспечения: Консультант + (Договор поставки и сопровождения экземпляров системы № 124 от 28.08.2020), 7-Zip (freeware), Acrobat Reader DC (freeware), Adobe Acrobat Reader DC (freeware), FireFox 77.0.1 (freeware), Google Chrome83.0.4103.97 (freeware), VLC media player (freeware), K-Lite Codec Pack Full (freeware).

Программное обеспечение общего и профессионального назначения бесплатное (с открытой лицензией): EclipseIDEforJavaEEDevelopers, .NETFrameworkJDK 8, MicrosoftSQLServerExpressEdition, RAD Studio, NetBeans, ARIS Inkscape, MySQLInstallerforWindows, SQLServerManagementStudio, MicrosoftSQLServerJavaConnector, AndroidStudio, IntelliJIDEA.

Наборы демонстрационного оборудования и учебно - наглядных пособий: мультимедийные приложения к лекционным курсам и практическим занятиям, интерактивные учебно-наглядные пособия.

Студия 3.2 разработки дизайна веб-приложений. Компьютерный класс (для проведения практических занятий, курсового проектирования (выполнения курсовых работ) с применением вычислительной техники). Специализированная мебель и оборудование: Учебная мебель на 15 посадочных мест (15 компьютерных столов, 15 черных кресел) рабочее место преподавателя (компьютерный стол 1шт., кресло 1шт.), доска меловая односекционная 1шт., доска маркерная на колесиках 1 шт., жалюзи 1шт. Компьютер Intel i5 4460/1Тб/8Гб/монитор Samsung 23" - 1 шт., Мультимедийный проектор Тип 1 Optoma x 400 - 1 шт. Компьютер Intel i5 4460/1Тб/8Гб/ монитор Samsung 23" - 15 шт. Графический планшет. Перечень лицензионного программного обеспечения:, Консультант + (Договор поставки и сопровождения экземпляров системы № 124 от 28.08.2020), 7-Zip (freeware), Acrobat Reader DC (freeware), Adobe Acrobat Reader DC (freeware), FireFox 77.0.1 (freeware), Google Chrome 83.0.4103.97 (freeware), VLC media player (freeware), K-Lite Codec Pack Full (freeware).

Программное обеспечение общего и профессионального назначения бесплатное (с открытой лицензией): EclipseIDEforJavaEEDevelopers, .NETFrameworkJDK 8, MicrosoftSQLServerExpressEdition, RAD Studio, NetBeans, ARIS Inkscape, MySQLInstallerforWindows, SQLServerManagementStudio, MicrosoftSQLServerJavaConnector, AndroidStudio, IntelliJIDEA.

Наборы демонстрационного оборудования и учебно-наглядных пособий: мультимедийные приложения к лекционным курсам и практическим занятиям, интерактивные учебнонаглядные пособия.

Наборы демонстрационного оборудования и учебно-наглядных пособий: мультимедийные приложения к лекционным курсам и практическим занятиям, интерактивные учебно-наглядные пособия.

Реализация профессионального модуля предполагает обязательную учебную практику.

Оборудование и технологическое оснащение рабочих мест:

Студия 3.1 инженерной и компьютерной графики. Компьютерный класс (для проведения практических занятий, курсового проектирования (выполнения курсовых работ) с применением вычислительной техники).

Специализированная мебель и оборудование: Учебная мебель на 15 посадочных мест, рабочее место преподавателя (стол - 1 шт., стул - 1 шт.). Компьютер Intel i5 4460/1Тб/8Гб/монитор Samsung 23" - 1 шт. Мультимедийный проектор Тип 1 Optoma x 400 - 1 шт. Компьютер Intel i5 4460/1Тб/8Гб/ монитор Samsung 23" - 15 шт. Графический планшет. Перечень лицензионного программного обеспечения: Консультант + (Договор поставки и сопровождения экземпляров системы № 124 от 28.08.2020), 7-Zip (freeware), Acrobat Reader DC (freeware), Adobe Acrobat Reader DC (freeware), FireFox 77.0.1 (freeware), Google Chrome 83.0.4103.97 (freeware), VLC media player (freeware), K-Lite Codec Pack Full (freeware).

Программное обеспечение общего и профессионального назначения бесплатное (с открытой лицензией): EclipseIDEforJavaEEDevelopers, .NETFrameworkJDK 8, MicrosoftSQLServerExpressEdition, RAD Studio, NetBeans, ARIS Inkscapе, MySQLInstallerforWindows, SQLServerManagementStudio, MicrosoftSQLServerJavaConnector, AndroidStudio, IntelliJIDEA.

Наборы демонстрационного оборудования и учебно - наглядных пособий: мультимедийные приложения к лекционным курсам и практическим занятиям, интерактивные учебно-наглядные пособия.

Студия 3.2 разработки дизайна веб-приложений. Компьютерный класс (для проведения практических занятий, курсового проектирования (выполнения курсовых работ) с применением вычислительной техники). Специализированная мебель и оборудование: Учебная мебель на 15 посадочных мест (15 компьютерных столов, 15 черных кресел) рабочее место преподавателя (компьютерный стол 1шт., кресло 1шт.), доска меловая односекционная 1шт., доска маркерная на колесиках 1 шт., жалюзи 1шт. Компьютер Intel i5 4460/1Тб/8Гб/монитор Samsung 23" - 1 шт., Мультимедийный проектор Тип 1 Optoma x 400 - 1 шт. Компьютер Intel i5 4460/1Тб/8Гб/ монитор Samsung 23" - 15 шт. Графический планшет. Перечень лицензионного программного обеспечения: Консультант + (Договор поставки и сопровождения экземпляров системы № 124 от 28.08.2020), 7-Zip (freeware), Acrobat Reader DC (freeware), Adobe Acrobat Reader DC (freeware), FireFox 77.0.1 (freeware), Google Chrome 83.0.4103.97 (freeware), VLC media player (freeware), K-Lite Codec Pack Full (freeware).

Программное обеспечение общего и профессионального назначения бесплатное (с открытой лицензией): EclipseIDEforJavaEEDevelopers, .NETFrameworkJDK 8, MicrosoftSQLServerExpressEdition, RAD Studio, NetBeans, ARIS Inkscapе, MySQLInstallerforWindows, SQLServerManagementStudio, MicrosoftSQLServerJavaConnector, AndroidStudio, IntelliJIDEA.

Наборы демонстрационного оборудования и учебно-наглядных пособий: мультимедийные приложения к лекционным курсам и практическим занятиям, интерактивные учебнонаглядные пособия.

Реализация профессионального модуля предполагает обязательную производственную практику.

Кабинет № 31 информатики (для проведения занятий лекционного типа и занятий семинарского типа, курсового проектирования (выполнения курсовых работ) групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации) оборудован мультимедийным комплексом. Специализированная мебель: Учебная мебель на 39 посадочных места (столов трехместных 13 шт., скамеек 13 шт.), рабочее место преподавателя (стол 1 шт., стул 1 шт.), кафедра 1 шт. доска меловая 3х секционная 1шт. Компьютер Intel Pentium Dual CPU E2160 1,8 GHz ОЗУ- 2 Gb, HDD-500Gb, DVD RW-ROM, Клавиатура, Мышь. ОС windows 7 Максимальная. Локальный сеть с выходом в Интернет. Видеопроектор потолочный Epson EB-S82, проекционный экран Clasic Solition 266x149, акустические колонки Genius.

Наборы демонстрационного оборудования и учебно-наглядных пособий: мультимедийные приложения к лекционным курсам и практическим занятиям, интерактивные учебно-наглядные пособия.

Наборы демонстрационного оборудования и учебно-наглядных пособий: мультимедийные приложения к лекционным курсам и практическим занятиям, интерактивные учебно-наглядные пособия.

Долгосрочные договора о проведении практики:

Договор от 10.05.2023 №ПП-02/23 между ГБУ РД «Кизлярская центральная городская больница» и ЧПОУ «Республиканский полипрофессиональный колледж». Срок действия договора до 28.02.2026. Адрес организации : 368830, Республика Дагестан, г.Кизляр, ул.Победы, д.48

Договор от 10.05.2023 №ПП-04/23 между Обществом с ограниченной ответственностью «Оптимасеть» и ЧПОУ «Республиканский полипрофессиональный колледж». Срок действия договора до 30.06.2027. Адрес организации : 368830, Республика Дагестан, г.Кизляр, Аэродромная улица, строение 4

Договор от 10.05.2023 №ПП-05/23 между АО «Кизлярагрокомплекс» и ЧПОУ «Республиканский полипрофессиональный колледж». Срок действия договора до 30.06.2027. Адрес организации : 368830, Республика Дагестан, г.Кизляр, улица Коммунистическая, д.53

Договор от 10.05.2023 №ПП-03/23 между Филиалом СПБГЭУ в городе Кизляре и ЧПОУ «Республиканский полипрофессиональный колледж». Срок действия договора до 30.06.2027. Адрес организации : 368830, Республика Дагестан, г.Кизляр, улица Ленина, 14

Реализация профессионального модуля предполагает обязательную сдачу квалификационного экзамена.

Кабинет № 31 информатики (для проведения занятий лекционного типа и занятий семинарского типа, курсового проектирования (выполнения курсовых работ) групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации) оборудован мультимедийным комплексом. Специализированная мебель: Учебная мебель на 39 посадочных места (столов трехместных 13 шт., скамеек 13 шт.), рабочее место преподавателя (стол 1 шт., стул 1 шт.), кафедра 1 шт. доска меловая 3х секционная 1шт. Компьютер Intel Pentium Dual CPU E2160 1,8 GHz ОЗУ- 2 Gb, HDD-500Gb, DVD RW-ROM, Клавиатура, Мышь. ОС windows 7 Максимальная. Локальный сеть с выходом в Интернет. Видеопроектор потолочный Epson EB-S82, проекционный экран Clasic Solition 266x149, акустические колонки Genius.

Наборы демонстрационного оборудования и учебно-наглядных пособий: мультимедийные приложения к лекционным курсам и практическим занятиям, интерактивные учебно-наглядные пособия.

Лаборатория . № 2.2 программного обеспечения и сопровождения компьютерных систем. Компьютерный класс (для проведения практических занятий, с применением вычислительной техники) оборудован мультимедийным комплексом.

Специализированная мебель и оборудование:

Учебная мебель на 16 посадочных мест, рабочее место преподавателя (стол – 1 шт., стул – 1 шт.). Компьютер Intel i5 7400/1Tb/8Gb/Philips 243V5Q 23' – 16 шт. Компьютер Intel i3 - 2100 2.4 Ghz/4/500Gb/Acer V193 19» – 1 шт. Мультимедийный проектор Тип 1 Optoma x 400 – 1 шт. Консультант + (Договор поставки и сопровождения экземпляров системы № 124 от 28.08.2020), 7-Zip (freeware), Acrobat Reader DC (freeware), Adobe Acrobat Reader DC (freeware), FireFox 77.0.1 (freeware), Google Chrome 83.0.4103.97 (freeware), VLC media player (freeware), K-Lite Codec Pack Full (freeware). Программное обеспечение общего и профессионального назначения бесплатное (с открытой лицензией): EclipseIDEforJavaEEDevelopers, .NETFrameworkJDK 8, MicrosoftSQLServerExpressEdition, RAD Studio, NetBeans, ARIS Inkscape, MySQLInstallerforWindows, SQLServerManagementStudio, MicrosoftSQLServerJavaConnector, AndroidStudio, IntelliJIDEA.

Наборы демонстрационного оборудования и учебно-наглядных пособий: мультимедийные приложения к лекционным курсам и практическим занятиям, интерактивные учебнонаглядные пособия.

5. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ПРОХОЖДЕНИЯ УЧЕБНОЙ ПРАКТИКИ ПО ПМ.01 РАЗРАБОТКА МОДУЛЕЙ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ КОМПЬЮТЕРНЫХ СИСТЕМ

Текущий контроль результатов прохождения учебной практики происходит при использовании следующих обязательных форм контроля: - ежедневный контроль посещаемости практики; - наблюдением за выполнением видов работ на практике, предусмотренных программой практики; - контроль качества выполнения видов работ по практике; - контроль за ведением дневника практики и составлением отчета. Форма промежуточной аттестации по учебной практике по ПМ.01 Разработка модулей программного обеспечения для компьютерных систем – зачет. Практика завершается зачетом при условии: - полноты и своевременности представления дневника практики и отчета о практике в соответствии с заданием на практику.

5. ПЕРЕЧЕНЬ ВИДОВ РАБОТ ДЛЯ ПРОВЕРКИ РЕЗУЛЬТАТОВ ОСВОЕНИЯ ПРОГРАММЫ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ НА ПРАКТИКЕ

Учебная практика УП.01.01

Виды работ	Коды проверяемых результатов		
	ПК	ОК	ПО, У
Раздел 1. Разработка программных модулей	ПК 1.1, ПК 1.2	ОК 01-ОК 09	П1, У1, У2, У4, У6
Раздел 2. Поддержка и тестирование программных модулей	ПК 1.3, ПК 1.4, ПК1.5	ОК 01-ОК 09	П2, П3, П4, У3, У5,У6
Раздел 3. Разработка мобильных	ПК 1.2, ПК 1.6	ОК 01-ОК 09	П1, У2, П5, У4
Раздел 4. Системное программирование	ПК 1.2, ПК 1.3	ОК 01-ОК 09	П1, У2, П2, У3
Раздел 5. Разработка веб-приложений	ДПК 1.7	ОК 01-ОК 09	П1, У4

Производственная практика ПП.01.01

Код и наименование профессиональных модулей и тем производственной практики	Содержание учебных занятий	Объем часов
Виды работ:		
Раздел 1. Разработка программных модулей	1. Разработка и составление спецификаций 2. Разработка и составление тестов	96
Раздел 2. Поддержка и тестирование программных модулей	3. Разработка и проектирование алгоритмов 4. Разработка модулей программы 5. Отладка и тестирование модулей	30
Раздел 3. Разработка мобильных	6. Сборка модулей 7. Оформление документации	30
Раздел 4. Системное программирование		30
Раздел 5. Разработка веб-приложений		30
	Всего	216

6. КОНТРОЛЬНО-ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ЭКЗАМЕНА (КВАЛИФИКАЦИОННОГО)

Экзамен (квалификационный) представляет собой практическую работу и предназначен для контроля и оценки результатов освоения профессионального модуля ПМ.01 Разработка программных модулей программного обеспечения для компьютерных систем по специальности СПО 09.02.03 Программирование в компьютерных системах. Итогом экзамена является однозначное решение: «вид профессиональной деятельности освоен / не освоен». Для вынесения положительного заключения об освоении ВПД, необходимо подтверждение сформированности всех компетенций, перечисленных в программе ПМ. При отрицательном заключении хотя бы одной из профессиональных компетенции принимается решение «Вид профессиональной деятельности не освоен». В состав комплекта входят задания для экзаменуемых, пакет для экзаменатора (эксперта) и оценочная ведомость.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ ДАГЕСТАН
 Частное профессиональное образовательное учреждение
 «Республиканский полипрофессиональный колледж»

ЗАДАНИЕ ДЛЯ ЭКЗАМЕНА (КВАЛИФИКАЦИОННОГО)

Место проведения: Лаборатория 3.1

Оборудование: лаборатория обработки информации отраслевой направленности, посадочные места с ПК по количеству обучающихся; рабочее место преподавателя.

Экзамен квалификационный включает:

1. Тестирование по теоретической части профессионального модуля.
2. Предоставление портфолио;

ПАКЕТ ЭКЗАМЕНАТОРА

Инструкция

1. Ознакомьтесь с заданиями для экзаменуемых

2. Проверьте готовность рабочих мест экзаменуемых
 3. Проверить наличие экзаменационного материала
- Количество вариантов заданий, экзаменуемых: 20

Требования к портфолио

Тип портфолио: смешанный

Цель портфолио: выявить уровень сформированности общих и профессиональных компетенций в процессе освоения всех элементов профессионального модуля. Портфолио оформляется студентом в течение всего периода освоения программы профессионального модуля (в том числе в период учебной и производственной практик) под руководством преподавателей, руководителей учебной и производственной практик.

Структура портфолио

Титульный лист

Диск в файле, который содержит: презентацию, портфолио, папки с программами.

- индивидуальные показатели успеваемости;
- ведомость выполнения практических работ по МДК 01.01 Разработка программных модулей
- ведомость выполнения практических работ по МДК 01.02 Поддержка и тестирование программных модулей
- ведомость выполнения практических работ по МДК 01.03 Разработка мобильных приложений
- ведомость выполнения практических работ по МДК 01.04 Системное программирование
- ведомость выполнения практических работ по МДК 01.05 Разработка веб-приложений
- аттестационный лист по учебной практике;
- аттестационный лист по производственной практике.

Основные требования

Требования к структуре и оформлению портфолио:

Портфолио оформляется на электронном и бумажном носителях.

- объём презентаций не должен превышать 20 слайдов. Презентации сохранять в формате Power Point 2010 (.pptx);
 - размер фото не более 100 кб, расширение не менее 800 x 600.
 - текстовые документы представляются в формате Word 2003 (doc.);
 - параметры текстового редактора: поля – верхнее, нижнее – 2.0 см, левое – 3.0 см, правое – 1 см, шрифт Times New Roman, высота 14, межстрочный интервал полуторный, выравнивание по ширине, красная строка 1,25 см;
 - документы, содержащие подписи и печати, сканируются в формате JPG или PDF, отсканированный текст, подписи и печати должны читаться без затруднений в масштабе 1:1;
 - в текстах не допускается сокращение названий и наименований;
 - все страницы нумеруются (нумерация начинается с титульного листа, номер на первой странице не ставится), номер страницы ставится в центре нижней части страницы, в колонтитуле на каждой странице указываются фамилия, имя и отчество студента.
- Портфолио на бумажном носителе формируется в одной папке-накопителе с файлами. Портфолио должно быть представлено за день до квалификационного экзамена

Задания для экзаменуемого

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ ДАГЕСТАН

Частное профессиональное образовательное учреждение

«Республиканский полипрофессиональный колледж»

КВАЛИФИКАЦИОННЫЙ ЭКЗАМЕН

(ПМ.01 Разработка программных модулей программного обеспечения для компьютерных систем)

1. Спецификация разрабатываемого программного обеспечения при использовании языка визуального моделирования UML объединяет несколько моделей. Содержит описание функций программного обеспечения с точки зрения пользователя

Впишите ответ:

[.....]

2. Спецификация разрабатываемого программного обеспечения при использовании языка визуального моделирования UML объединяет несколько моделей. Описывает ключевые понятия моделируемого программного обеспечения (классы, интерфейсы и т. п.), т. е. средства, обеспечивающие его функциональность ...

- 1) логическая модель
- 2) модель реализации
- 3) модель процессов
- 4) модель развертывания

3. Спецификация разрабатываемого программного обеспечения при использовании языка визуального моделирования UML объединяет несколько моделей. Определяет реальную организацию программных модулей в среде разработки

Впишите ответ:

[.....]

4. Спецификация разрабатываемого программного обеспечения при использовании языка визуального моделирования UML объединяет несколько моделей. Отображает организацию вычислений и позволяет оценить производительность, масштабируемость и надежность программного обеспечения

- 1) модель использования
- 2) модель реализации
- 3) модель процессов
- 4) модель развертывания

5. Служит связующим звеном между процессами, т. е. позволяет описать порядок взаимодействия процессов ...

- 1) нотация
- 2) прецедент
- 3) поток событий
- 4) сценарий

6. Подробное процедурное описание вариантов использования системы всеми заинтересованными лицами, а также внешними системами, реализует ...

- 1) нотация
- 2) прецедент
- 3) поток событий
- 4) сценарий

7. Видимость переменных и методов класса только в пределах данного модуля определяет спецификатора доступа ...

- 1) private

- 2) protected
- 3) public
- 4) published

8. Видимость переменных и методов класса в классах потомков определяет спецификатора доступа ...

- 1) private
- 2) protected
- 3) public
- 4) published

9. Видимость переменных и методов класса в любом месте программы определяет спецификатора доступа ...

- 1) private
- 2) protected
- 3) public
- 4) published

10. Процесс, в ходе которого один объект может приобретать свойства другого, называется...

Впишите ответ:

[.....]

11. Модульное тестирование предполагает ...

- 1) тестирование минимально возможного для тестирования компонента, например отдельного класса или функции
- 2) поиск каких-либо проблем в интерфейсах и взаимодействиях между компонентами программы
- 3) тестирование интегрированной системы на ее соответствие исходным требованиям
- 4) имитацию реальной работы с системой штатными разработчиками либо реальной работы с системой потенциальными пользователями/заказчиком на стороне разработчика

12. Интеграционное тестирование предполагает ...

- 1) тестирование минимально возможного для тестирования компонента, например отдельного класса или функции
- 2) поиск каких-либо проблем в интерфейсах и взаимодействиях между компонентами программы с. тестирование интегрированной системы на ее соответствие исходным требованиям
- 3) распространение версии с ограничениями (по функциональности или времени работы) для некоторой группы лиц с тем, чтобы убедиться, что продукт содержит достаточно мало ошибок

13. Системное тестирование предполагает ...

- 1) поиск каких-либо проблем в интерфейсах и взаимодействиях между компонентами программы в. тестирование интегрированной системы на ее соответствие исходным требованиям
- 2) имитацию реальной работы с системой штатными разработчиками либо реальной работы с системой потенциальными пользователями/заказчиком на стороне разработчика
- 3) распространение версии с ограничениями (по функциональности или времени работы) для некоторой группы лиц с тем, чтобы убедиться, что продукт содержит достаточно мало ошибок

14. Каждое направление перехода в программе должно быть реализовано, по крайней мере, один раз в соответствии с методом тестирования

Впишите ответ:

[.....]

15. На основе каких принципов строятся тесты для модульного тестирования?

- 1) анализ потоков управления модуля
- 2) анализ потоков данных модуля
- 3) анализ покрытия в соответствии с заданными критериями C0, C1, C2

16. Каковы фазы процесса построения тестовых путей?

- 1) построение управляющего графа программы
- 2) выбор тестовых путей
- 3) генерация тестов, соответствующих выбранным тестовым путям

17. Какие существуют методы построения тестовых путей?

- 1) статические
- 2) динамические
- 3) методы реализуемых путей

18. Какие существуют разновидности интеграционного тестирования?

Впишите ответ:

[.....]

19. Каковы особенности нисходящего тестирования?

- 1) необходимость разработки заглушек
- 2) необходимость разработки среды управления очередностью вызовов модулей
- 3) параллельная разработка эффективных модулей

20. Каковы особенности восходящего тестирования?

- 1) минимизация разработки заглушек
- 2) запаздывание в проверке функциональности реализуемого приложения
- 3) необходимость разработки среды управления очередностью вызовов модулей

21. Какие методы отражают жизненный цикл Activity:

- 1) onCreate()
- 2) onRestart()
- 3) onResume()
- 4) onRunning()
- 5) onClose()

22. Для создания всплывающего уведомления необходимо инициализировать объект:

- 1) Toast
- 2) Message
- 3) TextView
- 4) MessageBox

23. Какой класс можно использовать для перехода между Activity?

- 1) ActivityChanger
- 2) Activity Intent

3) Switcher

24. От какого класса наследуются все элементы управления?

Впишите ответ:

[.....]

25. Как программно получить доступ к элементам управления через идентификатор?

- 1) getViewById(id)
- 2) findViewById(id)
- 3) findViewById(id)
- 4) getView(id)

26. На основе какой операционной системы была разработана система Android

Впишите ответ:

[.....]

27. Как называется событие-клик объекта-кнопки:

- 1) OnClick
- 2) OnClicked
- 3) Click
- 4) ClickEvent

28. Какой элемент предоставляет наиболее простой вариант разметки. Он может быть использован для абсолютного позиционирования элементов с использованием координат

Впишите ответ:

[.....]

29. Как называется язык разметки Window Phone приложений:

- 1) XAML
- 2) XML
- 3) Java d. C#

30. Система программирования – это:

- 1) комплекс любимых программ программиста
- 2) комплекс программ, облегчающий работу программиста
- 3) комплекс программ, обучающих начальным шагам программиста

31. Системное программное обеспечение:

- 1) программы для организации совместной работы устройств компьютера как единой системы
- 2) программы для организации удобной системы размещения программ на диске
- 3) набор программ для работы устройства системного блока компьютера

32. Системные оболочки – это:

- 1) специальная кассета для удобного размещения дисков с операционной системой
- 2) специальная программа, упрощающая диалог пользователь – компьютер, выполняет команды операционной системы +
- 3) система приемов и способов работы конкретной программы при загрузке программ и завершении работы

33. Программа «драйвер» служит для
- 1) запуска программы на выполнение
 - 2) имитации запуска программы на выполнение
 - 3) проверки правильности работы программы
 - 4) передачи параметров в процедуры и функции
34. Порядковый номер элемента в массиве, который всегда начинается с нуля

Впишите ответ:

[.....]

35. Программирование сверху вниз – это
- 1) Процесс, при котором от начального предположения осуществляется движение по направлению к лучшим решениям
 - 2) Процесс пошагового разбиения алгоритма на все более мелкие части с целью получения таких элементов, для которых можно написать конкретные команды
 - 3) Метод сведения трудной задачи к последовательности более простых
 - 4) Исследование древовидной модели пространства решений и ориентация на поиск оптимального решения

36. Интегрированная система программирования включает компонент для перевода исходного текста программы в машинный код, который называется

Впишите ответ:

[.....]

37. Результатом компиляции программы на языке высокого уровня является
- 1) Командный файл
 - 2) Объектный файл
 - 3) Исходный текст программы на языке высокого уровня
 - 4) Дисплейный файл

38. Псевдокоманда DB означает

- 1) определить байт;
- 2) определить слово (2 байта);
- 3) определить двойное слово (4 байта);
- 4) определить 6 байт
- 5) определить учетверенное слово (8 байт);

39. HTML – это

- 1) Язык разметки
- 2) Библиотека гипертекста
- 3) Скриптовый язык
- 4) Объектно-ориентированный язык

40. Для задания размеров тэгу <frameset> требуются следующие атрибуты

Впишите ответ:

[.....]

41. Какими разделителями окружены скрипты php скрипта?

- 1) <? php ... ?>
- 2) <php> ... </php>

3) `</> php </>`

42. Текст, выделенный курсивом, представлен в следующей записи:

1) ` курсив `

2) `<i> курсив </i>`

3) `<hr> курсив </hr>`

43. В HTML не существует ... тэгов.

1) Одиночных

2) Парных

3) Тройных

4) Закрытых

44. Напишите класс "Канцтовары" с приватными полями для разработчика и компании. Добавьте публичные свойства для доступа и модификации этих полей.

45. Создайте класс "Ручка", который наследуется от класса "Канцтовары". Добавьте свойство для класса «ручка».

46. Создайте абстрактный базовый класс "Канцтовары" с методом "Показать()". Реализуйте этот метод в классах "Ручка" (который также наследуется от "Транспортное средство") по-разному.

47. Создайте интерфейс "IRучка", содержащий метод "Показать(int литры)". Реализуйте этот интерфейс в классах "Кибертехника" и "3Д ручка".

48. Создайте класс "Точка" с приватными полями x и y. Перегрузите операторы сложения, вычитания и умножения для этого класса.

49. Создайте класс "Банковский счет" с приватными полями для номера счета и баланса. Добавьте публичные методы для внесения и снятия денег, а также свойство для просмотра текущего баланса. Обеспечьте правильную инкапсуляцию данных.

50. Создайте перечисление "ДеньНедели" с элементами, соответствующими дням недели. Напишите метод, который принимает день недели в виде перечисления и возвращает, является ли это выходной день или нет.