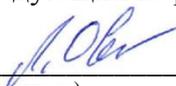


Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Усынин Максим Валерьевич  
Должность: Ректор  
Дата подписания: 07.06.2022 19:14:46  
Уникальный программный ключ:  
f498e59e83f65dd7c3ce7bb8a25cbbabb33ebc58

**Частное образовательное учреждение высшего образования  
«Международный Институт Дизайна и Сервиса»  
(ЧОУВО МИДиС)**

Кафедра математики и информатики

УТВЕРЖДЕН  
на заседании кафедры  
«30» мая 2022 г., протокол № 10  
Заведующий кафедрой

  
Л.Ю. Овсяницкая  
(подпись)

**ФОНД  
ОЦЕНОЧНЫХ СРЕДСТВ  
ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕГО КОНТРОЛЯ И ПРОМЕЖУТОЧНОЙ  
АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ  
ПО ПРОФЕССИОНАЛЬНОМУ МОДУЛЮ**

**ПМ.05 ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ИНФОРМАЦИОННЫХ  
СИСТЕМ**

Специальность:  
**09.02.07 Информационные системы и программирование**

Уровень образования обучающихся:  
**Основное общее образование**

Вид подготовки:  
**Базовый**

Челябинск 2022

**СОДЕРЖАНИЕ**

1. Паспорт фонда оценочных средств .....	3
1.1. Область применения .....	3
1.2. Планируемые результаты освоения компетенций .....	6
1.3. Показатели оценки результатов обучения по профессиональному модулю.....	9
2. Задания для контроля и оценки результатов освоения практического опыта, умений и усвоения знаний .....	10
3. Критерии оценивания. ....	18

## 1. Паспорт фонда оценочных средств

### 1.1. Область применения

Фонд оценочных средств для проведения текущего контроля и промежуточной аттестации обучающихся (далее – Фонд оценочных средств) предназначен для проверки результатов освоения профессионального модуля ПМ.05 Проектирование и разработка информационных систем основной профессиональной образовательной программы среднего профессионального образования - программы подготовки специалистов среднего звена (далее - ППССЗ) по специальности 09.02.07 Информационные системы и программирование.

Профессиональный модуль ПМ.05 Проектирование и разработка информационных систем изучается в течение четырех семестров и включает в себя: МДК.05.01 Проектирование и дизайн информационных систем, МДК 05.02. Разработка кода информационных систем, МДК 05.03. Тестирование кода информационных систем, учебную практику, производственную практику.

Форма аттестации по семестрам.

Наименование	Семестр	Форма аттестации
МДК.05.01 Проектирование и дизайн информационных систем	6 семестр	Экзамен
МДК 05.02. Разработка кода информационных систем	7 семестр	Курсовая работа
МДК 05.02. Разработка кода информационных систем	8 семестр	Дифференцированный зачет
МДК 05.03. Тестирование кода информационных систем	6, 8 семестр	Дифференцированный зачет
Учебная практика	6 семестр	Отчет по практике
Производственная практика	8 семестр	Отчет по практике

Фонд оценочных средств позволяет оценить достижение обучающимися **общих (ОК) и профессиональных (ПК) компетенций:**

*Общие компетенции (ОК):*

Код	Наименование общих компетенций
ОК 01	Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам
ОК 02	Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности.
ОК 03	Планировать и реализовывать собственное профессиональное и личностное развитие.
ОК 04	Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами.
ОК 05	Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста.
ОК 06	Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей
ОК 07	Содействовать сохранению окружающей среды, ресурсосбережению, 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	Использовать информационные технологии в профессиональной деятельности.
ОК 10	Пользоваться профессиональной документацией на государственном и иностранном языках

*Профессиональные компетенции (ПК):*

Код	Наименование видов деятельности и профессиональных компетенций
ВД 5	Проектирование и разработка информационных систем
ПК 5.1.	Собирать исходные данные для разработки проектной документации на информационную систему.
ПК 5.2.	Разрабатывать проектную документацию на разработку информационной системы в соответствии с требованиями заказчика
ПК 5.3	Разрабатывать подсистемы безопасности информационной системы в соответствии с техническим заданием
ПК 5.4	Производить разработку модулей информационной системы в соответствии с техническим заданием
ПК 5.5	Осуществлять тестирование информационной системы на этапе опытной эксплуатации с фиксацией выявленных ошибок кодирования в разрабатываемых модулях информационной системы
ПК 5.6	Разрабатывать техническую документацию на эксплуатацию информационной системы
ПК 5.7	Производить оценку информационной системы для выявления возможности ее модернизации.

В результате изучения профессионального модуля ПМ.05 Проектирование и разработка информационных систем обучающиеся должны:

*иметь практический опыт:*

- в управлении процессом разработки приложений с использованием инструментальных средств;
- обеспечении сбора данных для анализа использования и функционирования информационной системы;
- программировании в соответствии с требованиями технического задания;
- использовании критериев оценки качества и надежности функционирования информационной системы;
- применении методики тестирования разрабатываемых приложений;
- определении состава оборудования и программных средств разработки информационной системы;
- разработке документации по эксплуатации информационной системы;
- проведении оценки качества и экономической эффективности информационной системы в рамках своей компетенции;
- модификации отдельных модулей информационной системы.

*уметь:*

- осуществлять постановку задач по обработке информации;
- проводить анализ предметной области; осуществлять выбор модели и средства построения информационной системы и программных средств;
- использовать алгоритмы обработки информации для различных приложений;
- решать прикладные вопросы программирования и языка сценариев для создания программ;

- разрабатывать графический интерфейс приложения;
  - создавать и управлять проектом по разработке приложения;
  - проектировать и разрабатывать систему по заданным требованиям и спецификациям
- знать:*
- основные виды и процедуры обработки информации, модели и методы решения задач обработки информации;
  - основные платформы для создания, исполнения и управления информационной системой;
  - основные процессы управления проектом разработки;
  - основные модели построения информационных систем, их структуру, особенности и области применения;
  - методы и средства проектирования, разработки и тестирования информационных систем;
  - систему стандартизации, сертификации и систему обеспечения качества продукции.

## 1.2. Планируемые результаты освоения компетенций

В результате освоения программы профессионального модуля ПМ.05 Проектирование и разработка информационных систем учитываются планируемые результаты освоения общих (ОК) и профессиональных (ПК) компетенций:

Код компетенций	Содержание компетенции	Планируемые результаты освоения компетенций
ОК 01	Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам	<p><b>Умения:</b> распознавать задачу и/или проблему в профессиональном и/или социальном контексте; анализировать задачу и/или проблему и выделять её составные части; определять этапы решения задачи; выявлять и эффективно искать информацию, необходимую для решения задачи и/или проблемы; составить план действия; определить необходимые ресурсы;</p> <p>владеть актуальными методами работы в профессиональной и смежных сферах; реализовать составленный план; оценивать результат и последствия своих действий (самостоятельно или с помощью наставника)</p> <p><b>Знания:</b> актуальный профессиональный и социальный контекст, в котором приходится работать и жить; основные источники информации и ресурсы для решения задач и проблем в профессиональном и/или социальном контексте;</p> <p>алгоритмы выполнения работ в профессиональной и смежных областях; методы работы в профессиональной и смежных сферах; структуру плана для решения задач; порядок оценки результатов решения задач профессиональной деятельности</p>

ОК 02	Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности	<p><b>Умения:</b> определять задачи для поиска информации; определять необходимые источники информации; планировать процесс поиска; структурировать получаемую информацию; выделять наиболее значимое в перечне информации; оценивать практическую значимость результатов поиска; оформлять результаты поиска</p> <p><b>Знания:</b> номенклатура информационных источников, применяемых в профессиональной деятельности; приемы структурирования информации; формат оформления результатов поиска информации</p>
ОК 03	Планировать и реализовывать собственное профессиональное и личностное развитие.	<p><b>Умения:</b> определять актуальность нормативно-правовой документации в профессиональной деятельности; применять современную научную профессиональную терминологию; определять и выстраивать траектории профессионального развития и самообразования</p> <p><b>Знания:</b> содержание актуальной нормативно-правовой документации; современная научная и профессиональная терминология; возможные траектории профессионального развития и самообразования</p>
ОК 04	Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами.	<p><b>Умения:</b> организовывать работу коллектива и команды; взаимодействовать с коллегами, руководством, клиентами в ходе профессиональной деятельности</p> <p><b>Знания:</b> психологические основы деятельности коллектива, психологические особенности личности; основы проектной деятельности</p>
ОК 05	Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста.	<p><b>Умения:</b> грамотно излагать свои мысли и оформлять документы по профессиональной тематике на государственном языке, проявлять толерантность в рабочем коллективе</p> <p><b>Знания:</b> особенности социального и культурного контекста; правила оформления документов и построения устных сообщений.</p>
ОК 06	Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей.	<p><b>Умения:</b> описывать значимость своей специальности</p> <p><b>Знания:</b> сущность гражданско-патриотической позиции, общечеловеческих ценностей; значимость профессиональной деятельности по специальности</p>

ОК 07	Содействовать сохранению окружающей среды, ресурсосбережению, эффективно действовать в чрезвычайных ситуациях.	<p><b>Умения:</b> соблюдать нормы экологической безопасности; определять направления ресурсосбережения в рамках профессиональной деятельности по специальности</p> <p><b>Знания:</b> правила экологической безопасности при ведении профессиональной деятельности; основные ресурсы, задействованные в профессиональной деятельности; пути обеспечения ресурсосбережения</p>
ОК 08	Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности.	<p><b>Умения:</b> использовать физкультурно-оздоровительную деятельность для укрепления здоровья, достижения жизненных и профессиональных целей; применять рациональные приемы двигательных функций в профессиональной деятельности; пользоваться средствами профилактики перенапряжения характерными для данной специальности</p> <p><b>Знания:</b> роль физической культуры в общекультурном, профессиональном и социальном развитии человека; основы здорового образа жизни; условия профессиональной деятельности и зоны риска физического здоровья для специальности; средства профилактики перенапряжения</p>
ОК 09	Использовать информационные технологии в профессиональной деятельности	<p><b>Умения:</b> применять средства информационных технологий для решения профессиональных задач; использовать современное программное обеспечение</p> <p><b>Знания:</b> современные средства и устройства информатизации; порядок их применения и программное обеспечение в профессиональной деятельности</p>
ОК 10	Пользоваться профессиональной документацией на государственном и иностранном языках.	<p><b>Умения:</b> понимать общий смысл четко произнесенных высказываний на известные темы (профессиональные и бытовые), понимать тексты на базовые профессиональные темы; участвовать в диалогах на знакомые общие и профессиональные темы; строить простые высказывания о себе и о своей профессиональной деятельности; кратко обосновывать и объяснить свои действия (текущие и планируемые); писать простые связные сообщения на знакомые или интересующие профессиональные темы</p> <p><b>Знания:</b> правила построения простых и сложных предложений на профессиональные темы; основные общеупотребительные глаголы (бытовая и профессиональная лексика); лексический минимум, относящийся к описанию предметов, средств и процессов профессиональной деятельности; особенности произношения; правила чтения текстов профессиональной направленности</p>
ПК 5.1.	Собирать исходные данные для разработки	<p><b>Практический опыт:</b> Анализировать предметную область. Использовать инструментальные средства обработки</p>

	<p>проектной документации на информационную систему.</p>	<p>информации.          Обеспечивать сбор данных для анализа использования и функционирования информационной системы.          Определять состав оборудования и программных средств разработки информационной системы.          Выполнять работы предпроектной стадии.</p> <p><b>Умения:</b>          Осуществлять постановку задачи по обработке информации.          Выполнять анализ предметной области.          Использовать алгоритмы обработки информации для различных приложений.          Работать с инструментальными средствами обработки информации.          Осуществлять выбор модели построения информационной системы.          Осуществлять выбор модели и средства построения информационной системы и программных средств.</p> <p><b>Знания:</b>          Основные виды и процедуры обработки информации, модели и методы решения задач обработки информации.          Основные платформы для создания, исполнения и управления информационной системой.          Основные модели построения информационных систем, их структуру, особенности и области применения.          Платформы для создания, исполнения и управления информационной системой.          Основные процессы управления проектом разработки.          Методы и средства проектирования, разработки и тестирования информационных систем.</p>
ПК 5.2.	<p>Разрабатывать проектную документацию на разработку информационной системы в соответствии с требованиями заказчика.</p>	<p><b>Практический опыт:</b>          Разрабатывать проектную документацию на информационную систему.</p> <p><b>Умения:</b>          Осуществлять математическую и информационную постановку задач по обработке информации.          Использовать алгоритмы обработки информации для различных приложений.</p> <p><b>Знания:</b>          Основные платформы для создания, исполнения и управления информационной системой.          Национальную и международную систему стандартизации и сертификации и систему обеспечения качества продукции, методы контроля качества.          Сервисно - ориентированные архитектуры.          Важность рассмотрения всех возможных вариантов и получения наилучшего решения на основе анализа и интересов клиента.          Методы и средства проектирования информационных систем.          Основные понятия системного анализа.</p>

ПК 5.3.	Разрабатывать подсистемы безопасности информационной системы в соответствии с техническим заданием.	<p><b>Практический опыт:</b> Управлять процессом разработки приложений с использованием инструментальных средств. Модифицировать отдельные модули информационной системы. Программировать в соответствии с требованиями технического задания.</p> <p><b>Умения:</b> Создавать и управлять проектом по разработке приложения и формулировать его задачи. Использовать языки структурного, объектно-ориентированного программирования и языка сценариев для создания независимых программ. Разрабатывать графический интерфейс приложения.</p> <p><b>Знания:</b> Национальной и международной системы стандартизации и сертификации и систему обеспечения качества продукции. Методы контроля качества объектно-ориентированного программирования. Объектно-ориентированное программирование. Спецификации языка программирования, принципы создания графического пользовательского интерфейса (GUI), файлового ввода-вывода, создания сетевого сервера и сетевого клиента. Файлового ввода-вывода. Создания сетевого сервера и сетевого клиента.</p>
ПК 5.4.	Производить разработку модулей информационной системы в соответствии с техническим заданием.	<p><b>Практический опыт:</b> Разрабатывать документацию по эксплуатации информационной системы. Проводить оценку качества и экономической эффективности информационной системы в рамках своей компетенции. Модифицировать отдельные модули информационной системы.</p> <p><b>Умения:</b> Использовать языки структурного, объектно-ориентированного программирования и языка сценариев для создания независимых программ. Решать прикладные вопросы программирования и языка сценариев для создания программ. Проектировать и разрабатывать систему по заданным требованиям и спецификациям. Разрабатывать графический интерфейс приложения. Создавать проект по разработке приложения и формулировать его задачи.</p> <p><b>Знания:</b> Национальной и международной систему стандартизации и сертификации и систему обеспечения качества продукции, методы контроля качества. Объектно-ориентированное программирование. Спецификации языка программирования, принципы</p>

		<p>создания графического пользовательского интерфейса (GUI).</p> <p>Важность рассмотрения всех возможных вариантов и получения наилучшего решения на основе анализа и интересов клиента.</p> <p>Файлового ввода-вывода, создания сетевого сервера и сетевого клиента.</p> <p>Платформы для создания, исполнения и управления информационной системой.</p>
ПК 5.5.	Осуществлять тестирование информационной системы на этапе опытной эксплуатации с фиксацией выявленных ошибок кодирования в разрабатываемых модулях информационной системы.	<p><b>Практический опыт:</b></p> <p>Применять методики тестирования разрабатываемых приложений.</p>
		<p><b>Умения:</b></p> <p>Использовать методы тестирования в соответствии с техническим заданием.</p>
		<p><b>Знания:</b></p> <p>Особенности программных средств, используемых в разработке ИС.</p>
ПК 5.6.	Разрабатывать техническую документацию на эксплуатацию информационной системы.	<p><b>Практический опыт:</b></p> <p>Разрабатывать проектную документацию на информационную систему.</p> <p>Формировать отчетную документацию по результатам работ.</p> <p>Использовать стандарты при оформлении программной документации.</p>
		<p><b>Умения:</b></p> <p>Разрабатывать проектную документацию на эксплуатацию информационной системы.</p> <p>Использовать стандарты при оформлении программной документации.</p>
		<p><b>Знания:</b></p> <p>Основные модели построения информационных систем, их структура.</p> <p>Использовать критерии оценки качества и надежности функционирования информационной системы.</p> <p>Реинжиниринг бизнес-процессов.</p>
ПК 5.7.	Производить оценку информационной системы для выявления возможности ее модернизации.	<p><b>Практический опыт:</b></p> <p>Проводить оценку качества и экономической эффективности информационной системы в рамках своей компетенции.</p> <p>Использовать критерии оценки качества и надежности функционирования информационной системы.</p>
		<p><b>Умения:</b></p> <p>Использовать методы и критерии оценивания предметной области и методы определения стратегии развития бизнес-процессов организации.</p> <p>Решать прикладные вопросы интеллектуальных систем с</p>

		использованием статических экспертных систем, экспертных систем реального времени.
		<b>Знания:</b> Системы обеспечения качества продукции. Методы контроля качества в соответствии со стандартами.

### 1.3. Показатели оценки результатов обучения по профессиональному модулю ПМ.05 Проектирование и разработка информационных систем

Содержание профессионального модуля	Результаты обучения (ОК, ПК)	Вид контроля	Наименование оценочного средства/форма контроля
<b>МДК. 05.01 Проектирование и дизайн информационных систем</b>			
<b>5 семестр</b>			
Тема 5.1.1. Основы проектирования информационных систем	ОК 1. - 10. ПК 5.1. ПК 5.2. ПК 5.6. ПК 5.7.	Текущий	Проверка выполнения индивидуального задания. Устный опрос.
Тема 5.1.2. Система обеспечения качества информационных систем	ОК 1. - 10. ПК 5.1. ПК 5.2. ПК 5.6. ПК 5.7.	Текущий	Проверка выполнения индивидуального задания. Устный опрос.
<b>6 семестр</b>			
Тема 5.1.2. Система обеспечения качества информационных систем	ОК 1. - 10. ПК 5.1. ПК 5.2. ПК 5.6. ПК 5.7.	Текущий	Проверка выполнения индивидуального задания. Устный опрос.
Тема 5.1.3. Разработка документации информационных систем	ОК 1. - 10. ПК 5.1. ПК 5.2. ПК 5.6. ПК 5.7.	Текущий	Проверка выполнения индивидуального задания. Устный опрос.
Темы 5.1.1-5.1.3	ОК 1. - 10. ПК 5.1. ПК 5.2. ПК 5.6. ПК 5.7.	Промежуточный	Экзамен
<b>МДК. 05.02. Разработка кода информационных систем</b>			
<b>5 семестр</b>			
Тема 5.2.1. Основные инструменты для создания, исполнения и управления информационной системой	ОК 1. - 10. ПК 5.1. - 5.4.	Текущий	Проверка выполнения индивидуального задания. Устный опрос.
<b>6 семестр</b>			

<b>Тема 5.2.2. Основные приемы разработки прикладных решений информационной системы</b>	ОК 1. - 10. ПК 5.1. - 5.4.	Текущий	Проверка выполнения индивидуального задания. Устный опрос.
<b>7 семестр</b>			
<b>Тема 5.2.3. Разработка и модификация информационных систем</b>	ОК 1. - 10. ПК 5.1. - 5.4.	Текущий	Проверка выполнения индивидуального задания. Устный опрос.
<b>Тема 5.2.4. Разработка прикладных решений информационной системы</b>	ОК 1. - 10. ПК 5.1. - 5.4.	Текущий	Проверка выполнения индивидуального задания. Устный опрос.
<b>Курсовой проект (работа)</b>	ОК 1. - 10. ПК 5.1. - 5.4.	Промежуточный	Проверка курсового проекта
<b>8 семестр</b>			
<b>Тема 5.2.5. Разработка прикладных решений информационной системы</b>	ОК 1. - 10. ПК 5.1. - 5.4.	Текущий	Проверка выполнения индивидуального задания. Устный опрос.
<b>Тема 5.2.1 – 5.2.5</b>	ОК 1. - 10. ПК 5.1. - 5.4.	Промежуточный	Дифференцированный зачет
<b>МДК.05.03 Тестирование кода информационных систем</b>			
<b>6 семестр</b>			
<b>Тема 5.3.1 Отладка и тестирование программного обеспечения</b>	ОК 1. - 10. ПК 5.2. ПК 5.5. ПК 5.6..	Текущий	Проверка выполнения индивидуального задания. Устный опрос.
<b>Тема 5.3.1</b>	ОК 1. - 10. ПК 5.2. ПК 5.5. ПК 5.6..	Промежуточный	Дифференцированный зачет
<b>7 семестр</b>			
<b>Тема 5.3.1 Отладка и тестирование программного обеспечения</b>	ОК 1. - 10. ПК 5.2. ПК 5.5. ПК 5.6..	Текущий	Проверка выполнения индивидуального задания. Устный опрос.
<b>8 семестр</b>			
<b>Тема 5.3.1 Отладка и тестирование программного обеспечения</b>	ОК 1. - 10. ПК 5.2. ПК 5.5. ПК 5.6..	Текущий	Проверка выполнения индивидуального задания. Устный опрос.
<b>Тема 5.3.1</b>	ОК 1. - 10. ПК 5.2. ПК 5.5. ПК 5.6..	Промежуточный	Дифференцированный зачет

## **Система контроля и оценки результатов освоения практического опыта, умений и усвоения знаний**

В соответствии с учебным планом по профессиональному модулю **ПМ.05 Проектирование и разработка информационных систем** предусмотрен текущий контроль во время проведения занятий и промежуточная аттестация в форме дифференцированного зачета, курсового проекта, с выставлением итоговой оценки за весь курс.

### **2. Задания для контроля и оценки результатов освоения практического опыта, умений и усвоения знаний**

*Задания для контроля и оценки результатов освоения практического опыта, умений и усвоения знаний по учебной практике и по производственной практике (по профилю специальности) представлены в программах практик.*

#### **2.1.Задания для текущего контроля**

#### **ПМ.05 Проектирование и разработка информационных систем**

#### **МДК. 05.01 Проектирование и дизайн информационных систем**

#### **Тема 5.1.1. Основы проектирования информационных систем**

#### **Практическая работа 1 «Анализ предметной области различными методами: контент-анализ, вебметрический анализ, анализ ситуаций, моделирование и др.»**

**Цель работы:** получить навыки анализа предметной области различными методами.

#### *Краткие теоретические сведения*

Основным назначением информационных систем является обеспечение пользователей информацией о внешнем мире вне информационной системы. При этом получаемая пользователем информация описывает только определенный фрагмент (часть) внешнего мира. Такая часть внешнего мира получила название предметной области информационной системы.

Понятие предметной области ИС предполагает существование устойчивой во времени соотношенности между именами, понятиями и определенными реалиями внешнего мира, не зависящей от самой ИС и ее круга пользователей. Таким образом, введение в рассмотрение понятия предметной области ограничивает и делает обозримым информационное пространство ИС.

Предметная область ИС – часть реального мира, информация о которой, информация о которой используется в ИС. К предметной области ИС относятся предметы, объекты, явления реальной действительности, их свойства, функции, отношения между ними. Например, предметной областью может быть услуга, оказываемая предприятием. Тогда информация, которая будет использоваться в ИС, может содержать следующие сведения об услуге:

- название услуги;
- пространственные и временные параметры услуги;
- цена услуги и др.

Другим примером предметной области могут служить сотрудники предприятия. В этом случае в ИС будет использоваться следующая информация:

- фамилия, имя, отчество сотрудника;
- дата его рождения;
- должность;
- стаж работы;
- служебная характеристика и др.

В качестве предметной области может быть выбран какой-либо вид деятельности предприятия, например – складское хранение товаров. Информация об этой предметной области будет значительно больше по объему, так как должна описывать характеристики

хранящихся товаров, мест хранения, сотрудников, документов для приема и передачи товаров и др.

Анализ предметной области

Анализ предметной области является первым шагом, с которого начинается разработка ИС.

Анализ предметной области – деятельность, направленная на выявление информационных потребностей пользователей проектируемой ИС. В ходе анализа предметной области разработчики ИС должны:

- научиться понимать язык, на котором говорят заказчики;
- выяснить требования заказчика к ИС;
- определить набор задач для автоматизации.

Для этого разработчики с участием заказчика решают следующие основные задачи:

- изучают миссию предприятия, цели деятельности и стратегии их достижения;
- исследуют функции предприятия, их распределение между подразделениями предприятия;
- выполняют статическое описание предприятия (объектной, функциональной, организационной структуры, структуры управления);
- выполняют динамическое описание предприятия (бизнес-функций и потоков деятельности);
- моделируют отдельные бизнес-процессы на предприятии.

Результатом анализа предметной области является детальное описание данных информационных потребностей пользователей ИС.

Основными методами сбора информации для проведения анализа предметной области являются:

- изучение документов;
- анкетирование;
- наблюдение;
- интервьюирование;
- экспертные оценки;
- эксперименты.

### **Задание**

1. Определить, к какому классу относится заданный вид информационных систем, ответив на вопросы:
  1. по характеру использования информации – это система... (управляющая)...
  2. по характеру обработки данных – это система...(информационно –поисковые)...
  3. по сфере применения – это система...(ИС организационного управления)....
  4. по способу организации – это система...(распределенные)...
  5. по уровню и масштабу решаемых задач – это система...(корпоративная)..
2. Сравнить современные информационные систем, относящихся к заданному виду, составить таблицу общностей и отличий между информационными системами. Указать на их индивидуальные особенности, различающиеся количественные и качественные характеристики. Указать на характеристики и свойства, которые являются общими для всех рассматриваемых ИС.  
Например: СУБД Oraclev 7, System10, MySQL, PostgreSQL, INGRES/STAR, NoSQL-базы данных.
3. Проанализировать предметную область, выполнить структурное разбиение предметной области на отдельные подразделения (подсистемы) согласно выполняемым ими функциям.
4. Определить задачи и функции системы в целом и функции каждого подразделения (подсистемы).

5. Продумать подробное описание работы каждого подразделения (подсистемы), алгоритмов и сценариев выполнения ими отдельных работ.
6. Продумать виды входной и выходной информации для каждого подразделения (подсистемы).
7. Описать схему работы будущей информационной системы, учитывая выделенные и описанные ранее подсистемы.
8. Определить группу пользователей, для которой данная система будет более востребована.
9. Описать перечень функций системы, которые будут доступны данной группе пользователей.

*Варианты индивидуальных заданий.*

	Предметная область	Сущность задачи
1	Страховая медицинская компания	Страховая медицинская компания (СМК) заключает договоры добровольного медицинского страхования с населением и договоры с лечебными учреждениями на лечение застрахованных клиентов. При возникновении страхового случая клиент подает заявку на оказание медицинских услуг по условиям договора инспектору, который работает с данным клиентом. Инспектор направляет данного клиента в лечебное учреждение. Отчеты о своей деятельности инспектор предоставляет в бухгалтерию. Бухгалтерия проверяет оплату договоров, перечисляет денежные средства за оказанные услуги лечебным учреждениям, производит отчисления в налоговые органы и предоставляет отчетность в органы государственной статистики. СМК не только оплачивает лечение застрахованного лица при возникновении с ним страхового случая, но и, при возникновении каких-либо осложнений после лечения, оплачивает лечение этих осложнений
2	Агентство недвижимости	Агентство недвижимости занимается покупкой, продажей, сдачей в аренду объектов недвижимости по договорам с их собственниками. Агентство управляет объектами недвижимости как физических, так и юридических лиц. Собственник может иметь несколько объектов. В случае покупки или аренды клиент может произвести осмотр объекта. В качестве одной из услуг, предлагаемых агентством, является проведение инспектирования текущего состояния объекта для адекватного определения его рыночной цены. По результатам своей деятельности агентство производит отчисления в налоговые органы и предоставляет отчетность в органы государственной статистики

3	Кадровое агентство	Кадровое агентство способствует трудоустройству безработных граждан. Агентство ведет учет и классификацию данных о безработных на основании резюме от них. От предприятий города поступают данные о свободных вакансиях, на основании которых агентство предлагает различные варианты трудоустройства соискателям. В случае положительного исхода поиска вакансия считается заполненной, а безработный становится трудоустроенным. По результатам своей деятельности кадровое агентство производит отчисления в налоговые органы и предоставляет отчетность в органы государственной статистики
4	Компания разработке программных продуктов по	Компания заключает договор с клиентом на разработку программного продукта согласно техническому заданию. После утверждения технического задания определяется состав и объем работ, составляется предварительная смета. На каждый проект назначается ответственный за его выполнение – куратор проекта, который распределяет нагрузку между программистами и следит за выполнением технического задания. Когда программный продукт готов, то его внедряют, производят обучение клиента и осуществляют дальнейшее сопровождение. По результатам своей деятельности компания производит отчисления в налоговые органы и предоставляет отчетность в органы государственной статистики
5	Туроператор	Туроператор предоставляет возможность своим клиентам осуществить туристическую или деловую поездку в различные города России и мира. При разработке нового тура сначала анализируется текущая ситуация на рынке туризма и выбирается направление тура. После этого определяется статус тура, бронируются места в гостиницах и билеты на переезд к месту тура, разрабатывается культурная/ деловая/ развлекательная программа, утверждаются сроки тура. На каждый тур назначается ответственное лицо от туроператора, которое будет вести данный тур для улаживания проблем в случае возникновения каких-либо чрезвычайных или форс-мажорных ситуаций. Клиент приходит в офис туроператора, где вместе с менеджером выбирает уже разработанный тур и оформляет путевку. После возвращения из тура клиент может высказать свои замечания или пожелания, которые будут учтены при доработке существующих туров или при разработке новых. Также, для дальнейшего улучшения тура, туроператор проводит анализ отчетов от посредников (гостиница, гиды и т.д.). По результатам своей деятельности туроператор производит отчисления в налоговые органы и предоставляет отчетность в органы государственной статистики

### **Практическая работа 2«Изучение устройств автоматизированного сбора информации»**

**Цель работы:** научиться анализировать средства автоматизированного документирования; осуществлять постановку задач по обработке информации; проводить анализ предметной области; осуществлять выбор модели и средства построения информационной системы и

программных средств; проектировать интерфейс приложения с использованием программных средств.

Задание № 1 Изучить и описать технологии штрихового кодирования (Bar Code Technologies) сбора информации.

Задание № 2 Изучить и описать технологии радиочастотной идентификации (RFID – Radio Frequency Identification Technologies) сбора информации.

Задание № 3 Изучить и описать карточные технологии (Card Technologies) сбора информации.

Задание № 4 Изучить и описать технологии сбора данных (Data Communications Technologies).

Задание № 5 Изучить и описать технологии распознавания голоса, оптического и магнитного распознавания текста, биометрические технологии и некоторые другие.

Задание № 6 В зависимости от целей, сферы деятельности и располагаемых технических средств можно выделить методы сбора данных, применяемые:

- 1) в экономических информационных системах (например, маркетинга);
- 2) в геоинформационных системах;
- 3) в статистических информационных системах;
- 4) в информационных системах управления производственными процессами.

### **Практическая работа 3 «Оценка экономической эффективности информационной системы»**

**Цель:** изучение методов оценки экономической эффективности информационных систем.

*Краткие теоретические сведения*

Внедрение информационных технологий сопряжено с капитальными вложениями как на приобретение техники, так и на разработку проектов, выполнение подготовительных работ и подготовку кадров. Поэтому внедрению должно предшествовать экономическое обоснование целесообразности внедрения информационных систем (ИС).

Обобщенным критерием экономической эффективности является минимум затрат живого и овеществленного труда.

При этом установлено, что чем больше участков управленческих работ автоматизировано, тем эффективнее используется техническое и программное обеспечение.

Экономический эффект от внедрения вычислительной и организационной техники подразделяют на прямой и косвенный.

Под прямой экономической эффективностью понимают экономию материально-трудовых ресурсов и денежных средств, полученную в результате сокращения численности управленческого персонала, фонда заработной платы, расхода основных и вспомогательных материалов вследствие автоматизации конкретных видов планово-учетных и аналитических работ.

Не исключено, что внедрение АИТ на первом этапе не приведет к уменьшению числа работников планово-учетных служб. В этом случае учитывают косвенную эффективность, проявляющуюся в конечных результатах хозяйственной деятельности предприятия. Ее локальными критериями могут быть: сокращение сроков составления сводок, повышение качества планово-учетных и аналитических работ, сокращение документооборота, повышение культуры и производительности труда и т. д. Основным же показателем является повышение качества управления, которое, как и при прямой экономической эффективности, ведет к экономии живого и овеществленного труда. Оба вида рассмотренной экономической эффективности взаимосвязаны.

Задание № 1

Охарактеризуйте затратные методы оценки экономической эффективности информационных систем.

Задание № 2 Охарактеризуйте методы оценки прямого результата информационных систем.

Задание № 3 Охарактеризуйте методы оценки экономической эффективности информационных систем, основанные на оценке идеальности процесса.

Задание № 4 Охарактеризуйте квалиметрические подходы к оценке экономической эффективности информационных систем, основанные на оценке идеальности процесса.

Задание № 5 Проведите сравнительный анализ методов оценки экономической эффективности информационных систем.

Задание № 6 Рассчитайте экономическую эффективность заданной информационно системы (вариант информационной системы см. выше).

*Вопросы для проверки*

1. Понятие экономической эффективности информационных систем.
2. Методы оценки экономической эффективности информационных систем. Расчет экономической эффективности информационных систем.
3. Какие критерии эффективности используют при выборе средств системы?
4. Как должна быть рассчитана эффективность применения автоматизированных информационных технологий?
5. Что понимают под эффективностью автоматизированного преобразования информации?
6. Что называют обобщенным критерием экономической эффективности?
7. Что входит в прямую экономическую эффективность?
8. Что составляет косвенную эффективность автоматизированных информационных технологий?
9. Что определяет стандарт для обмена данными о промышленных изделиях?
10. Как рассчитывается экономическая эффективность информационных технологий?

#### **Практическая работа 4«Разработка модели архитектуры информационной системы»**

**Цель работы:** приобретение навыков моделирования архитектуры информационной системы

*Краткие теоретические сведения*

Унифицированный язык моделирования UML – это графический язык моделирования общего назначения, предназначенный для спецификации, визуализации, проектирования и документирования всех компонентов, создаваемых при разработке программных систем.

Язык UML является объектно-ориентированным языком. Его использование основывается на понимании общих принципов объектно-ориентированного анализа и проектирования:

1. Принцип абстрагирования предписывает включать в модель только те аспекты проектируемой системы, которые имеют непосредственное отношение к выполнению системой своих функций.
2. Принцип многомодельности означает, что никакое единственное представление системы не является достаточным для адекватного выражения всех ее особенностей.
3. Принцип иерархического построения моделей сложных систем предписывает рассматривать процесс построения моделей на разных уровнях абстрагирования или детализации в рамках фиксированных представлений.

Диаграмма UML – это графическое представление набора элементов, изображаемое в виде связанного графа с вершинами (сущностями) и ребрами (отношениями), используемое для визуализации системы с разных точек зрения.

Диаграммы UML используются для описания различных аспектов функционирования и структуры ИС на разных стадиях создания системы и, соответственно, на разных этапах моделирования: концептуального, логического и физического.

*Порядок выполнения работы*

Вариант индивидуального задания определяет ИС, для создания которой необходимо разработать совокупность моделей системы в виде комплекта диаграмм *UML*. Построенные модели ИС должны описывать различные аспекты проектирования и разработки системы на разных стадиях её жизненного цикла.

В процессе выполнения лабораторной работы необходимо:

1. Разработать модель прецедентов, описывающую бизнеспроцессы организации с точки зрения внешнего пользователя (клиента) и отражающую взгляд на деятельность организации извне. Результатом моделирования являются диаграммы деятельности и диаграммы прецедентов.
2. Разработать модель бизнес-объектов, описывающую выполнение бизнес-процессов организации ее внутренними исполнителями. Основными компонентами модели являются внешние и внутренние исполнители. Результатом моделирования являются диаграммы последовательности.
3. Разработать концептуальную модель данных, описывающую объекты предметной области и связи между ними. Результатом моделирования являются диаграммы классов и диаграммы объектов.
4. Разработать описание требований к системе. Результатом является исчерпывающий перечень функций, которые должны быть реализованы в системе, и подробное описание необходимой реализации этих функций.
5. Разработка моделей базы данных и приложений, представляющих собой детальное описание проекта базы данных и клиентских приложений информационной системы. Результатом моделирования являются диаграммы компонентов и диаграммы базы данных.
6. Разработать проект физической реализации информационной системы. Результатом проектирования являются диаграммы развёртывания и диаграммы компонентов.

#### *Варианты индивидуальных заданий*

В качестве списка вариантов индивидуальных заданий используется перечень информационных систем из лабораторной работы № 3.

1. ИС «Телефонный справочник» (поисковая система).
2. ИС «Библиотека» (информационно-справочная система, поисковая система).
3. ИС «Издательство» (СЭДО, САБП).
4. ИС «Поликлиника» (СЭДО, информационно-справочная система).
5. ИС «Школа» (обучающая система, информационносправочная система).
6. ИС «Ателье» (САБП).
7. ИС «Склад» (САБП).
8. ИС «Торговля» (САБП, СЭДО).
9. ИС «Автосалон» (САБП, СЭДО).
10. ИС «Продажа подержанных автомобилей» (информационно-справочная система, поисковая система).
11. ИС «Автосервис» (САБП).
12. ИС «Пассажирское автопредприятие» (САБП, СЭДО).
13. ИС «Диспетчерская служба такси» (ГИС, СЭДО).
14. ИС «Агентство по продаже авиабилетов» (информационно-справочная система, поисковая система).
15. ИС «Туристическое агентство» (информационносправочная система, поисковая система).
16. ИС «Гостиница» (информационно-справочная система, СЭДО).

#### **Практическая работа 5 «Обоснование выбора средств проектирования информационной системы»**

**Цель работы:** изучение средств проектирования информационных систем

*Краткие теоретические сведения*

Существует четыре пути внедрения информационной системы:

- приобретение готового проектного решения (типового пакета прикладных программ);
- адаптация существующего проектного решения;
- разработка нового проекта силами посторонней организации;  разработка нового проекта силами сотрудников самой фирмы.

На рынке информационных систем представлено несколько вариантов продуктов, способных после соответствующей адаптации решить поставленные задачи.

**Задание №1**

Проведите сравнительный анализ информационных систем, представленные на российском рынке, результат занесите в таблицу

	Инталев	Галактика	"1С: Предприятие 8.0."	BPlan	Контур Корпорация
Поддержка многопользовательской коллективной работы					
Функциональность системы					
Затраты на внедрение и сопровождение					
Область применения					
Используемая архитектура					
Иерархическая структура для работы со статьями					
Поддержка многомерного анализа данных (OLAP)					
Наличие удаленного доступа через Web					
Требования к производительности клиентской Части					

**Задание №2**

Проведите сравнительный анализ средств разработки приложений, результат занесите в таблицу

Параметр	Инструментальная среда		
	Delphi	C#	C++
Название, версия, фирма производитель.			
Поддерживаемые операционные системы.			
Требования к аппаратному обеспечению.			
Встроенный язык.			
Поддержка стандарта SQL.			
Поддержка ООП.			
Механизмы доступа к БД.			
Наличие визуальных и не визуальных компонент для работы с БД.			

Средства построения отчетов.			
Наличие средств построения диаграмм.			
Поддержка технологии Rapid Application Development.			
Средства поддержки транзакций.			
Утилиты для работы с БД.			
Возможность создания исполняемого файла.			
Сложность или простота работы с инструментальным средством.			

### **Практическая работа 6 «Описание бизнес-процессов заданной предметной области»**

**Цель работы:** сформировать список бизнес-процессов заданной предметной области.

**Задание.** Необходимо создать сценарий процесса привлечения и размещения ресурсов информационной системы. Собрать информацию по отделениям ИС, проверить и при необходимости скорректировать показатели полученного сводного плана ресурсов, составить планы привлечения и размещения ресурсов по ИС в целом и по его отделениям.

*Ход работы*

- 1 Выделение функциональных блоков (функций процесса).
- 2 Выделение связей между функциями.
3. Выделение действий или подпроцессов моделируемой системы.
4. Определение последовательности выполнения выделенных действий.

*Варианты индивидуальных заданий*

В качестве списка вариантов индивидуальных заданий используется перечень информационных систем из лабораторной работы № 3.

### **Тема 5.1.2. Система обеспечения качества информационных систем**

#### **Практическая работа 7 «Построение модели управления качеством процесса изучения модуля «Проектирование и разработка информационных систем»**

**Цель работы:** получение навыков построения модели управления качеством.

**Задание № 1** Привести национальные стандарты обеспечения качества автоматизированных информационных систем. Охарактеризовать международную систему стандартизации и сертификации качества продукции.

**Задание № 2** Описать стандарты группы ISO.

**Задание № 3** Привести методы контроля качества в информационных системах.

**Задание № 4** Постройте модель управления качеством процесса изучения модуля «Проектирование и разработка информационных систем».

#### **Практическая работа 8 «Реинжиниринг методом интеграции»**

**Цель работы:** Освоить методы организации работ по реинжинирингу бизнес-процессов на различных этапах, взаимодействия участников РБП.

**Задание 1.** Описать последовательность этапов РБП и содержание каждого из этапов:

- постановка проблемы и выделение бизнес-процессов
- идентификация бизнес-процессов, - исследование существующей организации бизнес-процессов
- обратный инжиниринг
- проектирование новых бизнес-процессов
- прямой инжиниринг, разработка новой организационной структуры и корпоративной информационной системы

- реализация проекта, обучение персонала, наладка и опытная эксплуатация информационной системы  
- внедрение проекта.

**Задание 2.** Сформулировать функциональные обязанности участников РБП на различных этапах РБП:

лидера проекта (менеджера верхнего уровня управления),  
регламентирующего комитета (контрольно-распорядительного органа),  
методологического центра (координирующего и обеспечивающего органа),  
команд реинжиниринга (смешанных групп экспертов предприятия и консультантов),  
менеджеров процессов (будущих ключевых исполнителей).

*Варианты индивидуальных заданий*

В качестве списка вариантов индивидуальных заданий используется перечень информационных систем из лабораторной работы № 3.

### **Практическая работа 9 «Разработка требований безопасности информационной системы»**

**Цель работы:** получение навыков разработки требований безопасности информационной системы

**Задание 1** Определите Цель и задачи системы защиты информации.

**Задание 2** Перечислите факторы, влияющие на организацию системы защиты информации.

**Задание 3** Определите дестабилизирующие воздействия на информационную систему и способы их нейтрализации.

**Задание 4** Разработайте требования безопасности информационной системы.

**Задание 5** Выберите методы и средства защиты информации для исследуемой информационной системы.

*Варианты индивидуальных заданий*

В качестве списка вариантов индивидуальных заданий используется перечень информационных систем из лабораторной работы № 3.

### **Практическая работа 10 «Реинжиниринг бизнес-процессов методом горизонтального и/или вертикального сжатия»**

**Цель работы:** научиться использовать метод горизонтального и/или вертикального сжатия для реинжиниринга бизнес-процессов.

*Краткие теоретические сведения:*

Создание модели «Как есть».

Определение видов анализируемых объектов: материальные потоки; финансовые потоки; потоки документов; информационные потоки.

Формирование вопросов для построения схемы бизнес-процессов для каждого подразделения:

- выполняемые функции;
- ответственные за выполнение процесса;
- результат работ.

Анализ.

Основные параметры:

1. полнота – необходимые данные, ресурсы и механизмы управления;
2. однозначность – понимание последовательности реализации процесса;
3. определенное количество участников процесса, последовательность.

Создание модели «Как будет».

Критерий оптимизации:

Построение системы обеспечивающей достижение компанией поставленных целей, при условии минимизации расходов на ее создание.

Возможные варианты решения:

1. объединение – задачи объединяются в одну, выполняемую одним специалистом;
2. делегирование – принимаются самостоятельные решения;
3. параллельность: некоторые этапы работы параллельно;
4. контроль: точки контроля должны быть там, где ошибки могут привести к существенным затратам.

#### Внедрение.

Здесь определяются недостающие фрагменты будущей системы и избыточные элементы существующей. Данный шаг определяет перечень мероприятий, образующий план перехода.

Учитываются важные характеристики реинжиниринга:

преимущества: повышение качества и скорости производства продукта с одновременным снижением профессионализма сотрудников, повышение конкурентоспособности компании;

недостатки: необходимость проведения работы по изменению корпоративной культуры;

**Задание** Построить модели «Как есть» и «Как будет».

*Варианты индивидуальных заданий*

В качестве списка вариантов индивидуальных заданий используется перечень информационных систем из лабораторной работы № 3.

### **Тема 5.1.3. Разработка документации информационных систем**

#### **Практическая работа 11 «Проектирование спецификации информационной системы по индивидуальному заданию»**

**Цель работы:** получить навыки проектирования спецификации информационной системы по индивидуальному заданию.

**Задание:** для выбранного варианта информационной системы определить набор требований и спецификаций на создание информационной системы.

*Указания к работе:*

Для успешной реализации базы данных объект проектирования должен быть прежде всего адекватно описан, должны быть построены полные и непротиворечивые функциональные модели базы данных. Опыт проектирования информационных систем показывает, что это логически сложная, трудоемкая и длительная по времени работа, требующая высокой квалификации участвующих в ней специалистов.

При проектировании информационной системы необходимо провести анализ целей этой системы и выявить требования к ней отдельных пользователей. Информация для построения модели информационной системы берется на основе проведения всестороннего обследования организации, для которой выполняется разработка информационной системы. Сбор данных начинается с изучения сущностей предметной области, процессов, использующих эти сущности, и связей между ними

Последовательность выполнения работы:

1. Ознакомьтесь с предложенным вариантом описания предметной области. Проанализируйте предметную область, уточнив и дополнив ее, руководствуясь собственным опытом, консультациями и другими источниками.
2. Выполните структурное разбиение предметной области на отдельные подразделения (отделы, службы, подсистемы, группы и пр.) согласно выполняемым ими функциям.
3. Определите задачи и функции системы в целом и функции каждого подразделения (подсистемы).
4. Выполните словесное описание работы каждого подразделения (подсистемы), алгоритмов и сценариев выполнения ими отдельных работ.
5. Оформите отчет со следующими разделами:
  - исходное задание;
  - расширенное описание предметной области с учетом сделанных дополнений;

- состав подразделений (подсистем) информационной системы;
- перечень функций и задач системы в целом и каждого подразделения (подсистемы) в отдельности, отношения их между собой;
- подробное описание работы каждого подразделения (подсистемы);
- описание отдельных сценариев работ подразделений (подсистем);
- входная и выходная информация для каждого подразделения (подсистемы).

*Варианты информационных систем (ИС)*

1. ИС «Центр бытового обслуживания».
2. ИС «Библиотека».
3. ИС «Издательство».
4. ИС «Поликлиника».
5. ИС «Учебный центр».
6. ИС «Ателье».
7. ИС «Склад».
8. ИС «Автосалон».
9. ИС «Автосервис».
10. ИС «Туристическое агентство».
11. ИС «Гостиница».

**Практическая работа 12 «Разработка общего функционального описания программного средства по индивидуальному заданию»**

**Цель работы:** получить навыки разработки общего функционального описания программного средства по индивидуальному заданию; осуществлять постановку задач по обработке информации; проводить анализ предметной области; осуществлять выбор модели и средства построения информационной системы и программных средств.

*Краткие теоретические сведения*

Общее функциональное описание программного средства дает краткую характеристику функциональных возможностей программного средства. Предназначено для пользователей, которые должны решить, насколько необходимо им данное программное средство (ПС).

**Задание.** Составьте общее описание руководства пользователей программного средства, которое включает в себя:

- краткое описание функций и характеристик ПС;
- описание внешней программной среды;
- перечень файлов, включая файлы базы данных, необходимых для применения ПС;
- порядок действий для продолжения или возобновления функционирования ПС в случаях возникновения непредвиденных ситуаций;
- организация и функционирование ПС с точки зрения пользователя;
- описание процедур, позволяющих фиксировать дефекты и ошибки;
- детальные, пошаговые действия пользователя при включении системы и дальнейшей работе с ней;
- ссылки на другие руководства системы и комплекса программ;
- перечень и пояснение выводимых системой сообщений.

*Варианты информационных систем (ИС)*

1. ИС «Центр бытового обслуживания».
2. ИС «Библиотека».
3. ИС «Издательство».
4. ИС «Поликлиника».
5. ИС «Учебный центр».
6. ИС «Ателье».
7. ИС «Склад».

8. ИС «Автосалон».
9. ИС «Автосервис».
10. ИС «Туристическое агентство».
11. ИС «Гостиница».

### **Практическая работа 13 «Разработка руководства по инсталляции программного средства по индивидуальному заданию»**

**Цель работы:** получить навыки разработки руководства по инсталляции программного средства.

*Краткие теоретические сведения.*

Необходимость в инсталляции приложения для возможности его использования обуславливается тем, что в изначальной форме оно размещается на носителе в сжатой форме. Установка выполняется специальным пакетным менеджером или инсталлятором и включает в себя:

- копирование файлов с носителя в память жесткого диска компьютера;
- проверку ПК на совместимость;
- распаковку архива;
- размещение всей файлов в правильных папках и в правильной очередности;
- создание самостоятельно выполняющихся команд в реестре и автозагрузке;
- создание конфигурационных файлов, позволяющих менять настройки программы в дальнейшем по желанию пользователя.

Чаще всего все эти процессы невидимы для пользователя и выполняются автоматически. Однако это зависит от вида инсталляции.

Процесс инсталляции не всегда проходит одинаково. В зависимости от особенностей программы, производители прописывают разные особенности её установки на компьютере. Используются такие виды инсталляции ПО:

- Ручная – инсталляция при отсутствии автоматического установщика, требующая выполнения большого количества операций непосредственно пользователем.
- Тихая – инсталляция, при которой отключается отображение сообщений и всплывающих окон системы и других приложений.
- Автоматическая – от пользователя требуется лишь запустить процесс инсталляции и, в некоторых случаях, провести начальные настройки и подтвердить согласие с лицензионным соглашением, в то время как все остальные требуемые операции будут совершены самой программой. Это наиболее распространенный способ установки на сегодня.
- Удаленная – инсталляция, которая происходит на другом устройстве, присоединенном по локальной сети. Является частым выбором для компьютеров, не имеющих экрана. Таким методом чаще всего пользуются, когда есть необходимость установить программу на большое число рабочих компьютеров в крупных корпорациях. Также используется для настройки серверов.
- Самостоятельная – запуск установки изначально прописан в файловой системе программы, что позволяет инсталляции начаться сразу же, без дополнительных действий со стороны пользователя. Чаще всего так инсталлируются драйвера, которые не являются программами, но обеспечивают их правильную работу.
- Чистая – инсталляция таким путем, который подходит для каждой компьютерной программы и не имеющим никаких отличий в зависимости от устанавливаемого приложения.
- Непосредственная – установка программы производится используя файлы, уже имеющиеся в памяти компьютера, не прибегая к внешнему доступу через дисковод или USB.

Руководство по инсталляции программного средства должно детально предписывать, как устанавливать системы в конкретной среде. Оно должно содержать описание машинно-

считываемого носителя, на котором поставляется программное средство, файлы, представляющие программное средство, и требования к минимальной конфигурации аппаратуры.

**Задание.** Составьте общее описание руководства по установке ПО (вариант см. в предыдущей лабораторной работе), которое включает в себя:

- концепции и обзоры системного управления программами и базами данных;
- документы, детализирующие концепцию процессов управления системой и ПС и требования к реализации каждой функции;
- информационная модель системы, комплекса программ, их атрибутов и операций;
- руководства для формализации и описания объектов управления системы и ПС;
- формализация непосредственной передачи управляющей информации между компонентами системы и ПС;
- документы, описывающие:
  - передаваемые типы данных;
  - формализованные объекты, их состояния, атрибуты, операции и извещения об обмене;
- классификатор объектов управления, отражающий взаимосвязь между классами объектов управления и правилами их применения;
- функции администратора программных средств;
- общие функции администрирования при применении данного ПС;
- процедуры по установке и подготовке ПС к эксплуатации;
- контроль ввода заданий и выработки запроса на их выполнение;
- контроль представления результатов обработки заданий;
- способы и формы контроля исполнения заданий;
- динамическое управление процессом реализации заданий.

#### **Практическая работа 14 «Разработка руководства пользователя программного средства по индивидуальному заданию»**

**Цель работы:** получить навыки разработки руководства пользователя программного средства.

Краткие теоретические сведения. Инструкция по использованию программного обеспечения (или «Инструкция пользователю», или «Руководство для пользователя») - это выдержка из полной документации, предназначенная для эксплуатации программного обеспечения. Она представляет собой независимый документ, в котором описывается: что делает программное обеспечение и как им пользоваться.

«Инструкция пользователю» должна содержать всю необходимую для пользователя информацию и должна быть ему понятна без дополнительных материалов (без обращения к другим спецификациям). Следовательно, необходимая для этой инструкции информация переписывается полностью из соответствующих спецификаций.

**Задание.** Составьте общее описание руководства пользователей программного средства (вариант см. в предыдущей лабораторной работе), которое включает в себя:

- наименование программного обеспечения;
- краткое описание программного обеспечения;
- перечень выполняемых им функций;
- краткую характеристику метода (или методов), его достоинство и недостатки;
- полную библиографическую ссылку на полное описание метода;
- описание входных и выходных данных.

Вторая часть документа должна описывать порядок работы с программным обеспечением. Она должна содержать описание всех режимов работы программного обеспечения, а также содержание всех печатей и диагностических сообщений, которые выдаются по ходу выполнения программы.

Следует помнить, что пользователь по своей квалификации не является программистом и поэтому его работа с программным обеспечением описывается на понятном ему языке и достаточно подробно, а именно:

- как запустить программное обеспечение;
- как продолжить работу с программным обеспечением (описывается подробный интерактивный режим его работы с ПО);
- подготовка и ввод исходных данных в программное обеспечение;
- как реагировать на запросы программного обеспечения;
- как вести работу в исключительных ситуациях;
- как реагировать на ошибки;
- как восстановить работу программного обеспечения в случае аварийного его завершения;
- как получить требуемый результат;
- как правильно закончить работу с программным обеспечением (запланированный программой выход).

### **Лабораторная работа 1«Изучение средств автоматизированного документирования»**

**Цель:** Изучение средств автоматизированного документирования

*Краткие теоретические сведения*

Автоматизированные системы документооборота не только обеспечивают сохранность документации, но и позволяют управлять ей на протяжении всего жизненного цикла. Многие организации уже оценили все плюсы современных технологий и успешно используют подобные системы в делопроизводственных процессах.

Автоматизация документооборота как направление делопроизводства является следствием роста числа официальной документации, сопровождающей деятельность любой компании. Система автоматизации или система электронного документооборота (СЭД) обеспечивает сопровождение управленческих процессов, позволяет автоматизировать работу с документами. Ее объектами являются не только электронные и бумажные документы, но и бизнес процессы, находящие отражение в их движении.

Автоматизация электронного документооборота на предприятии преследует следующие цели:

- минимизация рутинных операций и сокращение количества ручной работы; формирование единого массива электронных документов и создание удобной поисковой системы;
- уменьшение и полная ликвидация бумажных носителей информации;
- разработка алгоритмов прохождения документов в соответствии с технологическими процессами;
- поддержка административных процессов;
- быстрый обмен внутренней документацией между сотрудниками и подразделениями;
- ускорение формирования указаний и распоряжений, а также контроля их выполнения;
- планирование распределения рабочего времени персонала, контроль деятельности сотрудников;
- разработка технологии трансфера служебной документации через внешние системы; приведение схемы делопроизводства к единому алгоритму.

Система электронного документооборота имеет ряд неоспоримых преимуществ по сравнению с традиционным подходом к делопроизводству:

- повышение производительности за счет быстрого доступа к любой категории документов; поддержание актуальности информации;
- снижение влияние “человеческого фактора”;
- сокращение материальных издержек, связанных с созданием и хранением документов;
- создание условий для эффективного взаимодействия между подразделениями;

- возможность коллективной работы над служебной документацией; уменьшение потребностей в персонале; снижение рисков потери или повреждения информации;
- автоматизация составления отчетов; интеграция системы с офисными программами.

Система управления документооборотом представляет собой программное обеспечение, позволяющее персоналу работать электронными документами (создавать, изменять, осуществлять поиск) и взаимодействовать между собой (передавать документы, отправлять уведомления, выдавать задания).

Используемые в процессе автоматизации современные технологии позволяют осуществить классификацию СЭД. Принято выделять следующие виды систем:

- клиент-серверные (основные модели управления размещены на выделенном сервере, пользователи взаимодействуют с СЭД через специальный интерфейс, клиентскую часть);
- действующие на основе баз данных (интегрированы с базами SQL, Oracle и т.п., информация хранится в базе, а для ее обработки используются отдельные модули);
- основанные на технологиях web (обеспечивают удаленный доступ к серверу без специальных клиентских приложений, позволяют использовать web браузеры в качестве пользовательского интерфейса);
- «облачные» системы (используют сервер хостинг-провайдера).

Среди современных технологий, обеспечивающих эффективное функционирование СЭД можно выделить:

- хранение электронных документов; поточное сканирование;
- оптическое распознавание текста;
- штрих-кодирование; электронно-цифровые подписи;
- полнотекстовый и атрибутивный поиск.

Слияние документов предназначено для создания писем, рассылок, поздравлений и др. документов, содержащих основной однотипный текст и поле, в которое вставляются данные из внешнего источника:

- в основном документе содержится текст, являющийся одинаковым для всех выходных документов. Он может содержать бланк, текст и инструкции в полях слияния для вставки текста (например, имена получателей и адреса), который будет различаться в конечных документах;
- список получателей является базой данных, содержащей данные, которые объединены в выходных документах.
- База данных обычно является списком имён, адресов, телефонных номеров и других категорий персональных данных;
- результатом слияния являются выходные документы. Текст в документе вывода может быть одинаковым во всех выходных документах, но можно применить форматирование к отдельным документам.

**Задание:** реализовать автоматизированное слияние документов в MS Word для создания и печати писем с использованием данных из листа MS Excel.

*Выполнение работы*

#### 1. Настройка файла данных MS Excel

Создайте книгу MS Excel, содержащую базу данных адресов. Необходимо соблюдать следующие требования:

- имя каждого поля должно быть уникальным;
- каждая строка должна содержать сведения о каком-либо объекте / субъекте;
- таблица не должна содержать пустые строки.

Создайте файл данных MS Excel, закройте и сохраните его под именем Адреса.xlsx.

Фамилия	Имя	Отчество	Индекс	Регион	Город	Улица	Номер дома / номер квартиры
Иванов	Иван	Иванович	454000	Челябинская обл.	Челябинск	Солнечная	11/12
Петров	Петр	Петрович	653000	Тюменская обл.	Тюмень	Морозная	15/16
Сидорова	Анна	Андреевна	521000	Курганская обл.	Шадринск	Летняя	25/35
Кузнецова	Ирина	Ивановна	347000	Оренбургская обл.	Оренбург	Северная	45/12

## 2. Создание основного документа

Создайте документ в программе MS Word, который будет служить основой для письма. Выберите пункт *Рассылки – Начать слияние – Пошаговый мастер слияния.*

Получатель:
Индекс:
Регион:
Город:
Улица:
Дом/квартира:
Уважаемый
Уведомляем Вас о закрытии нашего филиала нашего магазина. Всю информацию Вы можете <u>подучить</u> а сайте нашей организации.
С уважением, директор магазина «Художественные изделия»
Е.С. Яковлев

Для ввода данных из списка выберите *Обзор*.

Укажите путь к сохраненному файлу Адреса.xlsx. Укажите Лист 1 и нажмите ОК.

При желании можно снять флажок рядом с фамилией человека, которому не нужно отправлять письмо.

Уберите флажки рядом с пустыми строками.

Нажмите ОК.

Выберите *Далее*. Создание письма.

Поставьте курсор под строкой *Получатель* и выберите *Другие элементы*.

Установите курсор справа от *Фамилия* и выберите *Вставить поле слияния*.

Вставьте последовательно поля: фамилия, имя и отчество.

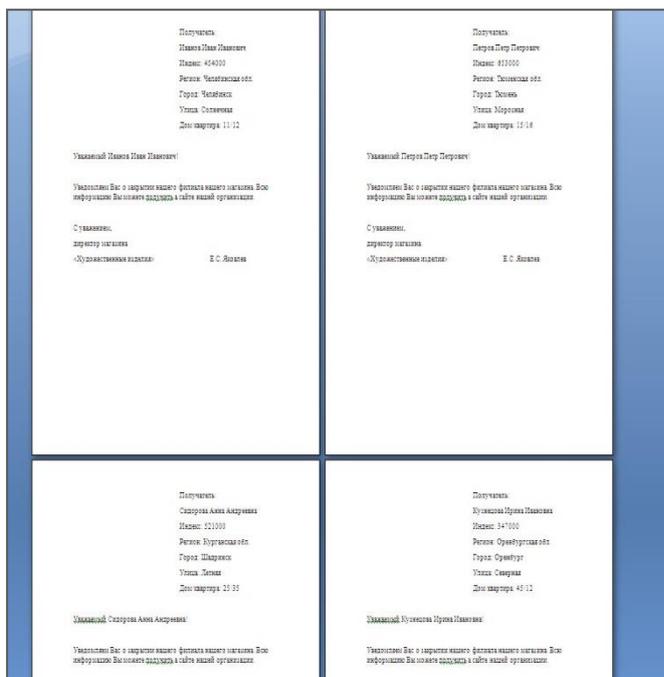
Аналогично вставьте остальные поля и нажмите *Далее.Просмотр писем*.

Фамилию, имя и отчество при вставке разделяйте пробелами.

Убедитесь в правильности документа и нажмите *Далее.Завершение слияния*.

Для просмотра результата нажмите *Изменить часть писем*.

Просмотрите результат:



**Задание 2** Создайте поздравительные открытки с именными поздравлениями, используя имеющийся список адресатов.

**Задание 3** Создайте конверты соответствующего формата с адресами для отправки открыток.

## МДК. 05.02. Разработка кода информационных систем

### Тема 5.2.1. Основные инструменты для создания, исполнения и управления информационной системой

#### Лабораторная работа 1 «Построение диаграммы Вариантов использования и диаграммы последовательности и генерация кода»

**Цель работы:** приобретение навыков построения диаграммы Вариантов использования и диаграммы последовательности.

*Краткие теоретические сведения.*

Диаграмма прецедентов/вариантов использования (use case diagram) – это один из видов диаграмм, которые применяются для моделирования динамических аспектов системы, подсистемы, класса. Данный вид диаграмм имеет большое значение для визуализации, специфицирования, документирования поведения систем, подсистем, классов и облегчает взаимодействие пользователя и аналитика. Диаграммы прецедентов/вариантов использования применяются при бизнес–анализе для моделирования процессов предметной области, формирования функциональных требований и направления их реализации.

Последовательность выполнения работы

- Изучить характеристики объекта автоматизации.
- Построить диаграмму вариантов использования (UsecaseDiagram).
- Сформировать отчет по лабораторной работе.

#### **Задание**

Для условной предметной области «Предприятие по сборке и продаже компьютеров» создайте структурированное описание действующих лиц/актеров, прецедентов/вариантов использования, главную диаграмму прецедентов/вариантов использования, показав на ней прецеденты/варианты использования, действующих лиц/актеров, отношения между элементами, добавить описания к действующим лицам/актерам и вариантам использования, для каждого (отдельных) варианта использования задайте поток событий в виде отдельного файла и прикрепите его к варианту использования.

Создайте Диаграмму прецедентов/вариантов использования с указанием Действующих лиц/Актеров, прецедентов, отношений между элементами.

При создании диаграммы необходимо учесть, что:

- Действующее лицо/Актер Инженер по сборке связан с Действующими лицами/Актерами Инженер по сборке настольных компьютеров и Инженер по сборке ноутбуков отношением обобщения, так как выступает в качестве родительского прецедента/варианта использования,
- между прецедентами Сборка компьютеров и Требование необходимых комплектующих установлено отношение включения, поскольку для сборки компьютеров обязательно нужно заказывать необходимые комплектующие со склада,
- для прецедента Работа с заказом показано отношение расширения (extend) с базовым прецедентом Управление информацией о клиенте, поскольку, когда актер Менеджер по работе с клиентами работает с заказом (оформляет, корректирует и т.д.), то не всегда при этом он управляет информацией о клиентах,
- для всех остальных прецедентов/вариантов использования использовано отношение ассоциации.

Описание действующих лиц/актеров предметной области приведено в таблице 1.

Таблица 1. Предприятие по сборке и продаже компьютеров

Действующее лицо/Актер	Краткое описание
Коммерческий директор	Сотрудник, который руководит и координирует процессы закупок, поставок, продаж, принимает стратегические решения в области сбыта, ценообразования, маркетинговой, ассортиментной политики.
Бухгалтер	Сотрудник, который осуществляет организацию бухгалтерского учета хозяйственно-финансовой деятельности и контроль за экономным использованием материальных, трудовых и финансовых ресурсов, сохранностью собственности предприятия.
Кассир	Сотрудник, который осуществляет операции по приему, учету, выдаче и хранению денежных средств и ценных бумаг, составляет кассовую отчетность.
Менеджер по работе с клиентами	Сотрудник, который ведет переговоры с клиентами, следит за ассортиментом продукции, проводит документальное оформление покупки, осуществляет контроль над комплектацией заказа, расчетом за укомплектованный заказ клиента.
Менеджер по снабжению	Сотрудник, который изучает оперативную маркетинговую информацию и рекламные материалы, подготавливает и корректирует преддоговорную документацию, проводит переговоры с поставщиками, обеспечивает доставку материальных ресурсов от поставщиков и продавцов, организует обработку поставок материально-технических ресурсов.
Инженер по сборке ПК	Сотрудник, который занимается сборкой настольных компьютеров.
Инженер по сборке ноутбуков	Сотрудник, который занимается сборкой ноутбуков.
Инженер по тестированию	Сотрудник, который занимается тестированием собранных компьютеров (функциональное, нагрузочное, на удобство использования); автоматизацией тестирования, анализом функциональных требований к продуктам, составление плана тестирования.
Завскладом	Сотрудник, руководит работой склада по приему, хранению и отпуску товарно-материальных ценностей, составляет установленную отчетность, обеспечивает сохранность складироваемых товарно-материальных

ценностей.
------------

Возможности системы для действующих лиц/актеров приведены в таблице 2.

Таблица 2. Возможности системы для действующих лиц/актеров

Актер Директор	Использует систему для руководства и координации процессов закупок, поставок, продаж, принятия стратегически важных решений в области сбыта, ценообразования, маркетинговой, ассортиментной политики.
Актер Бухгалтер	Использует систему для организации бухгалтерского учета хозяйственно–финансовой деятельности и контроля за экономным использованием материальных, трудовых и финансовых ресурсов, сохранностью собственности предприятия.
Актер Менеджер по работе с клиентами	Использует систему для оформления, редактирования, комплектацией заказов, контроля за расчетом за укомплектованный заказ, управления информацией о клиентах предприятия.
Актер Менеджер по снабжению	Использует систему для управления оперативной маркетинговой информацией и рекламных материалов, договорной документации с поставщиками, обеспечения доставки материальных ресурсов от поставщиков и продавцов, организации обработки поставок комплектующих материально–технических ресурсов.
Актер Инженер по сборке настольных компьютеров	Использует систему для просмотра нарядов на сборку персональных компьютеров, для заказа комплектующих со склада и отметки о ходе выполнения работы.
Актер Инженер по сборке ноутбуков	Использует систему для просмотра нарядов на сборку ноутбуков, для заказа комплектующих со склада и отметки о ходе выполнения работы.
Актер Инженер по тестированию	Использует систему для просмотра нарядов на тестирование собранной продукции и отметки о ходе выполнения работы, тестирования собранных компьютеров, автоматизации тестирования.
Актер Завскладом	Использует систему для приема, хранения, отпуска товарно-материальных ценностей – комплектующих, составления установленной отчетности, обеспечения сохранности складированных товарно-материальных ценностей.

Прецеденты/варианты использования представлены в таблице 3.

Таблица 3. Прецеденты/варианты использования

Прецедент/вариант использования	Краткое описание
Управление системой	Иницируется директором. Позволяет руководить и корректировать производственными процессами.
Учет финансово–хозяйственной деятельности	Иницируется бухгалтером. Позволяет вести учет хозяйственно–финансовой деятельности, контроль за использованием материальных, трудовых и финансовых ресурсов, сохранностью собственности предприятия.
Работа с заказом	Иницируется менеджером по работе с клиентами. Позволяет формировать, корректировать, удалять, просматривать заказ клиента.
Управление информацией о клиенте	Запускается менеджером по работе с клиентами. Позволяет добавлять, изменять или удалять клиентов, а также просматривать информацию о клиентах.
Управление информацией о поставщиках	Запускается менеджером по снабжению. Позволяет добавлять, изменять или удалять поставщиков, а также просматривать информацию о поставщиках.

Управление информацией о комплектующих изделиях и запасных частях	Запускается менеджером по снабжению. Позволяет просматривать информацию о комплектующих изделиях, производить анализ их расходования, прогнозировать необходимое их количество и делать заказ
Сборка компьютеров	Инициализируется инженером по сборке. Позволяет просматривать наряды на сборку компьютеров, делать заказ на сборку комплектующих изделий, делать отметки о ходе выполнения работы.
Пополнение комплектующими изделиями и запасными частями	Иницируется инженером по сборке компьютеров. Предназначено для затребования необходимых комплектующих изделий и запасных частей со склада.
Тестирование компьютеров	Иницируется инженером по тестированию. Позволяет просмотреть список компьютеров, подлежащих тестированию, провести тестирование, сделать отметки о ходе выполнения работ.
Учет поступления и выдачи комплектующих изделий и запасных частей	Иницируется заведующим складом. Позволяет вести учет поступления и выдачи запчастей и комплектующих изделий.

## Лабораторная работа 2 «Построение диаграммы Кооперации и диаграммы Развертывания и генерация кода»

**Цель работы:** приобретение навыков построения диаграммы Кооперации и диаграммы Развертывания и генерация кода.

Задание 1:

Построить диаграмму кооперации для моделирования процесса телефонного разговора с использованием обычной телефонной сети. Объектами в этом примере являются два абонента а и b, два телефонных аппарата, коммутатор и сам разговор как объект моделирования. Абоненты в данном случае рассматриваются как актеры, первый из них играет активную роль, а второй – пассивную. Поэтому первый получает фокус управления сразу после своего появления в системе, а второй имеет только линию жизни. Разговор как объект появляется только после установки соединения и уничтожается с его прекращением. Потому они изображаются позже на диаграмме последовательности.

Специфицируйте все связи на этой диаграмме, указав на их концах необходимую информацию. На диаграмму кооперации нанесите сообщения, указав их семантические особенности.

Вопросы для проверки знаний

- Каково назначение диаграмм кооперации (ДК)? Почему они так называются?
- Какие элементы располагаются на ДКп?
- С какими другими диаграммами в UML-модели связана ДКп? Какие элементы из этих диаграмм также представлены на ДКп?
- Каким образом связываются между собой элементы на ДКп?
- Какую роль на диаграммах кооперации играют отношения между объектами?
- Какова роль диаграмм развертывания в проектировании информационных систем?
- Какие группы специалистов участвуют в разработке ДР?
- Диаграмма развертывания (ДР) является единой для системы в целом или может описывать какую-то ее часть?
- В каких случаях использование диаграммы развертывания нецелесообразно?
- Что собой в языке UML представляет узел?
- Какие виды узлов используются в UML?

- В чем разница между элементами «Узел» и «Экземпляр узла»?
- Для чего на ДР могут быть представлены компоненты? Какие виды компонентов для этого используются?
- Какую роль на ДР играют интерфейсы?
- Для чего на ДР используются сообщения и зависимости?

## Задание 2. Диаграмма развертывания.

Краткие теоретические сведения. Диаграмма развертывания – это тип диаграммы, которая определяет физическое оборудование, на котором будет работать программная система. Она также определяет способ развертывания программного обеспечения на базовом оборудовании. Она отображает программные части системы на устройство, которое будет выполнять его.

Диаграмма развертывания отображает архитектуру программного обеспечения, созданную в проекте, в архитектуру физической системы, которая ее выполняет. В распределенных системах она моделирует распределение программного обеспечения по физическим узлам.

Существует две формы схемы развертывания.

- Форма дескриптора: содержит узлы, отношения между узлами и артефактами.
- Форма экземпляра: содержит экземпляр узла, связь между экземплярами узла и экземпляром артефакта.

Диаграмма развертывания состоит из следующих обозначений:

- Узел.
- Компонент.
- Артефакт.
- Интерфейс.

Артефакт представляет собой спецификацию конкретной сущности реального мира, связанной с разработкой программного обеспечения. Вы можете использовать артефакт, чтобы описать структуру, которая используется в процессе разработки программного обеспечения или исполняемый файл. Артефакты развертываются на узлах. Наиболее распространенными артефактами являются следующие,

- Исходные файлы.
- Исполняемые файлы.
- Таблицы базы данных.
- Сценарии.
- DLL файлы.
- Руководства пользователя или документация.
- Выходные файлы.

Артефакты развертываются на узлах. Он может обеспечить физическое проявление для любого элемента UML. Как правило, они проявляют компоненты. Артефакты помечены стереотипом << артефакт >> и могут иметь значок артефакта в верхнем правом углу.

У каждого артефакта есть имя файла в его спецификации, которое указывает физическое местоположение артефакта. Артефакт может содержать другой артефакт. Это может зависеть друг от друга.

Артефакты имеют свои свойства и поведение, которое ими манипулирует.

Экземпляр артефакта представляет собой экземпляр конкретного артефакта. Экземпляр артефакта обозначается тем же символом, что и у артефакта, за исключением того, что имя подчеркнуто. Диаграмма UML позволяет различать исходный артефакт и экземпляр. Каждая физическая копия или файл является экземпляром уникального артефакта.

Узел - это вычислительный ресурс, на котором развертываются артефакты для выполнения. Узел — это физическая вещь, которая может выполнять один или несколько артефактов. Узел может различаться по размеру в зависимости от размера проекта.

Узел является важным элементом UML, который описывает выполнение кода и взаимодействие между различными объектами системы. Он обозначается трехмерной рамкой с именем узла, написанным внутри него. Узлы помогают передавать оборудование, которое используется для развертывания программного обеспечения.

Ассоциация между узлами представляет собой канал связи, по которому осуществляется обмен информацией в любом направлении.

Диаграмма развертывания визуализирует топологическое представление всей системы. Он представляет собой развертывание системы.

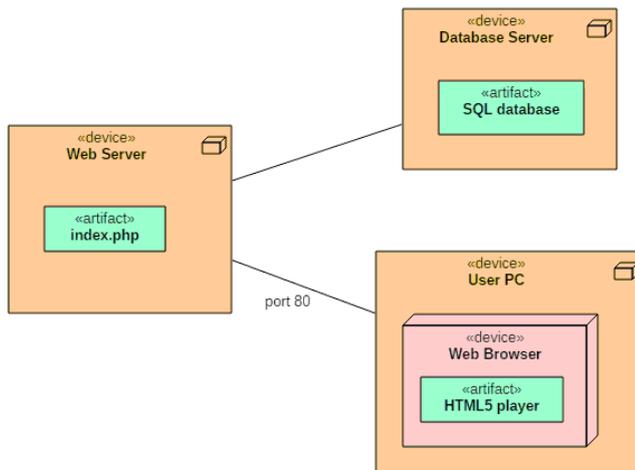
Диаграммы развертывания в основном используются системными администраторами, сетевыми инженерами и т. Д. Эти диаграммы используются с единственной целью описания того, как программное обеспечение внедряется в аппаратную систему. Он визуализирует, как программное обеспечение взаимодействует с аппаратным обеспечением для выполнения полной функциональности.

Чтобы программное обеспечение работало эффективно и быстрее, оборудование также должно быть хорошего качества. Он должен быть спроектирован эффективно, чтобы программное обеспечение работало правильно и давало точные результаты в короткие сроки.

Диаграммы развертывания могут быть использованы для,

- Моделирование топологии сети системы.
- Моделирование распределенных систем и сетей.
- Прямые и обратные инженерные процессы.

Постройте и объясните схему диаграммы развертывания, представляющую работу проигрывателя HTML5 в браузере:



Вопросы для проверки знаний

1. Какова роль диаграмм развертывания в проектировании информационных систем?
2. Какие группы специалистов участвуют в разработке ДР?
3. Диаграмма развертывания является единой для системы в целом или может описывать какую-то ее часть?
4. В каких случаях использование диаграммы развертывания нецелесообразно?
5. Что собой в языке UML представляет узел?
6. Какие виды узлов используются в UML?
7. В чем разница между элементами «Узел» и «Экземпляр узла»?
8. Для чего на ДР могут быть представлены компоненты? Какие виды компонентов для этого используются?
9. Какую роль на ДР играют интерфейсы?
10. Для чего на ДР используются сообщения и зависимости?

### **Лабораторная работа 3 «Построение диаграммы Деятельности, диаграммы Состояний и диаграммы Классов и генерация кода»**

**Цель работы:** освоить построение диаграмм деятельности, диаграмм состояний и диаграмм классов.

**Задание:** Задана предметная область. Для заданной предметной области построить указанные диаграммы.

Предметная область: туристическое агентство. Клиент может выбрать тур на веб-сайте агентства. Клиент является потенциальным покупателем туристического продукта, взаимодействует с информационной системой через интернет. Турагент реализует клиенту сформированный туроператором тур на тех условиях, которые предлагаются туроператором. Туроператор осуществляет деятельность по формированию, продвижению и реализации туристического продукта. Формирование туристического продукта складывается из бронирования и оплаты отеля, заказа авиабилета, обеспечения услуг по предоставлению транспорта, экскурсионных услуг и т.д. Кроме того, туроператор определяет цены на сформированный им тур и политику скидок.

Информационная система предоставляет каталог всех туров. Каталог содержит полную информацию о туре (страна, дата вылета и прилета, количество дней пребывания, стоимость). Клиент может забронировать только тот тур, который присутствует в каталоге, а также оставить пожелания на сайте. Туроператор имеет возможность добавить новый тур в каталог. Турагент оформляет все необходимые документы с клиентом (составляет договор).

Вопросы для контроля знаний

1. Методология объектно-ориентированного анализа и проектирования информационных систем.
2. Сущность объектно-ориентированного подхода к анализу и проектированию ИС. Варианты использования.
3. Описание варианта использования. Виды сценариев вариантов использования.
4. Диаграммы вариантов использования, их элементы и связи между элементами.
5. Диаграммы взаимодействия UML, их элементы и связи между элементами.
6. Виды диаграмм взаимодействия и область их применения. Диаграммы классов UML, их элементы и связи между элементами. Область применения.
7. Диаграммы состояний, их элементы и связи между элементами. Область применения.
8. Диаграммы деятельности, их элементы и связи между элементами. Область применения.
9. Диаграммы компонентов и диаграммы размещения, их элементы и связи между элементами. Область применения.

### **Лабораторная работа 4 «Построение диаграммы компонентов и генерация кода»**

**Цель работы:** освоить построение диаграмм компонентов.

**Задание:** Задана предметная область (по вариантам). Для заданной предметной области постройте диаграмму компонентов.

*Варианты индивидуальных заданий:*

#### 1. Розничный магазин

Розничный магазин занимается продажей продуктов. Основные процессы, на которых основывается деятельность магазина: поступление товаров, возврат товаров поставщику, реализация товаров, инвентаризация. Каждая партия товаров сопровождается накладной, счет-фактурой и сертификатом качества. Оператор сверяет количество товара с документами, принимает и отправляет товары на склад. Администратор рассчитывает розничные цены для поступившего товара, а также формирует заявку на товар, который необходимо вынести в торговый зал. В случае несоответствия товара требованиям магазина товаровед принимает решение о его возврате по возвратной накладной. В конце дня старший кассир закрывает смену на каждой кассе, формирует отчеты. В процессе инвентаризации осуществляется сверка остатков по базе данных с реальными остатками на складе и в магазине.

## 2. Гостиница

Основные процессы, на которых основывается деятельность гостиницы: бронирование мест, прием, регистрация и размещение гостей, предоставление услуг проживания и питания, предоставление дополнительных услуг проживающим, окончательный расчет и оформление выезда. При регистрации и оформлении выезда работники гостиницы осуществляют расчет за проживание в гостинице. При выписке проверяется счет гостя, уточняются все его расходы за время проживания, принимается оплата. Дежурный администратор, начиная работу, должен просмотреть журнал с записями предыдущей смены. Перед началом работы необходимо также просмотреть информацию о наличии свободных мест и заявки на текущие сутки. Дежурный администратор контролирует своевременность оплаты услуг, получает плату за проживание при наличном расчете и составляет кассовые отчеты для бухгалтерии. Необходимо спроектировать бизнес-процессы заданной предметной области.

## 3. Салон красоты

Основные процессы, на которых основывается деятельность салона: оказание услуг в сфере красоты, продажа и заказ косметических средств, финансовый учет. Администратор непосредственно работает с клиентами (принимает заявки, анализирует отзывы и пожелания), составляет график смен. График смен формируется вручную без учета пожеланий обслуживающего персонала. Закупка средств осуществляется менеджером по закупкам у выбранных поставщиков, предварительно составляется список необходимых позиций. В случае если позиция не соответствует требованиям салона, менеджер по закупкам может вернуть товар, если такое условие предусмотрено договором с поставщиком. Запись клиентов осуществляется вручную, что вызывает определенные неудобства (могут возникать ошибки, дублирование записей). Необходимо спроектировать бизнес-процессы заданной предметной области.

## 4. Моделирование бизнес-процессов стоматологической клиники

Клиника оказывает медицинские услуги: лечение и протезирование зубов. Клиент подает заявку на посещение стоматолога в регистратуру. Поступившая заявка записывается в журнал. Журнал ведется в бумажном виде. Во время оформления заявки с клиентом оговариваются условия дальнейшего обследования, время приема и стоимость услуг. После того как условия согласованы, данные клиента заносятся в базу данных и заключается договор. Во время посещения клиенту оформляется медицинская карта, в которую записываются личные данные. В этой карте фиксируются все дальнейшие приемы. После того как клиенту оказаны услуги, лечащий врач заносит информацию об оказанных услугах в медицинскую карту и выдает ее клиенту. На основании записи в медицинской карте бухгалтер в соответствии с прайслистом выписывает квитанцию, по которой клиент должен будет оплатить услуги в кассе. Необходимо спроектировать бизнес-процессы заданной предметной области.

## 5. Моделирование бизнес-процессов транспортно-логистической компании

Основными задачами транспортно-логистической компании являются разработка оптимальной транспортно-технологической схемы, организация перевозки. Разработкой транспортно-технологической схемы перевозки занимается логистический отдел. Этот процесс включает в себя выбор типа транспортного средства, вида транспортировки, маршрута. Сотрудники компании осуществляют прием и обработку заявок на перевозку, заключение договоров с клиентами, проведение необходимых бухгалтерских операций, экспедирование перевозки, таможенное оформление. После обращения клиента заключается договор, оформляется заявка на перевозку, составляется транспортно-технологическая схема. До начала транспортировки груза необходимо произвести расчет за перевозку с компанией-перевозчиком. Затем происходит транспортировка груза, оказываются экспедиционные услуги.

## **Лабораторная работа 5 «Построение диаграмм потоков данных и генерация кода»**

**Цель работы:** получение навыков построения диаграмм потоков данных и генерация кода.

### *Краткие теоретические сведения*

Диаграммы потоков данных (Data Flow Diagrams – DFD) используются для описания движения документов и обработки информации как дополнение к IDEF0. В отличие от IDEF0, где система рассматривается как взаимосвязанные работы, стрелки в DFD показывают лишь то, как объекты (включая данные) движутся от одной работы к другой. DFD отражает функциональные зависимости значений, вычисляемых в системе, включая входные значения, выходные значения и внутренние хранилища данных. DFD – это граф, на котором показано движение значений данных от их источников через преобразующие их процессы к их потребителям в других объектах.

DFD содержит процессы, которые преобразуют данные, потоки данных, которые переносят данные, активные объекты, которые производят и потребляют данные, и хранилища данных, которые пассивно хранят данные.

Диаграмма потоков данных содержит:

- процессы, которые преобразуют данные;
- потоки данных, переносящие данные;
- активные объекты, которые производят и потребляют данные; - хранилища данных, которые пассивно хранят данные.

Процесс DFD преобразует значения данных и изображается в виде эллипса, внутри которого помещается имя процесса. Поток данных соединяет выход объекта (или процесса) с входом другого объекта (или процесса) и представляет собой промежуточные данные вычислений. Поток данных изображается в виде стрелки между производителем и потребителем данных, помеченной именами соответствующих данных. Дуги могут разветвляться или сливаться, что означает соответственно разделение потока данных на части либо слияние объектов.

Активным объектом является объект, который обеспечивает движение данных, поставляя или потребляя их. Хранилище данных – это пассивный объект в составе DFD, в котором данные сохраняются для последующего доступа. Хранилище данных – это пассивный объект в составе DFD, в котором данные сохраняются для последующего доступа. Хранилище данных допускает доступ к хранимым в нем данным в порядке, отличном от того, в котором они были туда помещены. Агрегатные хранилища данных, как, например, списки и таблицы, обеспечивают доступ к данным в порядке их поступления, либо по ключам.

Первым шагом при построении иерархии DFD является построение контекстных диаграмм. Обычно при проектировании относительно простых информационных систем строится единственная контекстная диаграмма со звездообразной топологией, в центре которой находится так называемый главный процесс, соединенный с приемниками и источниками информации, посредством которых с системой взаимодействуют пользователи и другие внешние системы.

Если же для сложной системы ограничиться единственной контекстной диаграммой, то она будет содержать слишком большое количество источников и приемников информации, которые трудно расположить на листе бумаги нормального формата, и, кроме того, главный единственный процесс не раскрывает структуры распределенной системы.

Для сложных информационных систем строится иерархия контекстных диаграмм. При этом контекстная диаграмма верхнего уровня содержит не главный единственный процесс, а набор подсистем, соединенных потоками данных. Контекстные диаграммы следующего уровня детализируют контекст и структуру подсистем.

При построении иерархии DFD переходить к детализации процессов следует только после определения содержания всех потоков и накопителей данных, которое описывается при помощи структур данных. Структуры данных конструируются из элементов данных и могут содержать альтернативы, условные вхождения и итерации. Условное вхождение означает, что данный компонент может отсутствовать в структуре. Альтернатива означает, что в структуру может входить один из перечисленных элементов. Итерация означает

вхождение любого числа элементов в указанном диапазоне. Для каждого элемента данных может указываться его тип (непрерывные или дискретные данные). Для непрерывных данных может указываться единица измерения (кг, см и т.п.), диапазон значений, точность представления и форма физического кодирования. Для дискретных данных может указываться таблица допустимых значений.

**Задание.** Постройте диаграмму потоков данных для выбранной информационной системы.  
*Варианты индивидуальных заданий:*

Варианты представлены в предыдущей работе.

### **Тема 5.2.2. Основные приемы разработки прикладных решений информационной системы**

#### **Практическая работа 1 «Создание информационной базы на платформе 1С: Предприятие 8.3.»**

**Цель:** Получить практические навыки создания информационной базы.

**Задание:** С помощью платформы 1С Предприятие 8.3 создать информационную базу «Личные доходы», которая включает в себя следующие объекты:

- a. 2 подсистемы
- b. 3 справочника
- c. 2 документа

*Методические указания*

1. Название объектов пишутся без пробелов
2. Для корректного запуска каталог базы должен располагаться на диске D

### **Тема 5.2.3. Разработка и модификация информационных систем**

#### **Практическая работа 2 «Обоснование выбора технических средств»**

**Цель:** приобрести навыки выбора и обоснования технических средств для определенной предметной области.

**Задание 1.** Подберите комплекс технических средств: средства компьютерной техники, коммуникационной техники, организационной техники, оперативной полиграфии, необходимых для функционирования информационной системы (по вариантам).

Уделяйте внимание характеру задач, которые должна решать информационная система. Количество единиц техники выбирайте в зависимости от величины предполагаемого предприятия.

**Задание 2.** Опишите выбранные технические средства, заполнив таблицу:

№	Товар (техническое средство)	Конфигурация	Количество	Единица измерения	Цена	Сумма
...	...		...	...	...	...

**Задание 3** Обоснуйте выбор конфигурации технических средств в зависимости от назначения.

*Варианты информационных систем (ИС)*

1. ИС «Центр бытового обслуживания».
2. ИС «Библиотека».
3. ИС «Издательство».
4. ИС «Поликлиника».
5. ИС «Учебный центр».
6. ИС «Ателье».
7. ИС «Склад».
8. ИС «Автосалон».
9. ИС «Автосервис».
10. ИС «Туристическое агентство».
11. ИС «Гостиница».

### Практическая работа 3 «Стоимостная оценка проекта»

**Цель:** приобрести навыки расчета затрат на разработку проекта.

*Краткие теоретические сведения:*

Капитальные вложения, связанные с автоматизацией обработки информации, рассчитываются по формуле

$$K = K_{\Pi} + K_p,$$

где  $K_{\Pi}$  - капитальные вложения на проектирование, руб.;

$K_p$  - капитальные вложения на реализацию проекта, руб.

Предпроизводственные затраты представляют собой единовременные расходы на разработку обеспечивающих или функциональных систем или элементов на всех этапах проектирования, а также затраты на их усовершенствование, т.е. на проведение обследования и обработку материалов исследования, разработку технического задания, разработку технического и рабочего проекта системы и ее опытного внедрения. Сюда включаются затраты на разработку алгоритмов и программ, стоимость разработок по привязке типовых проектных решений (ТПР) и пакетов прикладных программ (ППП) к конкретному объекту автоматизации.

Суммарные затраты на проектирование системы, ее разработку и отладку на компьютере определяются по формуле:

$$K_{\Pi} = ((1 + W_d)(1 + W_c) + W_n) \sum_{i=1}^m Z_{oi} + C_M + M_e,$$

где  $m$  – количество работников, участвующих в разработке проекта;

$Z_{oi}$  – затраты на основную заработную плату работника  $i$ -й категории, руб.;

$W_d$  – коэффициент, учитывающий дополнительную заработную плату в долях к основной заработной плате;

$W_c$  – коэффициент, учитывающий отчисления на социальные нужды, в долях к сумме основной и дополнительной заработной платы разработчиков;

$W_n$  – коэффициент, учитывающий накладные расходы организации, в долях к основной заработной плате разработчиков (*принимается по фактическим данным,  $W_n = 0,6$* );

$C_M$  – затраты на материалы;

$M_e$  – затраты на использование машинного времени.

Затраты на основную заработную плату работника  $i$ -й категории:

$$Z_{oi} = Z_{\text{дни}} t_i,$$

где  $Z_{\text{дни}}$  – среднедневная заработная плата работника  $i$ -й категории, руб./дн.;  $t_i$  – количество дней, отработанных работником  $i$ -й категории.

Затраты времени на разработку системы по каждому исполнителю принимаются, исходя из его загрузки по календарному графику выполнения работ.

Ввиду того, что проектируемая информационная система должна быть запрограммирована и отлажена с помощью компьютеров, к суммарным затратам на разработку добавляются затраты на использование машинного времени, исчисляемые как

$$M_e = t_{\text{мв}} S_{\text{мч}} K_{\text{м}},$$

где  $t_{\text{мв}}$  – машинное время компьютера, необходимое для разработки программного продукта;  $K_{\text{м}}$  – коэффициент мультипрограммности (показывает долю машинного времени, отводимого непосредственно на работу над проектом).

Капитальные вложения на реализацию проекта вычисляются по формуле:

$$K_p = K_o + K_{\text{зд}} + K_{\text{пп}} + K_{\text{св}} + K_{\text{иб}} + K_{\text{пк}},$$

где  $K_o$  – затраты на основное и вспомогательное оборудование, руб.;  $K_{\text{зд}}$  – затраты на строительство, реконструкцию здания и помещений, руб.;  $K_{\text{пп}}$  – затраты на приобретение типовых разработок, пакетов, руб.;  $K_{\text{св}}$  – затраты на прокладку линий связи, руб.;  $K_{\text{иб}}$  –

затраты на создание информационной базы, руб.; Кпк – затраты на подготовку и переподготовку кадров, руб.

**Задание:** провести расчет и обосновать затраты на разработку проекта (по вариантам).

- Обучающие информационные системы.
- Автоматизированное рабочее администратора / менеджера (медицинского центра, спортивного клуба, детского развивающего центра, сервисного центра, кафе и т.д.).
- Интернет-магазин (по категориям товаров).
- Web-приложение для учебного центра.
- Web-приложение для торговой организации.
- Web-приложение для сервисного центра.
- Web-приложение для медицинского центра.
- Web-приложение для туристического агентства.

#### **Практическая работа 4 «Построение и обоснование модели проекта»**

**Цель:** получить навыки построения моделей проекта.

**Задание:** постройте и обоснуйте модель проекта, связанного с информатизацией процессов.

*Варианты индивидуальных заданий:*

##### 1. Моделирование бизнес-процессов розничного магазина

Розничный магазин занимается продажей продуктов. Основные процессы, на которых основывается деятельность магазина: поступление товаров, возврат товаров поставщику, реализация товаров, инвентаризация. Каждая партия товаров сопровождается накладной, счет-фактурой и сертификатом качества. Оператор сверяет количество товара с документами, принимает и отправляет товары на склад. Администратор рассчитывает розничные цены для поступившего товара, а также формирует заявку на товар, который необходимо вынести в торговый зал. В случае несоответствия товара требованиям магазина товаровед принимает решение о его возврате по возвратной накладной. В конце дня старший кассир закрывает смену на каждой кассе, формирует отчеты. В процессе инвентаризации осуществляется сверка остатков по базе данных с реальными остатками на складе и в магазине.

Необходимо спроектировать бизнес-процессы заданной предметной области.

##### 2. Моделирование бизнес-процессов гостиницы

Основные процессы, на которых основывается деятельность гостиницы: бронирование мест, прием, регистрация и размещение гостей, предоставление услуг проживания и питания, предоставление дополнительных услуг проживающим, окончательный расчет и оформление выезда. При регистрации и оформлении выезда работники гостиницы осуществляют расчет за проживание в гостинице. При выписке проверяется счет гостя, уточняются все его расходы за время проживания, принимается оплата. Дежурный администратор, начиная работу, должен просмотреть журнал с записями предыдущей смены. Перед началом работы необходимо также просмотреть информацию о наличии свободных мест и заявки на текущие сутки. Дежурный администратор контролирует своевременность оплаты услуг, получает плату за проживание при наличном расчете и составляет кассовые отчеты для бухгалтерии. Необходимо спроектировать бизнес-процессы заданной предметной области.

##### 3. Моделирование бизнес-процессов салона красоты

Основные процессы, на которых основывается деятельность салона: оказание услуг в сфере красоты, продажа и заказ косметических средств, финансовый учет. Администратор непосредственно работает с клиентами (принимает заявки, анализирует отзывы и пожелания), составляет график смен. График смен формируется вручную без учета пожеланий обслуживающего персонала. Закупка средств осуществляется менеджером по закупкам у выбранных поставщиков, предварительно составляется список необходимых позиций. В случае если позиция не соответствует требованиям салона, менеджер по закупкам может вернуть товар, если такое условие предусмотрено договором с

поставщиком. Запись клиентов осуществляется вручную, что вызывает определенные неудобства (могут возникать ошибки, дублирование записей). Необходимо спроектировать бизнес-процессы заданной предметной области.

#### 4. Моделирование бизнес-процессов стоматологической клиники

Клиника оказывает медицинские услуги: лечение и протезирование зубов. Клиент подает заявку на посещение стоматолога в регистратуру. Поступившая заявка записывается в журнал. Журнал ведется в бумажном виде. Во время оформления заявки с клиентом оговариваются условия дальнейшего обследования, время приема и стоимость услуг. После того как условия согласованы, данные клиента заносятся в базу данных и заключается договор. Во время посещения клиенту оформляется медицинская карта, в которую записываются личные данные. В этой карте фиксируются все дальнейшие приемы. После того как клиенту оказаны услуги, лечащий врач заносит информацию об оказанных услугах в медицинскую карту и выдает ее клиенту. На основании записи в медицинской карте бухгалтер в соответствии с прайслистом выписывает квитанцию, по которой клиент должен будет оплатить услуги в кассе. Необходимо спроектировать бизнес-процессы заданной предметной области.

#### 5. Моделирование бизнес-процессов транспортно-логистической компании

Основными задачами транспортно-логистической компании являются разработка оптимальной транспортно-технологической схемы, организация перевозки. Разработкой транспортно-технологической схемы перевозки занимается логистический отдел. Этот процесс включает в себя выбор типа транспортного средства, вида транспортировки, маршрута. Сотрудники компании осуществляют прием и обработку заявок на перевозку, заключение договоров с клиентами, проведение необходимых бухгалтерских операций, экспедирование перевозки, таможенное оформление. После обращения клиента заключается договор, оформляется заявка на перевозку, составляется транспортно-технологическая схема. До начала транспортировки груза необходимо произвести расчет за перевозку с компанией-перевозчиком. Затем происходит транспортировка груза, оказываются экспедиционные услуги.

Необходимо спроектировать бизнес-процессы заданной предметной области. 6. Моделирование бизнес-процессов рекламного агентства

Основными процессами в рекламном агентстве являются: рассмотрение заявок, обработка заказов, подготовка к выпуску и выпуск рекламной продукции. Рекламное агентство в своей работе использует систему антиплагиата и руководствуется текущим законодательством. Агентство занимается изготовлением щитов, баннеров, рекламных буклетов и продвижением в социальных сетях. Сроки и стоимость заказа согласовываются на этапе заключения договора, но могут меняться в процессе выполнения заказа. В случае изменения условий составляется дополнительное соглашение к договору. Продукция проходит контроль качества. Необходимо спроектировать бизнес-процессы заданной предметной области.

### **Лабораторная работа 6 «Установка и настройка системы контроля версий с разграничением ролей»**

**Цель:** получить навыки установка и настройка системы контроля версий с разграничением ролей

*Краткие теоретические сведения*

Git - это система контроля версий (VCS), которая позволяет отслеживать и фиксировать изменения в коде: вы можете восстановить код в случае сбоя или откатить до более ранних версий.

#### **Задание 1.**

- Создайте пустой репозиторий в роли master на сервисе github.com (или gitlab.com, или bitbucket.com).

- Выполните инструкцию по созданию первого коммита в роли мастера репозитория на странице репозитория, созданного на предыдущем шаге.
- Создайте файл `hello_world.cpp` в локальной копии репозитория (который должен был появиться на шаге 2). Реализуйте программу Hello world на языке C#.
- Добавьте этот файл в локальную копию репозитория.
- Закоммитьте изменения с *осмысленным* сообщением.
- Измените исходный код так, чтобы программа через стандартный поток ввода запрашивалось имя пользователя. А в стандартный поток вывода печаталось сообщение Hello world from @name, где @name имя пользователя.
- Закоммитьте новую версию программы. Почему не надо добавлять файл повторно `git add`?
- Запустите изменения в удалённый репозиторий.
- Проверьте, что история коммитов доступна в удалённый репозитории.

**Задание 2.** Работаем с теми же репозиториями, что и в первой части задания.

- В локальной копии репозитория создайте локальную ветку `patch1`.
- Внесите изменения в ветке `patch1` по исправлению кода и избавления от `using namespace std;`.
- `commit, push` локальную ветку в удалённый репозиторий.
- Проверьте, что ветка `patch1` доступна в удалённый репозитории.
- Создайте `pull-request patch1 -> master`.
- В локальной копии в ветке `patch1` добавьте в исходный код комментарии.
- `commit, push`.
- Проверьте, что новые изменения есть в созданном на шаге 5 `pull-request`
- В удалённый репозитории выполните слияние `PR patch1 -> master` и удалите ветку `patch1` в удалённом репозитории.
- Локально выполните `pull`.
- С помощью команды `git log` просмотрите историю в локальной версии ветки `master`.
- Удалите локальную ветку `patch1`.

**Задание 3.** Работаем с теми же репозиториями, что и в первой части задания.

- Создайте новую локальную ветку `patch2`.
- Измените *code style* с помощью утилиты `clang-format`. Например, используя опцию `-style=Mozilla`.
- `commit, push`, создайте `pull-request patch2 -> master`.
- В ветке `master` в удалённом репозитории измените комментарии, например, расставьте знаки препинания, переведите комментарии на другой язык.
- Убедитесь, что в `pull-request` появились *конфликтны*.
- Для этого локально выполните `pull + rebase` (точную последовательность команд, следует узнать самостоятельно). Исправьте конфликты.
- Сделайте *force push* в ветку `patch2`
- Убедитесь, что в `pull-request` пропали конфликтны.
- Вмержите `pull-request patch2 -> master`.

Вопросы для проверки знаний

1. Каждый `commit` содержит в себе:

- Путь к папке с проектом.
- Резервную копию проекта.
- Ссылку на удалённый репозиторий.
- Информацию о внесенных изменениях.

2. После выполнения `commit` информация об изменениях содержится в:

- Локальном репозитории.
- Удалённом репозитории.
- Обоих репозиториях.

3. Команда revert откатывает ветку к исходному состоянию путем:

- Удаления последнего commit'a в локальном репозитории.
- Удаления последнего commit'a в удаленном репозитории.
- Создания нового commit'a в локальном репозитории.
- Создания нового commit'a в удаленном репозитории.

### **Лабораторная работа 7 «Проектирование и разработка интерфейса пользователя»**

**Цель:** овладеть навыками проектирования и разработки интерфейса пользователя.

*Алгоритм выполнения работы:*

- Проведите анализ предметной области (по вариантам): описание основного и других сценариев.
- Разработайте диаграммы вариантов использования, диаграммы последовательностей, диаграммы классов, диаграммы взаимодействий.
- При разработке диаграмм уделите внимание различным этапам их разработки:
- описание списков кандидатов на роль прецедентов или концептуальных классов;
- определение ассоциаций;
- определение атрибутов, необходимых для выполнения информационных требований.
- Разработайте класс интерфейса пользователя. Данный класс должен получиться путем преобразования разработанных ранее диаграмм.
- Разработайте формы интерфейса пользователя.
- Проведите GOM оценку интерфейса пользователя для основного сценария.
- Модернизируйте интерфейс для улучшения GOM оценки.
- Разработайте метрики предпочтений для проектируемого интерфейса.
- На основе метрик предпочтений составьте анкету для опроса потенциальных пользователей и проведите этот опрос среди однокурсников.
- Разработайте структурные метрики для оценки спроектированного интерфейса.
- На основе структурных метрик составьте анкету и оцените спроектированный интерфейс.

*Варианты предметных областей*

- Интернет-магазин. Должны быть реализованы сценарии: покупка товара, поиск товара, добавление нового товара в базу данных магазина, просмотр и обработка заказов покупателей, регистрация нового покупателя.
- Книжный каталог. Должны быть реализованы сценарии:
- добавления новой книги, поиск книги по нескольким полям, бронирование книги, списание старых книг, регистрация пользователей каталога.
- Адресная книга. Должны быть реализованы сценарии: добавление нового абонента, добавление категории абонентов, поиск абонентов по нескольким полям, добавления администраторе каталога (пользователей, которые имеют право редактировать данные адресной книги), редактирование данных абонента.
- Расписание занятий. Должны быть реализованы сценарии: добавление новой группы, добавление занятия (с указанием названия предмета, времени, аудитории, группы, недели, преподавателя, типа занятия), просмотр списка занятия на выбранную дату, добавление списка преподавателей, поиск занятия по нескольким полям (предмету, преподавателя, группе, времени, типу занятия).
- База студентов. Должны быть реализованы сценарии: добавление новой группы, добавление нового студента, поиск студента по различным полям, добавления информации об оценках по различным предметам, отчисление студента.
- Прайс-лист фирмы. Должны быть реализованы сценарии: добавление новой категории товаров, добавление нового товара, поиск товара по различным полям,

добавление администратора прайс-листа (пользователей, которые имеют право редактировать прайс-лист), перемещение товара из одной категории в другую.

- База склада фирмы. Должны быть реализованы сценарии:
- добавление нового товара на склад, списание товара, выдача товара, поиск товара по различным полям, изменение месторасположения товара на складе.
- Аптечная база. Должны быть реализованы сценарии: прием заказа от клиента на изготовление раствора, продажа лекарства, списание просроченных лекарств, добавление новые лекарств в базу данных, поиск заказов по различным полям.

## Лабораторная работа 8 «Разработка графического интерфейса пользователя»

### Цель:

- изучить принципы разработки графического интерфейса приложений для ОС Windows в Visual Studio .Net;
- освоить использование элементов графического интерфейса для управления работой приложения.
- освоить принципы построения иерархических меню, создания диалоговых окон;
- изучить модель обработки событий в языке C#.

### Задание:

- Создать учебные примеры и разобрать принцип их работы. Поместите в отчет примеры работы программ, их код с комментариями.
- Создать SDI-приложение (Single Document Interface, однодокументный интерфейс) с элементами ввода и отображения полей класса. Для этого используйте различные элементы управления: текстовые поля, списки, независимые и радиокнопки, а также панели и менеджеры компоновки.
- Ввод новых данных осуществлять через дополнительную диалоговую форму.
- При изменении данных запрашивать подтверждение через окно диалога. В случае неполных данных сообщать об ошибке.
- Объекты сохранять в коллекции.
- Реализовать просмотр всей коллекции объектов через список. Для редактирования выбранного объекта создать дополнительную форму модального диалога.
- Добавить на форму меню, позволяющее работать с пунктами: добавить, просмотреть, удалить, редактировать, справка.
- Дублировать основные операции панелью инструментов.

### Краткие теоретические сведения

Средством взаимодействия пользователя с программой является графический пользовательский интерфейс (Graphical User Interface, GUI). На практике программирование Windows-приложений с GUI предполагает экстенсивное использование различных инструментальных средств и мастеров, которые намного упрощают этот процесс.

Windows Forms – это часть каркаса .NET Framework, которая поддерживает создание приложений со стандартным GUI на платформе Windows.

Форма – это экранный объект, обеспечивающий функциональность программы. Как правило, приложение содержит главное окно, которое реализовано с помощью некоторого класса MyForm, производного от класса Form.

Запускаем Visual Studio .NET, создаем новый проект, для которого выбираем тип Windows Forms Application, задаем имя проекта – FirstForm и сохраняем его в папку, определяемую полем Location.

Окно **Class View** (обзор классов, View → Class View) позволяет перемещаться в коде по выбранному объекту; содержит методы, классы, данные всего листинга проекта. Для перехода, например, в class Form1, щелкаем на соответствующем названии в окне **Class View**.

Окно свойств **Properties** – основной инструмент настройки формы и ее компонент. Содержимое этого окна представляет собой весь список свойств выбранного в данный

момент компонента или формы. Вызывается окно несколькими способами, чаще на выбранном объекте щелкаем правой кнопкой мыши и в контекстном меню пункт

**Properties**. Когда вы только создали проект, в окне **Properties** отображаются свойства самой формы.

Кнопка **Events** переключает окно **Properties** в режим управления обработчиками различных событий (например, мыши, клавиатуры) и одновременно выводит список всех событий компонента. Двойной щелчок мыши в поле значения события генерирует обработчик для него и переключает в режим кода.

Окно **Toolbox** (панель инструментов, **View** —> **Toolbox**) содержит компоненты **Windows-форм**, называемые элементами управления, которые размещаются на форме. Оно состоит из нескольких закладок. Наиболее часто употребляемой закладкой является **All Windows Forms**. Для размещения нужного элемента управления достаточно просто щелкнуть на нем в окне **Toolbox** или, ухватив, перетащить его на форму.

### Режимы дизайна и кода

При создании нового проекта запускается режим дизайна – форма представляет собой основу для расположения элементов управления. Для работы с программой следует перейти в режим кода. Это можно сделать несколькими способами: щелкнуть правой кнопкой мыши в любой части формы и выбрать **View Code**, в окне **Solution Explorer** сделать то же самое на компоненте **Form1**. После перехода в режим кода в этом проекте появится вкладка **Form1.cs\***, нажимая на которую, тоже можно переходить в режим кода.

Базовым классом всех интерфейсных элементов является класс **Control**, содержащийся в пространстве имен **System.Windows.Forms**. В этом классе определены общие для всех интерфейсных элементов свойства, события и методы. Наиболее важные из них:

- **Cursor, Font, BackColor, ForeColor** – свойства, значения которых элемент управления наследует от содержащего его контейнера, если значение этого свойства в явном виде не установлено и не определено в родительском классе;
- **Top, Left, Width, Height, Size, Location** – свойства, отвечающие за размер и местоположение элемента относительно контейнера (для формы контейнером является экран);
- **Anchor и Dock** – свойства, определяющие, согласно каким принципам перемещается и меняет размеры интерфейсный элемент при изменении размеров контейнера;
- **Text, ImeMode, RightToLeft** – свойства, определяющие надпись или текст в элементе управления, а также направление текста и способ его редактирования;
- **Enabled, Visible** – свойства, определяющие, доступен ли пользователю интерфейсный элемент и отображается ли он;
- **Parent** – свойство, указывающее, какой из интерфейсных элементов является контейнером для данного элемента.

### Обработка событий в Windows Forms

GUI управляется событиями. Приложение выполняет действия в ответ на события, вызванные пользователем, например, на щелчок кнопкой мыши или выбор пункта меню. В **Windows Forms** применяется модель обработки событий **.NET**, в которой **делегаты** используются для того, чтобы связать события с обрабатывающими их методами.

В классах **Windows Forms** используются групповые делегаты. Групповой делегат содержит список связанных с ним методов. Когда в приложении происходит событие, управляющий элемент возбуждает событие, вызвав делегат для этого события, который вызывает связанные с ним методы. Для того чтобы добавить делегат к событию, используется перегруженный оператор **+=**.

Например:

```
this.MouseClick += new
MouseEventHandler(this.Form1_MouseClick);
Объявление обработчика для этого события:
```

```
private void Form1_MouseClick(object sender, MouseEventArgs e)
{ }
```

### **Вопросы для проверки знаний**

- Что такое форма и каково ее назначение?
- Что такое элементы управления? На какие группы они делятся?
- Как установить элемент управления на форму и задать его свойства?
- Что такое событие? Как в .Net реализуются события?
- Что такое обработчик события?
- Как выбрать событие для элемента управления?
- Что такое делегат?
- Как задаются обработчики событий для элементов управления?
- Как происходит подключение к прослушиванию события?
- Как создать собственные события и их обработчики?
- Как создать верхнее меню?
- Как добавить выпадающее меню в верхнее меню?
- Как установить определенному пункту меню сочетание клавиш?
- Как создать панель инструментов?
- Как добавить несколько кнопок на панель инструментов?
- Как разместить на кнопке изображение?
- Как добавить новую форму в приложение?
- Как организовать переход к добавленной форме?
- Что такое модальная форма и немодальная? Как они вызываются?
- Как организовать передачу данных между формами?
- Как добавить новый класс в проект?

### **Лабораторная работа 9 «Реализация алгоритмов обработки числовых данных. Отладка приложения»**

**Цель:** получение навыков реализации алгоритмов обработки числовых данных и отладки приложения.

**Задание:** составьте программу вычисления для заданных значений X, Y, Z арифметического выражения (по вариантам).

Для вывода результатов работы программы используйте текстовое окно, которое пользователь сможет растягивать по горизонтали и вертикали. Предусмотрите вертикальную и горизонтальную полосы прокрутки. На рисунке ниже представлен пример выполнения задачи.

Введите значение X: 3,4  
 Введите значение Y: 0,74  
 Введите значение Z: 19,43  
 Результат выполнения программы:  
 X = 3,4  
 Y = 0,74  
 Z = 19,43  
 Результат U = 2,40859707308056  
 Выполнить

Варианты индивидуальных заданий с приведенным ответом для самопроверки:

$$1. \quad t = \frac{2 \cos\left(x - \frac{\pi}{6}\right)}{0.5 + \sin^2 y} \left(1 + \frac{z^2}{3 - z^2/5}\right).$$

При  $x = 14.26$ ,  $y = -1.22$ ,  $z = 3.5 \times 10^{-2}$   $t = 0.564849$ .

$$2. \quad u = \frac{\sqrt[3]{8 + |x - y|^2 + 1}}{x^2 + y^2 + 2} - e^{|x-y|} (\operatorname{tg}^2 z + 1)^x.$$

При  $x = -4.5$ ,  $y = 0.75 \times 10^{-4}$ ,  $z = 0.845 \times 10^2$   $u = -55.6848$ .

$$3. \quad v = \frac{1 + \sin^2(x + y)}{\left|x - \frac{2y}{1 + x^2 y^2}\right|} x^{|y|} + \cos^2\left(\operatorname{arctg} \frac{1}{z}\right).$$

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

При  $x = 0.4 \times 10^4$ ,  $y = -0.875$ ,  $z = -0.475 \times 10^{-3}$   $w = 1.9873$ .

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

При  $x = -15.246$ ,  $y = 4.642 \times 10^{-2}$ ,  $z = 20.001 \times 10^2$   $\alpha = -182.036$ .

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

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

При  $x = 0.1722$ ,  $y = 6.33$ ,  $z = 3.25 \times 10^{-4}$   $\gamma = -172.025$ .

$$8. \quad \varphi = \frac{e^{|x-y|} |x - y|^{x+y}}{\operatorname{arctg}(x) + \operatorname{arctg}(z)} + \sqrt[3]{x^6 + \ln^2 y}.$$

При  $x = -2.235 \times 10^{-2}$ ,  $y = 2.23$ ,  $z = 15.221$   $\varphi = 39.374$ .

$$9. \quad \psi = \left|x^{\frac{y}{x}} - \sqrt[3]{\frac{y}{x}}\right| + (y - x) \frac{\cos y - \frac{z}{(y-x)}}{1 + (y-x)^2}.$$

При  $x = 1.825 \times 10^2$ ,  $y = 18.225$ ,  $z = -3.298 \times 10^{-2}$   $\psi = 1.2131$ .

$$10. \quad a = 2^{-x} \sqrt{x + \sqrt[4]{|y|}} \sqrt[3]{e^{x-1/\sin z}}.$$

При  $x = 3.981 \times 10^{-2}$ ,  $y = -1.625 \times 10^3$ ,  $z = 0.512$   $a = 1.26185$ .

### Лабораторная работа 10 «Реализация алгоритмов поиска. Отладка приложения»

**Цель:** получить навыки реализации алгоритмов поиска и отладки приложения.

**Задание 1:** написать программу для образовательной организации по запросу выводит следующую информацию: список группы; названия предметов, по которым проходят занятия в данной группе, экзаменационные оценки студентов по всем предметам.

Номер группы  ▼

Список студентов

Список предметов

Список студентов с оценками

### Задание 2. Решение задач по вариантам.

1. Найти в массиве все числа, составленные из одних и тех же цифр
2. В заданной последовательности целых чисел найти максимально длинную подпоследовательность чисел такую, что каждый последующий элемент подпоследовательности делился нацело на предыдущий.
3. Дан целочисленный массив размера  $N$ . Определить максимальное количество его одинаковых элементов.
4. Дан целочисленный массив размера  $N$ . Назовем серией группу подряд идущих одинаковых элементов, а длиной серии — количество этих элементов (длина серии может быть равна 1). Вывести строку, содержащую длины всех серий исходного массива.
5. Дан целочисленный массив размера  $N$ . Вывести вначале все его четные элементы, а затем — нечетные.
6. Дано  $N$  целых чисел. Требуется выбрать из них три таких числа, произведение которых максимально.
7. Дан массив ненулевых целых чисел размера  $N$ . Проверить, чередуются ли в нем четные и нечетные числа. Если чередуются, то вывести 0, если нет, то вывести номер первого элемента, нарушающего закономерность.
8. Дан массив из координат  $x$   $N$  точек на прямой. Найти такую точку из данного множества, сумма расстояний от которой до остальных его точек минимальна, и саму эту сумму.
9. Дан целочисленный массив размера  $N$ . Преобразовать его, прибавив к четным числам последний элемент. Первый и последний элементы массива не изменять.
10. Дан целочисленный массив  $A$  размера  $[11]$ . Вывести номер первого из тех его элементов  $A[i]$ , которые удовлетворяют двойному неравенству:  $A[1] < A[i] < A[10]$ . Если таких элементов нет, то вывести 0.

### Лабораторная работа 11 «Реализация обработки табличных данных. Отладка приложения»

**Цель:** получить навыки разработки обработки табличных данных и отладки приложений

*Краткие теоретические сведения*

Представление в виде таблицы используется для отображения списков файлов в папках, результатов расчетов функций, сведений об успеваемости и т.п. Для представления табличных данных можно использовать визуальные компоненты DataGridView и ListView. Эти визуальные компоненты являются наиболее сложными из числа рассмотренных ранее по совокупности свойств, событий и методов. Далее будут приведены начальные сведения по работе с этими компонентами.

Для работы со строками, столбцами и ячейками таблицы предусмотрена целая иерархия классов. Таблицы состоят из колонок (столбцов, Column) и строк (рядов, Row). Пользователю можно разрешить или запретить изменять порядок расположения колонок и

строк. Очевидно, что при изменении порядка необходимо обеспечить программную поддержку обработки с учетом внесенных перестановок. Колонки и строки могут иметь заголовки Headers.

В отличие от массивов, размерность которых задается через константы в момент компиляции программы, применительно к таблицам количество строк (RowCount) и столбцов (ColumnCount) можно менять в ходе выполнения программы (динамическое выделение и освобождение памяти под таблицу). Ячейка (клетка, Cell) таблицы может содержать текст (числа тоже представлены в виде текста), изображения, раскрывающиеся списки и другие формы данных. По умолчанию все ячейки таблицы содержат текст, т.е. являются объектами класса DataGridViewTextBoxCell.

Колонки и строки нумеруются начиная с нуля. Для указания ячейки таблицы используется двумерный индексатор вида [col, row] – первым указывается номер колонки, а затем номер ряда. В качестве номера можно использовать выражения целого типа, в том числе целые константы, переменные целого типа.

Элемент управления ListView также позволяет отображать данные в виде таблицы.

Ячейки таблицы могут содержать текст и изображение, а первый столбец кроме текста еще и значки. Элемент управления ListView также сложен, освещение всех его возможностей выходит за рамки настоящего занятия. Будут представлены наиболее важные функциональные возможности элемента управления ListView, что обеспечит основу для его применения в программе. Будет также охарактеризован элемент управления ImageList, который служит для хранения изображений, используемых в элементе ListView.

#### Свойства компонента DataGridView.

- AllowUserToResizeColumns – разрешает (true) или запрещает (false) пользователю изменять ширину столбцов;
- AllowUserToResizeRows – разрешает (true) или запрещает (false) пользователю изменять высоту строк;
- AutoSizeColumnsMode – определяет режим автоматической установки ширины столбцов (например, выбор значения Fill определит ширину видимых столбцов так, чтобы они заняли всю область, отведенную под таблицу);
- ClipboardCopyMode – указывает возможность и вид копирования содержимого ячейки в буфер обмена;
- ColumnCount – количество столбцов в таблице. Доступно только во время выполнения программы (установка возможна только программным путем);
- ColumnHeadersDefaultCellStyle – стиль заголовка по умолчанию (шрифт, цвет, выравнивание, перенос слов и др.);
- ColumnHeadersVisible – разрешает (true) или запрещает (false) показ заголовков столбцов;
- Columns – коллекция. Позволяет задать заголовки, имена столбцов, тип содержимого столбца во время проектирования таблицы в специальном окне Правка столбцов;
- DefaultCellStyle – стиль ячейки по умолчанию (шрифт, цвета, выравнивание, разрешение переноса слов и др.);
- EditMode – определяет режим начала редактирования, например, по нажатию клавиши Enter или клавиши F2;
- Enabled – указывает доступность (true) или недоступность (false) компонента на форме;
- MultiSelect – разрешает (true) или запрещает (false) выбор нескольких ячеек пользователем;
- ReadOnly – разрешает (false) или запрещает (true) пользователю редактировать ячейки таблицы;
- RowCount – количество рядов в таблице. Доступно только во время выполнения программы (установка возможна только программным путем);

- RowHeadersVisible – разрешает (true) или запрещает (false) показ заголовков рядов.

**Задание:** разработайте программу, обеспечивающую:

- расчет значений функции  $f(x) = 1 - \exp(-a \cdot x) \cdot \cos(b \cdot x)$  для аргумента  $x$ , изменяющегося от  $x_n$  до  $x_k$  с шагом  $x$ . Значения параметров  $a$  и  $b$ , пределы и шаг изменения аргумента задаются вводом. Все величины неотрицательные;
- оформление результатов расчета функции виде таблицы dataGridView из трех колонок (первая соответствует порядковому номеру, вторая – «Аргумент», третья – «Функция»). Количество строк определяется исходными данными;
- сохранение исходных данных и результатов расчета в файле на диске;
- чтение данных из файла и их отображение;
- очистку элементов для ввода исходных данных и таблицы (в таблице оставить названия столбцов и одну пустую строку);
- очистку таблицы при изменении исходных данных.

Расчет значений функции  $f(x) = 1 - \exp(-a \cdot x) \cdot \cos(b \cdot x)$

Параметры функции

a:  b:

X начальное:  X конечное:  dx:

	Аргумент	Функция
1	1,0000	1,15309186567423
2	1,4000	1,23234916856685
3	1,8000	1,14823316922339
4	2,2000	1,03405345266217
5	2,6000	0,965201590369452
6	3,0000	0,952195936293747
7	3,4000	0,970985362851725
8	3,8000	0,994379123384976
9	4,2000	1,00778703290481
10	4,6000	1,00979896795906
11	5,0000	1,00565361949136
12	5,4000	1,00087770675208
13	5,8000	0,998279472034648
14	6,2000	0,997998591128476
15	6,6000	0,998903701222053

Вопросы для проверки знаний

1. Назначение и свойства таблицы dataGridView.
2. Программирование занесения данных в ячейки таблицы.
3. Программирование извлечения данных из ячеек таблицы.
4. Программирование занесения данных в заголовки столбцов таблицы.
5. Программирование занесения данных в заголовки строк таблицы.
6. Типовые события, связанные с таблицей dataGridView.
7. Методы обработки событий таблицы dataGridView.

## Лабораторная работа 12 «Разработка и отладка генератора случайных символов»

**Цель:** получение навыков разработки и отладки генератора случайных символов.

**Задание 1** Разработать генератор случайных чисел.

Случайные числа в языке программирования C# могут быть сгенерированы функцией Next() из стандартной класса Random. В первую очередь необходимо создать объект класса Random. Конструктор этого класса имеет 2 перегрузки: Random() при которой в качестве ключа генерации псевдослучайных чисел используется системное время и Random(int seed) где ключом является переданное функции значение. В языке C#

используется такой алгоритм генерации псевдослучайных чисел, что при введении одного ключа, на выходе всегда будет одинаковый результат.

Пример создания объекта класса Random:

```
Random randomExample = new Random(1);
```

Функция Next() имеет 3 перегрузки:

- Next() генерирует случайное положительное число типа integer.
- Next(int MaxValue) возвращает положительное значение типа integer меньше указанного.
- Next(int minValue, int maxValue) возвращает положительное число в указанных пределах.

Таким образом, используя эту функцию, можно получать как просто случайные значения, так и значения в необходимом диапазоне (масштабированное значение).

Пример получения случайного числа:

```
int a = randomExample.Next();
```

**Задание 2** Разработать программу, использующую масштабируемый генератор случайных чисел.

Ниже показан код программы, которая несколько раз запускает функцию Next().

```
using System;
namespace RandLab
{
    class Program
    {
        static void Main(string[] args)
        {
            Random randomExample = new Random(1);
            for(int i = 0; i < 5; i++)
            {
                Console.WriteLine($"Случайное число №{i+1}: {randomExample.Next()}");
            }
        }
    }
}
```

При повторном запуске программы, печатаются те же самые числа. Это естественное следствие того, что при создании randomExample был указан ключ генерации, равный единице. При отладке программы, внося какие-то изменения, необходимо удостовериться, что программа срабатывает правильно, а это возможно, если входные данные остались те же, то есть сгенерированные числа. Когда программа успешно отлажена, нужно, чтобы при каждом выполнении программы генерировались случайные числа. Для этого нужно воспользоваться другим конструктором класса Random т.е. не указывать ключ генерации. Также можно пересоздать объект randomExample используя другой ключ. Однако при использовании ключа, производного от системного времени нельзя забывать, так как он не является случайным и может быть воспроизведен пользователем. Например, известны случаи злоупотребления в многопользовательских играх, когда пользователи выясняли, что при определенных настройках системного времени генератор псевдослучайных чисел всегда выдает одинаковый известный результат.

Ниже представлен пример программы, в которой первые 5 чисел всегда будут одинаковыми (сгенерированными по ключу 1), а следующие 5 чисел всегда будут различными (использует ключ от системного времени).

```

using System;
namespace RandLab
{
    class Program
    {
        static void Main(string[] args)
        {

            Random randomExample = new Random(1);
            for(int i = 0; i < 5; i++)
            {
                Console.WriteLine($"Случайное число №{i+1}: {randomExample.Next()}");
            }

            randomExample = new Random();
            for (int i = 0; i < 5; i++)
            {
                Console.WriteLine($"Случайное число №{i + 1}: {randomExample.Next()}");
            }
        }
    }
}

```

### Задание 3

Изучить другие возможности класса `Random`.

Помимо генерации целых чисел, класс `Random` также имеет функцию `NextDouble()` которая возвращает число типа `Double` (нецелое с большим количеством знаков после запятой) со значением от 0 до 1.0. Эта функция не имеет дополнительных перегрузок.

Класс `Random` также содержит процедуру (т.е. метод, не возвращающий значение напрямую). `NextBytes(byte[] buffer)`, которая при выполнении заполняет переданный в параметрах массив значениями.

Пример программы, использующей другие возможности класса `Random`:

```

using System;
namespace RandLab
{
    class Program
    {
        static void Main(string[] args)
        {

            Random randomExample = new Random(1);
            byte[] a = new byte[10];
            randomExample.NextBytes(a);
            Double d = randomExample.NextDouble();

        }
    }
}

```

}  
**Задание 4** Разобрать генератор случайных символов. Сформировать случайную символьную последовательность.

### Лабораторная работа 13 «Разработка приложений для моделирования процессов и явлений. Отладка приложения»

**Цель :** приобрести навыки разработки приложений для моделирования процессов и явлений.

*Краткие теоретические сведения:*

Модель – это материальный или идеальный объект, замещающий исследуемую систему и адекватным образом отображающий ее существенные стороны. Модель объекта отражает его наиболее важные качества, пренебрегая второстепенными.

Компьютерная модель (англ. computer model), или численная модель (англ. computational model) – компьютерная программа, работающая на отдельном компьютере, суперкомпьютере или множестве взаимодействующих компьютеров (вычислительных узлов), реализующая представление объекта, системы или понятия в форме, отличной от реальной, но приближенной к алгоритмическому описанию, включающей и набор данных, характеризующих свойства системы и динамику их изменения со временем.

Говоря о компьютерной реконструкции, мы будем подразумевать разработку компьютерной модели определенного физического явления или среды.

Физическое явление – процесс изменения положения или состояния физической системы. Физическое явление характеризуется изменением определенных физических величин, связанных между собой. Например, к физическим явлениям относятся все известные виды взаимодействия материальных частиц.

Основной задачей построения компьютерных моделей является функциональное исследование физического явления или процесса с получением исчерпывающих аналитических данных и реализация интерактивного взаимодействия пользователя с компьютерной моделью.

**Задание 1.** Построить и исследовать модели на примере движения тела, брошенного под углом к горизонту.

В процессе тренировок теннисистов используются автоматы по бросанию мячика в определенное место площадки. Необходимо задать автомату необходимую скорость и угол бросания мячика для попадания в мишень определенного размера, находящуюся на известном расстоянии.

Качественная описательная модель. Сначала построим качественную описательную модель процесса движения тела с использованием физических объектов, понятий и законов, т.е. идеализированную модель движения объекта. Из условия задачи сформулируем следующие основные предположения:

- мячик мал по сравнению с Землей, поэтому его можно считать материальной точкой;
- изменение высоты мячика, поэтому ускорение свободного падения можно считать постоянной величиной  $g=9.8 \text{ м/с}^2$  и движение по оси OY можно считать равноускоренным;
- скорость бросания тела мала, поэтому сопротивлением можно пренебречь и движение по оси OX можно считать равномерным.

Построение формальной модели.

Для формализации модели используем известные из курса физики формулы равномерного и равноускоренного движения. При заданных начальной скорости  $V_0$  и угле бросания  $\alpha$  значения координат дальности полета  $X$  и высоты  $Y$  от времени можно описать следующими формулами:

$$X = V_0 \cdot \cos \alpha \cdot t;$$

$$Y = V_0 \cdot \sin \alpha \cdot t - g \cdot t^2 / 2$$

Пусть мишень высотой  $h$  будет размещаться на расстоянии  $S$  от автомата. Из первой формулы выразим время, которое понадобится мячику, чтобы преодолеть расстояние  $S$ :

$$t=S/(V_0 \cdot \cos \alpha)$$

Подставим это значение во вторую формулу. Получим высоту мячика  $L$  над землей на расстоянии  $S$ :

$$L=S \cdot \operatorname{tg} \alpha - g \cdot S^2 / (2 \cdot V_0^2 - \cos^2 \alpha)$$

Формализуем условие попадания мячика в мишень. Попадание произойдет, если значение высоты  $L$  мячика будет удовлетворять условию в форме неравенства:  $0 \leq L \leq h$ .

Если  $L$  меньше 0, то это значит «недолет», а если  $L$  больше  $h$ , то «перелет».

Исследование модели

Исследуйте модель и определите с заданной точностью 0,10 диапазон изменений угла, который обеспечивает попадание в мишень, находящуюся на расстоянии 30 м и имеющую высоту 1 м, при заданной начальной скорости 18 м/с.

**Задание 2.** На основе анализа индивидуальных биоритмов прогнозировать неблагоприятные дни, выбирать благоприятные дни для разного рода деятельности.

Существует гипотеза, что жизнь человека подчиняется трем циклическим процессам, называемым биоритмами. Эти циклы характеризуют три стороны самочувствия человека: физическую, эмоциональную и интеллектуальную. Не все считают эту теорию строго научной, но многие верят в нее. За точку отсчета всех трех биоритмов берется день рождения человека. Физический биоритм характеризует жизненные силы человека, его периодичность составляет 23 дня.

Эмоциональный биоритм характеризует внутренний настрой человека, продолжительность периода 28 дней.

Третий биоритм характеризует мыслительные способности, интеллектуальное состояние человека, его цикличность – 33 дня

Качественная описательная модель.

Указанные циклы можно описать приведенными ниже выражениями, в которых переменная  $X$  – количество прожитых человеком дней:

Физический цикл  $\text{ФИЗ}(X) = \operatorname{SIN}(2\pi X/23)$

Эмоциональный цикл  $\text{ЭМО}(X) = \operatorname{SIN}(2\pi X/28)$

Интеллектуальный цикл  $\text{ИНТ}(X) = \operatorname{SIN}(2\pi X/33)$

Составьте компьютерную модель, включающую дату рождения человека, дату отсчета и длительность прогноза.

Проведение компьютерного эксперимента

- Провести тестовый расчет компьютерной модели по введенным данным.
- По результатам расчетов постройте график функций для трех биоритмов (по оси  $OX$  – даты,  $OY$  – значение биоритма).
- Определите дни, в которых значение биоритма равно нулю.
- Введите дату своего рождения и дату отсчета (сегодня).
- Проследите пересчет значений и изменения на графике.
- Определите благоприятные и неблагоприятные дни для сдачи зачета по.
- По кривой интеллектуального состояния выбрать дни, когда ответы на уроках будут наиболее удачными
- Когда у людей совпадают или очень близки графики по одному, двум или даже трем биоритмам, то можно предположить довольно высокую совместимость этих людей. Постройте модель физической, эмоциональной и интеллектуальной совместимости двух друзей.

**Задание 3.** Используя модель, выберите: наиболее благоприятные дни для общения; дни, когда вам не рекомендуется общаться; в какой области деятельности вы могли бы преуспеть в паре с другом?

#### Лабораторная работа 14 «Интеграция модуля в информационную систему»

Цель: получить практические навыки разработки модулей программной системы и интеграции этих модулей.

*Краткие теоретические сведения:*

Термин «интеграция» относится к такой операции в процессе разработки ПО, при которой разработчик объединяете отдельные программные компоненты в единую систему. В небольших проектах интеграция может занять одно утро и заключаться в объединении горстки классов. В больших – могут потребоваться недели или месяцы, чтобы связать воедино весь набор программ. Независимо от размера задач в них применяются одни и те же принципы.

Тема интеграции тесно переплетается с вопросом последовательности конструирования. Порядок, в котором создаются классы или компоненты, влияет на порядок их интеграции: нельзя интегрировать то, что еще не было создано. Последовательности интеграции и конструирования имеют большое значение.

Поскольку интеграция выполняется после того, как разработчик завершил модульное тестирование, и одновременно с системным тестированием, ее иногда считают операцией, относящейся к тестированию. Однако она достаточно сложна, и поэтому ее следует рассматривать как независимый вид деятельности.

Аккуратная интеграция обеспечивает:

- упрощенную диагностику дефектов;
- меньшее число ошибок;
- раннее создание первой работающей версии продукта;
- уменьшение общего времени разработки;
- лучшие отношения с заказчиком;
- улучшение морального климата;
- увеличение шансов завершения проекта;
- более надежные оценки графика проекта;
- более аккуратные отчеты о состоянии;
- лучшее качество кода;
- меньшее количество документации.

**Задание 1.** На основе сведений результатах сдачи экзаменов студентами за прошедший семестр:

- а) вычислите средние баллы по каждому предмету;
- б) определите группу из трех лучших учеников;
- в) определите группу из трех самых слабых учеников.

**Задание 2.** Известен расход по  $N$  видам горючего в каждом из  $M$  автохозяйств. Определите для каждого хозяйства вид горючего с наибольшим и с наименьшим расходом.

*Пояснения к выполнению*

1. Оформите внешнюю спецификацию.
2. Составить в виде блок-схемы алгоритм решения задачи.
3. Спроектировать и разработать модули программы для решения задачи на любом алгоритмическом языке программирования.
4. Выполнить отладку и тестирование модулей программы.
5. Выполнить инкрементную интеграцию модулей с использованием одного из подходов.
6. Выполнить системное тестирование программы.
7. Оформить отчет по лабораторной работе.

Отчет по лабораторной работе должен включать:

1. Внешнюю спецификацию.
2. Алгоритм решения задачи.
3. Текст программы на языке программирования.
4. Набор тестов для отладки модулей программы.
5. Описание процесса интеграции модулей.

**Лабораторная работа 15 «Программирование обмена сообщениями между модулями»**

**Цель:** получение навыков программирования обмена сообщениями между модулями.

**Задание 1**

Составить программу, помогающую сотрудникам Государственной инспекции безопасности дорожного движения обработать следующие данные: регистрационный номер автомобиля, марка автомобиля, цвет автомобиля, год выпуска, адрес владельца. Программа должна по требованию пользователя выдавать следующие сведения:

- адреса владельцев автомобилей заданной марки, определенного цвета;
- все данные об автомобиле с заданным регистрационным номером;
- все данные об автомобилях с известной цифровой частью регистрационного номера.

**Задание 2**

Программу, разработанную в задании №1, разбить на модули. Например, создать такие модули, как главный (содержащий функцию main()), чтения из файла в массив структур, вывод на экран содержимого массива структур, сортировка данных (при необходимости), меню, формирование документов и т.д.

**Задание 3**

Разработать схему межмодульных вызовов.

**Задание 4**

Проанализировать способы передачи аргументов между функциями и целесообразность использования глобальных данных.

**Лабораторная работа 16«Организация файлового ввода-вывода данных»**

**Цель:** получить навыки реализации файлового ввода и вывода данных.

**Задание 1.** Работа с двоичными файлами:

Создать файл и записать в него вещественные числа из диапазона от  $a$  до  $b$  с шагом  $h$ . Вывести на экран все компоненты файла с нечетными порядковыми номерами.

Пример:

```
using System;
using System.Text;
using System.IO;
```

```
namespace MyProgram
{
    class Program
    {
        static void Main()
        {
            Console.WriteLine("a= ");
            double a=double.Parse(Console.ReadLine());
            Console.WriteLine("b= ");
            double b=double.Parse(Console.ReadLine());
            Console.WriteLine("h= ");
            double h=double.Parse(Console.ReadLine());
            //Записываем в файл t.dat вещественные числа из заданного диапазона
            FileStream f=new FileStream("t.dat",FileMode.Open);
            BinaryWriter fOut=new BinaryWriter(f);
            for (double i=a; i<=b; i+=h)
            {
                fOut.Write(i);
            }
            fOut.Close();
            //Объекты f и fIn связаны с одним и тем же файлом
            f=new FileStream("t.dat",FileMode.Open);
```

```

BinaryReader fIn=new BinaryReader(f);
long m=f.Length; //определяем количество байт в потоке
//Читаем данные из файла t.dat начиная с элемента с номером 1, т.е с 8 байта,
//перемещая внутренний указатель на 16 байт, т.е. на два вещественных числа
for (long i=8; i<m; i+=16)
{
    f.Seek(i,SeekOrigin.Begin);
    a=fIn.ReadDouble();
    Console.WriteLine("{0:f2} ",a);
}
fIn.Close();
f.Close();
}
}
}

```

*Варианты индивидуальных заданий:*

1. Создать файл и записать в него степени числа 3. Вывести на экран все компоненты файла с четным порядковым номером.
2. Создать файл и записать в него обратные натуральные числа  $1, 1/2, \dots, 1/n$ . Вывести на экран все компоненты файла с порядковым номером, кратным 3.
3. Создать файл и записать в него  $n$  первых членов последовательности Фибоначчи. Вывести на экран все компоненты файла с порядковым номером, не кратным 3.
4. Дана последовательность из  $n$  целых чисел. Создать файл и записать в него все четные числа последовательности. Вывести содержимое файла на экран.
5. Дана последовательность из  $n$  целых чисел. Создать файл и записать в него все отрицательные числа последовательности. Вывести содержимое файла на экран.
6. Дана последовательность из  $n$  целых чисел. Создать файл и записать в него числа последовательности, попадающие в заданный интервал. Вывести содержимое файла на экран.
7. Дана последовательность из  $n$  целых чисел. Создать файл и записать в него числа последовательности, не кратные заданному числу. Вывести содержимое файла на экран.
8. Дана последовательность из  $n$  вещественных чисел. Записать все эти числа в файл. Вывести на экран все компоненты, не попадающие в данный диапазон.
9. Дана последовательность из  $n$  вещественных чисел. Записать все эти числа в файл. Вывести на экран все компоненты файла с нечетными номерами, большие заданного числа.
10. Дана последовательность из  $n$  вещественных чисел. Записать все эти числа в файл. Вывести на экран все компоненты файла с четными номерами, меньше заданного числа.
11. Дана последовательность из  $n$  вещественных чисел. Записать все эти числа в файл. Вывести на экран все положительные компоненты файла.
12. Дана последовательность из  $n$  вещественных чисел. Записать все эти числа в файл. Подсчитать среднее арифметическое компонентов файла, стоящих на четных позициях.
13. Дана последовательность из  $n$  вещественных чисел. Записать все эти числа в файл. Найти максимальное значение среди компонентов файла, стоящих на нечетных позициях.
14. Дано предложение. Создать файл и записать в него все символы данного предложения, отличные от знаков препинания. Вывести содержимое файла на экран.
15. Дано предложение. Создать файл и записать в него все символы данного предложения, отличные от цифр. Вывести содержимое файла на экран.
16. Создать файл, состоящий из слов. Вывести на экран все слова, которые начинаются на заданную букву.

17. Создать файл, состоящий из слов. Вывести на экран все слова, длина которых равна заданному числу.
18. Создать файл, состоящий из слов. Вывести на экран все слова, которые начинаются и заканчиваются одной буквой.
19. Создать файл, состоящий из слов. Вывести на экран все слова, которые начинаются на ту же букву, что и последнее слово.

**Задание 2.** Работа с текстовым (символьным) файлом.

Дан текстовый файл. Найти количество строк, которые начинаются с данной буквы. Файл text.txt должен содержать английский текст

Пример выполнения задания:

```
using System;
using System.Text;
using System.IO;
using System.Text.RegularExpressions;

namespace MyProgram
{
    class Program
    {
        static void Main()
        {
            Console.WriteLine("Введите заданную букву: ");
            char a=char.Parse(Console.ReadLine());
            StreamReader fileIn = new StreamReader("text.txt");
            string text=fileIn.ReadToEnd(); //читываем из файла весь текст
            fileIn.Close();
            int k=0;
            //разбиваем текст на слова используя регулярные выражения
            string []newText=Regex.Split(text,"[ ,:;]+");
            //подсчитываем количество слов, начинающихся на заданную букву
            foreach( string b in newText)
                if (b[0]==a)++k;
            Console.WriteLine("k= "+k);
        }
    }
}
```

*Варианты индивидуальных заданий*

1. Дан текстовый файл. Найти количество строк, которые начинаются и заканчиваются одной буквой.
2. Дан текстовый файл. Найти самую длинную строку и ее длину.
3. Дан текстовый файл. Найти самую короткую строку и ее длину.
4. Дан текстовый файл. Найти номер самой длинной строки.
5. Дан текстовый файл. Найти номер самой короткой строки.
6. Дан текстовый файл. Выяснить, имеется ли в нем строка, которая начинается с данной буквы. Если да, то напечатать ее.
7. Дан текстовый файл. Напечатать первый символ каждой строки.
8. Дан текстовый файл. Напечатать символы с k1 по k2 в каждой строке.
9. Дан текстовый файл. Напечатать все нечетные строки.
10. Дан текстовый файл. Напечатать все строки, в которых имеется хотя бы один пробел.
11. Дан текстовый файл. Напечатать все строки, длина которых равна данному числу.
12. Дан текстовый файл. Напечатать все строки, длина которых меньше заданного числа.
13. Дан текстовый файл. Напечатать все строки с номерами от k1 до k2.

14. Дан текстовый файл. Получить слово, образованное k-ыми символами каждой строки.
15. Дан текстовый файл. Переписать в новый файл все его строки, вставив в конец каждой строки ее номер.
16. Дан текстовый файл. Переписать в новый файл все его строки, вставив в конец каждой строки количество символов в ней.
17. Дан текстовый файл. Переписать в новый файл все его строки, длина которых больше заданного числа.
18. Дан текстовый файл. Переписать в новый файл все его строки четной длины.
19. Дан текстовый файл. Переписать в новый файл все его строки, удалив из них символы, стоящие на четных местах.

### **Лабораторная работа 17 «Разработка модулей экспертной системы»**

**Цель:** Получение практических навыков создания прототипа экспертной системы с использованием алгоритмических языков высокого уровня.

*Краткие теоретические сведения:*

Экспертная система (ЭС) представляет собой интеллектуальную систему, создаваемую с помощью эксперта – профессионала в той предметной (проблемной) области, в которой ЭС будет работать. Знания эксперта используются для создания базы знаний ЭС. Цель ЭС – сделать эти знания доступными для пользователя, когда ему необходимо помочь в решении практической задачи, с которой он самостоятельно справиться не может. ЭС ориентирована на решение трудно формализуемых задач, которые обладают одной или несколькими из следующих характеристик:

- решаемые задачи не могут быть описаны в математической форме;
- цели не могут быть выражены в терминах точно определенной математической целевой функции;
- не существует алгоритмического решения задач и др.

В отличие от традиционных систем обработки данных в ЭС используется символьный (а не числовой) способ представления знаний, символьный вывод на знаниях и эвристический поиск решения задачи, а не исполнение математического алгоритма.

При разработке ЭС вначале создается прототип ЭС, который является начальной версией создаваемой ЭС. Прототип создается для проверки правильности представления в ЭС знаний и рассуждений эксперта. База знаний прототипа ЭС содержит обычно несколько десятков правил, в то время как база знаний действующей ЭС может включать несколько тысяч (в отдельных случаях – десятки тысяч) правил и тестовых примеров. Прототип должен продемонстрировать пригодность используемых методов решения задач, возникающих в данной предметной области.

Процесс функционирования ЭС состоит в следующем. Пользователь ЭС, желающий получить решение своей задачи, через пользовательский интерфейс посылает запрос к ЭС, т.е. ставит задачу. Решатель ЭС, используя базу знаний, в диалоговом режиме с пользователем решает эту задачу и выдает ему ответ или подходящую рекомендацию.

В данной работе ЭС решает задачу распознавания (идентификации), которая заключается в следующем. В базу знаний ЭС заранее заносится информация о множестве объектов, составляющих словарь данной предметной области. Каждый объект характеризуется своим набором значений атрибутов (признаков, свойств). Задача ЭС состоит в том, чтобы по отдельным заданным атрибутам определить название искомого объекта, который содержится в базе знаний. Таких объектов может быть несколько.

#### **Задание:**

Выбрать вариант предметной области (студент может предложить свой вариант предметной области).

#### Перечень предметных областей (варианты задания)

№ варианта	Предметная область	Объекты (словарь базы)	Ориентировочные свойства объектов	Ориентировочный запрос пользователя
------------	--------------------	------------------------	-----------------------------------	-------------------------------------

		знаний)	(знания об объектах)	(задача ЭС)
1	Сотрудники фирмы	Фамилии сотрудников	Возраст, способности, число прогулов и др.	Кого из сотрудников премировать?
2	Предприятия области	Названия предприятий	Объем продукции, число рабочих, площадь и др.	Какое предприятие объявить банкротом?
3	Работники предприятия	Фамилии работников	Возраст, способности, число прогулов и др.	Кого из работников уволить?
4	Строительство	Строительные объекты	Стоимость, наличие средств и др.	Какой из объектов строить первым?
5	Товарная продукция	Названия товаров	Цена, дефицитность, качество и др.	Какой из товаров приобрести?
6	Сводка погоды	Дни месяца	Температура, влажность, осадки, давление и др.	Определить аномальный день месяца
7	Уголовные преступления	Фамилии преступников	Приметы: возраст, рост, цвет глаз и др.	Найти преступника
8	Нарушения правил	Фамилии водителей	Возраст водителя, цвет автомобиля и др.	Найти водителя, часто нарушающего правила
9	Автопредприятие	Фамилии водителей	Возраст, семейное положение, прогулы и др.	Кого послать в трудный рейс?
10	Абоненты сотовой связи	Фамилии абонентов	Возраст, льготы, задолженность и др.	Кто может перейти к другому оператору?
11	Студенты ВУЗа	Фамилии студентов	Успеваемость, задолженность и др.	Кого исключить?
12	Футбольная команда	Фамилии игроков	Амплуа, результативность штрафное время и др.	Кого послать на чемпионат?
13	Хоккейная команда	Фамилии игроков	Амплуа, результативность, штрафное время и др.	Кого послать на чемпионат?
14	Команда боксеров	Фамилии боксеров	Весовая категория, число боев, число побед и др.	Кого послать на соревнование?
15	Космические объекты	Перечень объектов	Характеристики объектов	Определить искомый объект
16	Одежда	Виды одежды	Модность, цена, работа, отдых, театр, пляж и др.	Что купить в первую очередь
17	Справочник	ВУЗы	Дневной, вечерний, плата, возможности и др.	Куда пойти учиться

- Составить перечень объектов (словарь) (10-15 наименований).
- Определить свойства (признаки) объектов (5-7 наименований).

- Установить состав базы знаний.
- Сформулировать запрос пользователя (задачу распознавания) с целью определения искомого объекта по некоторым заданным его свойствам (3-4 наименования).
- Разработать вопросник (соответствующие вопросы) для осуществления диалога пользователя с ЭС.
- Разработать блок-схему (пошаговое описание) алгоритма решения задачи ЭС.

### Лабораторная работа 18 «Создание сетевого сервера и сетевого клиента»

**Цель:** Получение практических навыков создания сетевого сервера и сетевого клиента.

*Краткие теоретические сведения:*

Код создания сервера, работающий по протоколу TCP, с помощью сокетов, следующей:

```
using System;
using System.Text;
using System.Net;
using System.Net.Sockets;
namespace SocketTcpServer
{
class Program
{
static int port = 8005; // порт для приема входящих запросов
static void Main(string[] args)
{
// получаем адреса для запуска сокета
EndPoint ipPoint = new EndPoint(IPAddress.Parse("127.0.0.1"), port);
// создаем сокет
Socket listenSocket = new Socket(AddressFamily.InterNetwork, SocketType.Stream,
ProtocolType.Tcp);
try
{
// связываем сокет с локальной точкой, по которой будем принимать данные
listenSocket.Bind(ipPoint);

// начинаем прослушивание
listenSocket.Listen(10);

Console.WriteLine("Сервер запущен. Ожидание подключений...");

while (true)
{
Socket handler = listenSocket.Accept();
// получаем сообщение
StringBuilder builder = new StringBuilder();
int bytes = 0; // количество полученных байтов
byte[] data = new byte[256]; // буфер для получаемых данных
do
{
bytes = handler.Receive(data);
builder.Append(Encoding.Unicode.GetString(data, 0, bytes));
}
while (handler.Available > 0);
Console.WriteLine(DateTime.Now.ToShortTimeString() + ": " + builder.ToString());
// отправляем ответ
```

```

        string message = "ваше сообщение доставлено";
        data = Encoding.Unicode.GetBytes(message);
        handler.Send(data);
        // закрываем сокет
        handler.Shutdown(SocketShutdown.Both);
        handler.Close();
    }
}
catch(Exception ex)
{
    Console.WriteLine(ex.Message);
}
}
}
}

```

### Задание

После создания сокета свяжите его с локальной точкой методом Bind:

```
listenSocket.Bind(ipPoint);
```

Сокет будет прослушивать подключения по 8005 порту на локальном адресе 127.0.0.1. То есть клиент должен будет подключаться к локальному адресу и порту 8005.

Начните прослушивание:

```
listenSocket.Listen(10);
```

Метод Listen вызывается только после метода Bind. В качестве параметра он принимает количество входящих подключений, которые могут быть поставлены в очередь сокета.

После вызова метода Listen начните прослушивание входящих подключений, и если подключения приходят на сокет, то их можно получить с помощью метода Accept:

```
Socket handler = listenSocket.Accept();
```

Метод Accept извлекает из очереди ожидающих запрос первый запрос и создает для его обработки объект Socket. Если очередь запросов пуста, то метод Accept блокирует вызывающий поток до появления нового подключения.

Для обработки запроса сначала в цикле do..while получите данные с помощью метода Receive:

```
do{bytes = handler.Receive(data);builder.Append(Encoding.Unicode.GetString(data, 0, bytes));}while (handler.Available > 0);
```

Метод Receive в качестве параметра принимает массив байтов, в который считываются полученные данные, и возвращает количество полученных байтов.

После получения данных клиенту посылается ответное сообщение с помощью метода Send, который в качестве параметра принимает массив байтов:

```
handler.Send(data);
```

В конце обработки запроса закройте связанный с ним сокет:

```
handler.Shutdown(SocketShutdown.Both);handler.Close();
```

Добавьте проект для клиента.

Полный код клиента:

```

using System;
using System.Text;
using System.Net;
using System.Net.Sockets;
namespace SocketTcpClient
{
class Program
{
// адрес и порт сервера, к которому будем подключаться

```

```

static int port = 8005; // порт сервера
static string address = «127.0.0.1»; // адрес сервера
static void Main(string[] args)
{
try
{
IPEndPoint ipPoint = new IPEndPoint(IPAddress.Parse(address), port);
    Socket socket = new Socket(AddressFamily.InterNetwork, SocketType.Stream,
ProtocolType.Tcp);
    // подключаемся к удаленному хосту
    socket.Connect(ipPoint);
    Console.WriteLine("Введите сообщение:");
    string message = Console.ReadLine();
    byte[] data = Encoding.Unicode.GetBytes(message);
    socket.Send(data);

    // получаем ответ
    data = new byte[256]; // буфер для ответа
    StringBuilder builder = new StringBuilder();
    int bytes = 0; // количество полученных байт
    do
    {
        bytes = socket.Receive(data, data.Length, 0);
        builder.Append(Encoding.Unicode.GetString(data, 0, bytes));
    }
    while (socket.Available > 0);
    Console.WriteLine("ответ сервера: " + builder.ToString());

    // закрываем сокет
    socket.Shutdown(SocketShutdown.Both);
    socket.Close();
}
catch(Exception ex)
{
    Console.WriteLine(ex.Message);
}
Console.ReadLine();
}
}
}

```

Для клиента характерно все то же самое, только теперь после создания сокета вызывается метод **Connect()**, в который передается адрес сервера:

```
IPEndPoint ipPoint = new IPEndPoint(IPAddress.Parse(address), port);socket.Connect(ipPoint)
```

Запустите программы сервера и клиента. Консоль клиента:

Введите сообщение: привет мир

ответ сервера: ваше сообщение доставлено

Консоль сервера:

Сервер запущен. Ожидание подключений...

22:34: привет мир

#### Тема 5.2.4. Разработка прикладных решений информационной системы

### Практическая работа 5 «Подсистемы информационной базы на платформе 1С: Предприятие 8.3...»

Задание : Создать информационную базу «Расписание», которая будет обладать следующим функционалом

- a. Вести график работы специалистов организации
- b. Записывать клиентов\посетителей на прием в формате дата и время
- c. Возможность отменить прием
- d. Проверять занятость специалистов перед записью

Варианты организаций:

1	Поликлиника
2	Салон красоты
3	Свадебное ателье
4	Стоматологическая клиника
5	Школа искусств
6	Йога - центр
7	Автосервис

### Тема 5.2.5. Разработка прикладных решений информационной системы

#### Практическая работа 6 «Разработка конфигурации «Личные доходы»

**Цель:.** Получить практические навыки разработки конфигурации информационной базы.

**Задание:** В ранее созданной информационной базе «Личные доходы» в документе «Мои доходы» создать событие «ПередЗаписью». Данная процедура должна проверять, чтобы все поля в документе были заполнены и данные были корректны введены.

*Методические указания*

1. Событие ПередЗаписью прикладного объекта отличается от события ПередЗаписью формы, связанной с этим прикладным объектом. Обработчик события в модуле формы вызывается при интерактивной записи, а обработчик в модуле объекта при любом способе записи элемента в базу данных.

2. Считается грубой ошибкой в процедурах-обработчиках событий объектов вызывать такие интерактивные команды, как Вопрос и Предупреждение. Эти команды показывают на экране диалоговое окно и ждут реакции пользователя. Так как событие обрабатывается в рамках транзакции, то это вызовет значительную задержку в обработке события и часть данных (или вся таблица) будет заблокирована на время ожидания.

#### Практическая работа 7«Разработка конфигурации «Расписание»

**Цель:.** Получить практические навыки разработки конфигурации информационной базы.

**Задание 1:** Для созданной ранее ИС «Расписание» создать 2 отчета.

Название отчет	Описание
Занятость специалистов	Данный отчет должен выводить список специалистов и их график занятости. У пользователя должна быть возможность, выбирать за какой период он желает посмотреть данные.
График приемов	Данный отчет должен выводить список всех будущих приемов со следующей информацией: фио посетителя, дата приема, специалист.

**Задание 2:** В ранее созданной конфигурации «Расписание» создать документы: доходы организации и расходы организации.

Обязательные реквизиты для документов:

- a. Наименование операции
- b. Дата операции
- c. Сумма
- d. Общий комментарий

При желании студент может создать дополнительные реквизиты!

**Задание 3:** Создать регистр накопления «Текущие обороты» с типом «обороты». Создать движение документов по данному регистру.

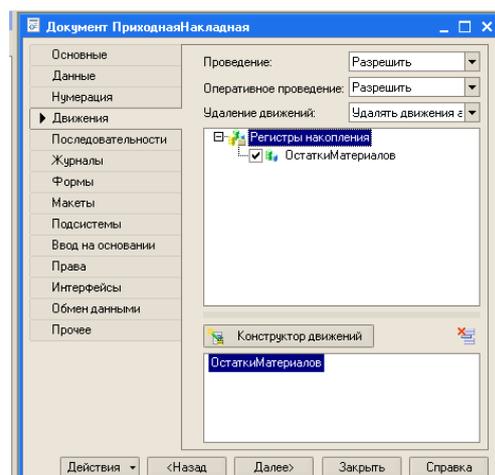
*Методические указания*

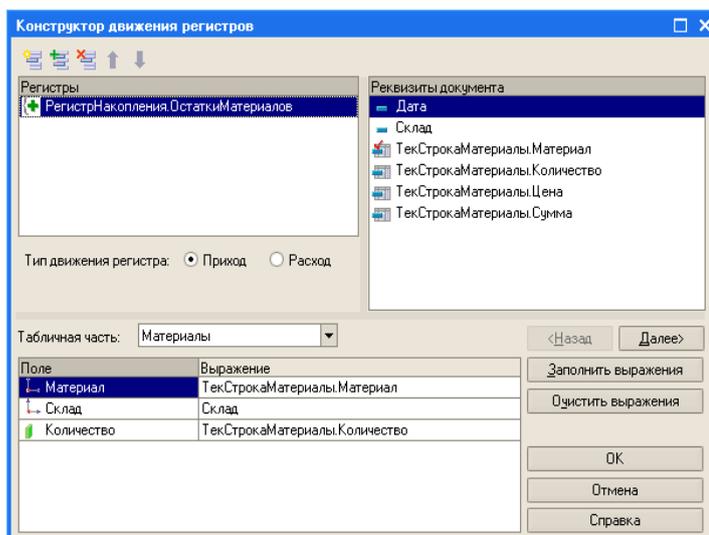
1. Создайте новый объект конфигурации Регистр накопления с именем ОстаткиМатериалов.
2. Измерения регистра:
  - Материал – тип СправочникСсылка.Номенклатура
  - Склад – тип СправочникСсылка.Склады.
3. Ресурс регистра Количество с длиной 15 и точностью 3.
4. В результате эти действий регистр ОстаткиМатериалов должен иметь вид, представленный на рисунке.



Движения документа – это записи в регистрах, которые создаются в процессе проведения документа и отражают изменения, производимые документом.

5. Откройте окно редактирования документа ПриходнаяНакладная.
6. На закладке Движения в списке регистров отметьте регистр ОстаткиМатериалов.
7. После отметки регистра станет доступной кнопка Конструктор движений. Воспользуйтесь ею.
8. По умолчанию Конструктор предлагает создавать движения прихода. В этой накладной это и нужно.
9. В поле выбора Табличная часть выберите табличную часть документа Материалы. Список реквизитов документа заполнится автоматически.
10. Теперь необходимо нажать кнопку Заполнить выражения для того, чтобы в нижнем окне сформировалось соответствие полей и выражений.
11. Это соответствие можно заполнить и другим способом: устанавливать курсор в поле Выражение соответствующего реквизита регистра, а в окне Реквизиты документа двойным щелчком мыши отмечать нужный реквизит. При этом будет заполняться выражение.





12. Нажмите кнопку ОК и посмотрите, какой текст сформировал конструктор движений в модуле объекта.

```

Документ ПриходнаяНакладная: Модуль объекта
[ Процедура ОбработкаПроведения (Отказ, Режим)

    /// (_КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
    // Данный фрагмент построен конструктором.
    // При повторном использовании конструктора, внесенные вручную изменения будут утеряны!!!
    Для Каждого ТекСтрокаМатериалы Из Материалы Цикл
        // регистр ОстаткиМатериалов Приход
        Движение = Движения.ОстаткиМатериалов.Добавить ();
        Движение.ВидДвижения = ВидДвиженияНакопления.Приход;
        Движение.Период = Дата;
        Движение.Материал = ТекСтрокаМатериалы.Материал;
        Движение.Склад = Склад;
        Движение.Количество = ТекСтрокаМатериалы.Количество;
    КонечЦикла;
    // записываем движения регистров
    Движения.ОстаткиМатериалов.Записать ();
    ///) _КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
КонечПроцедуры

```

13. Запустите 1С:Предприятие в режиме отладки и откройте одновременно два окна: список документов ПриходнаяНакладная и список регистра накопления ОстаткиМатериалов.
14. Откройте Приходную накладную и нажмите кнопку ОК. При проведении документа в регистре накопления должны появиться соответствующие записи.

## МДК.05.03 Тестирование кода информационных систем

### Тема 5.3.1. Отладка и тестирование информационных систем

#### Лабораторная работа 1 «Разработка тестового сценария проекта»

**Цель:** получить навыки разработки тестовых сценариев проекта.

**Задание 1** Написать программу решения квадратного уравнения  $ax^2 + bx + c = 0$ .

#### Задание 2

Изучите минимальный набор тестов для программы нахождения вещественных корней квадратного уравнения  $ax^2 + bx + c = 0$ . Решение представлено в таблице.

Но- мер теста	a	b	c	Ожидаемый результат	Что проверяется
1	2	-5	2	$x_1=2, x_2=0,5$	Случай вещественных корней
2	3	2	5	Сообщение	Случай комплексных корней
3	3	-12	0	$x_1=4, x_2=0$	Нулевой корень
4	0	0	10	Сообщение	Неразрешимое уравнение
5	0	0	0	Сообщение	Неразрешимое уравнение
6	0	5	17	Сообщение	Неквадратное уравнение
7	9	0	0	$x_1=x_2=0$	Нулевые корни

Таким образом, для этой программы предлагается минимальный набор функциональных тестов, исходя из 7 классов выходных данных.

### Задание 3

Разработайте набор тестовых сценариев (как позитивных, так и негативных) для следующей программы:

Имеется консольное приложение (разработайте самостоятельно). Ему на вход подается 2 строки. На выходе приложение выдает число вхождений второй строки в первую.

Например:

Строка 1	Строка 2	Вывод
абвгабвг	абв	2
вапродж	ап	1

Набор тестовых сценариев запишите в виде таблицы, приведенной выше.

Оформите отчет.

*Вопросы для проверки знаний*

1. Понятие тестирования. Виды тестирования
2. Место тестирования в жизненном цикле разработки ПО. Подходы к тестированию на разных этапах жизненного цикла
3. Ошибки в требованиях. Неполнота, некорректность, двусмысленность, непроверяемость. Примеры
4. Основные виды ошибок в проекте
5. Оценка качества тестовой базы
6. Критерии выбора тестов на интерфейс
7. Инструменты тестирования веб-сайтов

### Лабораторная работа 2 «Разработка тестовых пакетов»

**Цель:** получить навыки разработки тестовых пакетов.

**Задание** В ходе выполнения лабораторной работы провести тестирование по принципу «белого ящика».

Критерии покрытия кода:

- покрытие операторов – каждая ли строка исходного кода была выполнена и протестирована;
- покрытие условий – каждая ли точка решения (вычисления истинно ли или ложно выражение) была выполнена и протестирована;
- покрытие путей – все ли возможные пути через заданную часть кода были выполнены и протестированы;
- покрытие функций – каждая ли функция программы была выполнена;
- покрытие вход/выход – все ли вызовы функций и возвраты из них были выполнены;
- покрытие значений параметров – все ли типовые и граничные значения параметров были проверены.

### Метод покрытия операторов

Целью этого метода тестирования является выполнение каждого оператора программы хотя бы один раз.

Пример:

Если для тестирования задать значения переменных  $A = 2, B = 0, X = 3$ , будет реализован путь *ace*, т. е. каждый оператор программы выполнится один раз (рис. 1, *a*). Но если внести в алгоритм ошибки – заменить в первом условии *and* на *or*, а во втором  $X > 1$  на  $X < 1$  (рис. 1, *б*), ни одна ошибка не будет обнаружена (табл. 1). Кроме того, путь *abd* вообще не будет охвачен тестом, и если в нем есть ошибка, она также не будет обнаружена. В табл. 1 ожидаемый результат определяется по блок-схеме на рис. 1, *a*, а фактический – по рис. 1, *б*. Как видно из этой таблицы, ни одна из внесенных в алгоритм ошибок не будет обнаружена.

Таблица 1. Результат тестирования методом покрытия операторов

Тест	Ожидаемый результат	Фактический результат	Результат тестирования
$A=2, B=0$	$X=2,5$	$X=2,5$	Неуспешно

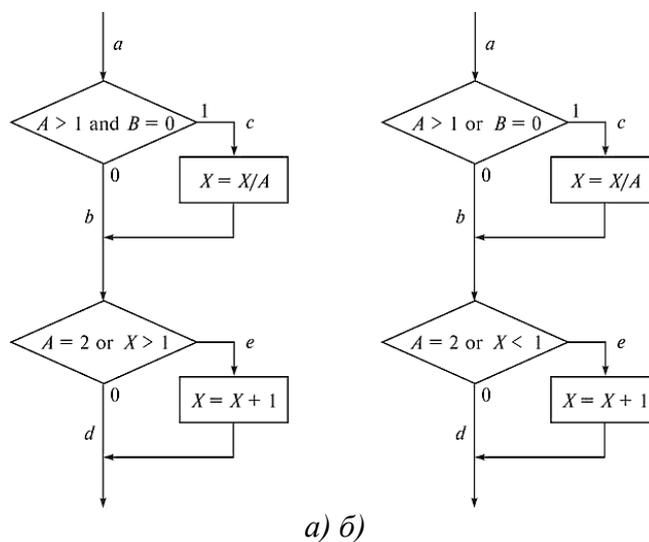


Рис. 1. Пример алгоритма программы: *a* – правильный; *б* – с ошибкой

### Метод покрытия решений (покрытия переходов)

Согласно методу покрытия решений каждое направление перехода должно быть реализовано, по крайней мере, один раз. Этот метод включает в себя критерий покрытия операторов, так как при выполнении всех направлений переходов выполнятся все операторы, находящиеся на этих направлениях.

Для программы, приведенной на рис. 1, покрытие решений может быть выполнено двумя тестами, покрывающими пути  $\{ace, abc!\}$ , либо  $\{acc1, abe\}$ . Для этого выберем следующие исходные данные;  $\{A = 3, B = 0, X = 3\}$  – в первом случае и  $\{A = 2, B = 1, X = 1\}$  – во втором. Однако путь, где  $X$  не меняется, будет проверен с вероятностью 50%: если во втором условии вместо условия  $X > 1$  записано  $X < 1$ , то ошибка не будет обнаружена двумя тестами.

Результаты тестирования приведены в табл. 2.

Таблица 2. Результат тестирования методом покрытия решений

Тест	Ожидаемый результат	Фактический результат	Результат тестирования
$A=3, B=0, X=2$	$X=1$	$X=1$	Неуспешно
$A=2, B=1, X=1$	$X=2$	$X=1$	Упешно

### Тема 5.3.1. Отладка и тестирование информационных систем

#### Лабораторная работа 3 «Использование инструментария анализа качества»

**Цель:** получить навыки использования инструментария анализа качества.

**Задание 1.** Написать программу, генерирующую массив вещественных чисел в диапазоне от -10 до 10 и определяющую все минимальные положительные элементы.

**Задание 2.** Оценить эффективность разработанной программы:

	Исходная программа		Улучшенная программа	
	Недостатки	Количественная оценка	Улучшения	Количественная оценка
Время выполнения				
Оперативная память				
Внешняя память				

**Задание № 3.** Оценить качество разработанной программы:

	Правильность	Универсальность	Проверяемость	Точность результатов
Недостатки				
Оценка				

#### Лабораторная работа 4 «Анализ и обеспечение обработки исключительных ситуаций»

**Цель:** получение навыков анализа и обеспечения обработки исключительных ситуаций.

**Задание 1.** Написать программу, в которой обрабатываются следующие исключительные ситуации: "отрицательное значение возраста" и "год рождения, больше текущего". То есть не может быть осуществлен ввод отрицательного возраста и не может быть осуществлен ввод года рождения больше 2020.

**Задание 2.** Составить программу циклического вычисления значений функций, определенных из таблицы вариантов заданий. Значения R должны вводиться с клавиатуры. R1 и R2 –вещественные, R3 –комплексное. Предусмотреть вывод подсказок, например:

Funkciya  $\sin(x)$

Q –Vyihod iz programmy

Vvedite chislo ili Q:

Для вычисления значений функции написать функцию, вычисляющую требуемые по заданию значения.

Предусмотреть анализ всей введенной информации на ошибки. Вывод на экран и чтение с клавиатуры организовать при помощи стандартных потоков ввода/вывода/ошибки.

Вывести исходные данные и результат в виде(например):  $\sin(R) = rez$ , где rez –результаты вычисления (вещественный)

Вариант	Функция	Обработчики
1	$\sin(R1) \cdot \pi / R2 - R3$	Потеря разряда Деление на 0
2	$\sin(R2) / \pi \cdot R1 + R3$	Потеря разряда Переполнение
3	$\tan(R1) / R3 + \text{Cmod}(R3)$	Потеря разряда Прерывание
4	$\arctan(R1) \cdot R2 + R3$	Потеря разряда Переполнение
5	$\ln(R1 - R2) \cdot R2 - R3$	Обл. опр. арг. Исчезновение порядка

### Лабораторная работа 5 «Функциональное тестирование»

**Цель:** научиться производить функциональное тестирование кода и оценивать его покрытие и качество тестов.

#### Задание.

1. Разработать функцию в соответствии со своим вариантом.
2. Разработать функциональные тесты для написанного кода методом черного ящика. Добиться 100% прохождения этих тестов. Описать принципы выбора тестов.
3. С помощью утилиты анализа покрытия кода оценить покрытие кода тестами. Написать недостающие тесты. Добиться 100% покрытия по ветвям.
4. Оценить качество тестовой базы методом инъекции ошибок. Исправить тестовую базу.

#### *Варианты индивидуальных заданий*

1. Компилятор простых арифметических выражений, например  $2+(-5)*(7-8)$ . Вход и выход в виде строк.
2. Функция поиска пути в неориентированном графе методом поиска в ширину
3. Функция поиска пути в неориентированном графе методом поиска в глубину
4. Функция поиска пути в неориентированном графе методом A\*
5. Функция балансировки двоичного дерева
6. Функция, рассчитывающая контур пересечения двух треугольников
7. Хеш-таблица, не перетирающая элементы при вводе значений с совпадающим ключом, а хранящая список таких элементов и, соответственно, возвращающая их методом get. Метод – двойное хеширование
8. Функция, рассчитывающая следующий ход в игре крестики-нолики на доске заданного размера и для заданной длины выигрышной последовательности путем построения полного дерева решений (например, доска 5 на 5 и длина выигрышной последовательности 4)
9. Функция, производящая поиск заданного набора строк в текстовом файле. Поиск должен уметь находить любую строку из набора, при этом должен правильно обрабатывать переносы текста. Использовать алгоритм Ахо-Корасик.
10. Парсер, использующий простые регулярные выражения, вводимые с клавиатуры, содержащие управляющие конструкции . – любой символ, \* - 0 и более символов, + - 1 и более символов (вводится регулярное выражение и строка, результатом является позиция, с которой это выражение встречается в тексте)
11. Молекула ДНК состоит из последовательностей нуклеотидов А, Г, Ц и У. Несколько одинаковых молекул известной длины были нарезаны на фрагменты произвольной длины. Функция восстанавливает исходную молекулу в том случае, если это возможно сделать единственным образом  
Пример: АГЦЦГГУААЦЦ нарезана на фрагменты АГЦЦ, ЦГГУ, ГГУАА и УААЦЦ.  
Пример невозможной последовательности: АГЦЦГГУААЦЦ нарезана на фрагменты АГЦЦ, ГГУАА и УААЦЦ.
12. Вариант алгоритма решения. В памяти строится ориентированный граф, в вершинах которого находятся фрагменты, а связи соединяют два фрагмента, если фрагмент-источник может быть слева от фрагмента-приемника. Далее в графе ищутся все пути и для каждого проверяется, что он содержит в себе все фрагменты. Если такой путь один, то задача считается решенной
13. Вариант 12 В матричной форме задается система линейных уравнений, необходимо ее решить (например, методом Гаусса).
13. Реализовать структуру «Список с пропусками», см. Реализовать функции добавления, удаления и поиска.
14. В заданном произвольном тексте найти все повторяющиеся фрагменты текста длиной не менее трех слов (без использования стемминга, т.е. слова в различных склонениях и падежах считаются разными, знаки препинания не учитываются). При этом для каждого повторяющегося фрагмента должна указываться максимальная длина, например, для данного текста

- «поиска пути в неориентированном графе методом поиска в” встречается дважды
- «поиска пути в неориентированном графе методом” встречается трижды
- Более короткие части отдельно не встречаются, поэтому не рассматриваются

### Лабораторная работа 6 «Тестирование безопасности»

**Цель:** получение навыков тестирования безопасности информационной системы.

*Краткие теоретические сведения:*

Тестирование безопасности – оценка уязвимости программного обеспечения к различным атакам.

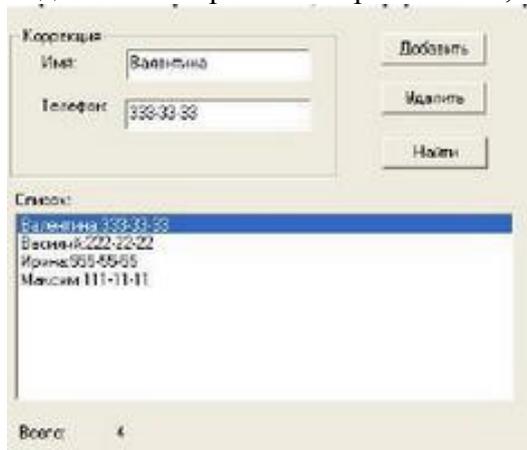
Компьютерные системы очень часто являются мишенью незаконного проникновения.

Под проникновением понимается широкий диапазон действий: попытки хакеров проникнуть в систему из спортивного интереса, мести рассерженных служащих, взлом мошенниками для незаконной наживы. Тестирование безопасности проверяет фактическую реакцию защитных механизмов, встроенных в систему, на проникновение. В ходе тестирования безопасности испытатель играет роль взломщика. Ему разрешено все:

- попытки узнать пароль с помощью внешних средств;
- атака системы с помощью специальных утилит, анализирующих защиты;
- подавление, ошеломление системы (в надежде, что она откажется обслуживать других клиентов);
- целенаправленное введение ошибок в надежде проникнуть в систему в ходе восстановления;
- просмотр несекретных данных в надежде найти ключ для входа в систему.

При неограниченном времени и ресурсах хорошее тестирование безопасности взломает любую систему. Задача проектировщика системы – сделать цену проникновения более высокой, чем цена получаемой в результате информации.

**Задание 1.** Разработать приложение, интерфейс которого представлен на рисунке.



**Задание 2** Добавить в программу форму авторизации по имени и паролю.

### Лабораторная работа 7 «Нагрузочное тестирование, стрессовое тестирование»

**Цель:** произвести стресс тестирование Thin интерфейса приложения.

**Задание.**

1. Взять приложение, разработанное в рамках предыдущих лабораторных работ.
2. Разработать тестовый сценарий нагрузочного тестирования (рекомендуется использовать JMeter)
3. Ответить на вопрос – сколько запросов в секунду может обработать приложение при условии, что они идут последовательно.
4. Построить график зависимости времени ответа от количества параллельных запросов (рассматривать логарифмическую шкалу по основанию два, т.е. 1, 2,4,8,16,32 и т.д. запроса)

5. Ответить на вопрос – какое максимальное количество параллельных запросов может обработать приложение без сбоев.

### **Лабораторная работа 8 «Тестирование интеграции»**

**Цель:** получение навыков тестирования интеграции.

**Задание 1** Разработать приложение, состоящее из трех модулей:

- главный модуль, считывающий из текстового файла координаты точек на плоскости;
- модуль, содержащий функции расчета расстояния между двумя точками;
- модуль, содержащий функцию, определяющую треугольник с максимальной площадью.

**Задание 2**

Описать этапы нисходящего проектирования разработанного приложения.

**Задание 3**

Описать этапы восходящего проектирования разработанного приложений.

### **Лабораторная работа 9 «Конфигурационное тестирование»**

**Цель:** получение навыков проведения конфигурационного тестирования.

*Краткие теоретические сведения*

Особенности конфигурационного тестирования.

Конфигурационное тестирование (Configuration testing). Проверяется работоспособность при различных конфигурациях, предполагает тестирование работы системы на различных платформах: различных вариантах аппаратной конфигурации, версиях операционной системы и окружения.

**Задание 1**

Дана структура с именем ZNAK, состоящая из полей:

- фамилия, имя;
- знак Зодиака;
- дата рождения (массив из трех чисел).

Написать программу, которая выполняет следующие действия:

- ввод с клавиатуры данных в массив, состоящий из 8 элементов типа ZNAK, и занесение их в файл данных;
- чтение данных из файла и вывод их на экран;
- вывод на экран информации о людях, родившихся в месяц, значение которого введено с клавиатуры (если таких нет – вывести об этом сообщение);
- список должен быть упорядочен по знакам Зодиака.

**Задание 2**

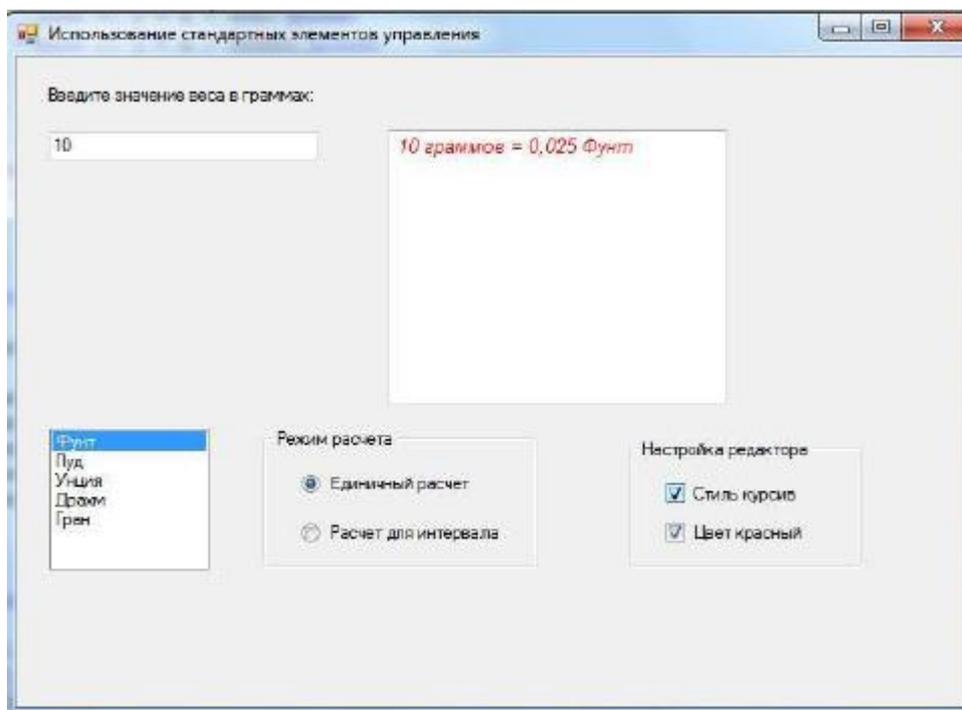
Описать и обосновать итоги тестирования работы разработанного приложения на различных платформах: различных вариантах аппаратной конфигурации, версиях операционной системы и окружения.

### **Лабораторная работа 10 «Тестирование установки»**

**Цель:** получение навыков тестирования установки.

**Задание 1**

Разработать приложение, интерфейс которого представлен на рисунке



## Задание 2

Провести комплексное тестирование разработанного приложения.

## МДК 05.02. Разработка кода информационных систем

### Примерная тематика курсовых работ по модулю:

1. Проектирование и разработка программного модуля информационной системы «Спортивный клуб».
2. Проектирование и разработка программного модуля информационной системы учета продаж в магазине одежды.
3. Проектирование и разработка программного модуля информационной системы по учету товаров и услуг.
4. Проектирование и разработка программного модуля информационной системы для магазина бытовой техники.
5. Проектирование и разработка программного модуля информационной системы учета услуг и продаж в компании бытового обслуживания.
6. Проектирование и разработка программного модуля информационной системы для организации сервисного обслуживания населения.
7. Проектирование и разработка информационной системы учета продаж и услуг для дилерского центра «Mercedes-Benz».
8. Проектирование и разработка программного модуля информационной системы учета продаж и услуг для магазина садово-огородных товаров.
9. Проектирование и разработка программного модуля информационной системы по учету продаж запчастей.
10. Проектирование и разработка программного модуля информационной системы по учету товаров и услуг станции технического обслуживания.
11. Проектирование и разработка программного модуля информационной системы "Мебельная мастерская".
12. Проектирование и разработка программного модуля информационной системы "Центр детского творчества".
13. Проектирование и разработка программного модуля информационной системы "Учебный центр".

14. Проектирование и разработка программного модуля информационной системы "Стоматологическая клиника".
15. Проектирование и разработка программного модуля информационной системы "Транспортная компания".
16. Проектирование и разработка программного модуля информационной системы "Ремонт автомобилей".
17. Проектирование и разработка программного модуля информационной системы "Автоцентр".

*Индивидуальные задания к учебной и производственной практикам представлены в программах практик.*

### **МДК.05.01 Проектирование и дизайн информационных систем**

#### **Вопросы к экзамену**

1. Основные понятия и определения ИС.
2. Жизненный цикл информационных систем
3. Организация и методы сбора информации.
4. Анализ предметной области. Основные понятия системного и структурного анализа.
5. Постановка задачи обработки информации.
6. Основные виды, алгоритмы и процедуры обработки информации, модели и методы решения задач обработки информации.
7. Основные модели построения информационных систем, их структура, особенности и области применения.
8. Сервисно - ориентированные архитектуры.
9. Анализ интересов клиента. Выбор вариантов решений
10. Методы и средства проектирования информационных систем.
11. Case-средства для моделирования деловых процессов (бизнес-процессов). Инструментальная среда –структура, интерфейс, элементы управления.
12. Принципы построения модели IDEF0: контекстная диаграмма, субъект моделирования, цель и точка зрения.
13. Диаграммы IDEF0: диаграммы декомпозиции, диаграммы дерева узлов, диаграммы только для экспозиции (FEO).
14. Работы (Activity). Стрелки (Arrow). Туннелирование стрелок. Нумерация работ и диаграмм. Каркас диаграммы.
15. Слияние и расщепление моделей.
16. Особенности информационного, программного и технического обеспечения различных видов информационных систем.
17. Экспертные системы. Системы реального времени
18. Оценка экономической эффективности информационной системы.
19. Стоимостная оценка проекта.
20. Классификация типов оценок стоимости: оценка порядка величины, концептуальная оценка, предварительная оценка, окончательная оценка, контрольная оценка.
21. Основные процессы управления проектом.
22. Средства управления проектами
23. Основные понятия качества информационной системы.
24. Национальный стандарт обеспечения качества автоматизированных информационных систем.
25. Международная система стандартизации и сертификации качества продукции.
26. Стандарты группы ISO.
27. Методы контроля качества в информационных системах.
28. Особенности контроля в различных видах систем

## **МДК 05.02. Разработка кода информационных систем**

### **Вопросы к дифференцированному зачету**

1. Структура CASE-средства. Структура среды разработки. Основные возможности.
2. Основные инструменты среды для создания, исполнения и управления информационной системой. Выбор средств обработки информации
3. . Организация работы в команде разработчиков. Система контроля версий: совместимость, установка, настройка
4. Обеспечение кроссплатформенности информационной системы
5. Сервисно - ориентированные архитектуры.
6. . Интегрированные среды разработки для создания независимых программ.
7. Особенности объектно-ориентированных и структурных языков программирования.
8. Разработка сценариев с помощью специализированных языков
9. Общие сведения о системе. Внешний вид интерфейса прикладного решения.
10. Справочники и их виды. Формы документов. Механизм основных форм. Обработчики событий.
11. Обоснование и осуществление выбора модели построения или модификации информационной системы.
12. Обоснование и осуществление выбора средства построения информационной системы и программных средств.
13. Построение архитектуры проекта. Шаблон проекта
14. Определение конфигурации информационной системы. Выбор технических средств.
15. Формирование репозитория проекта, определение уровня доступа в системе контроля версий. Распределение ролей
16. Настройки среды разработки
17. Мониторинг разработки проекта. Сохранение версий проекта
18. Требования к интерфейсу пользователя. Принципы создания графического пользовательского интерфейса (GUI).
19. Понятие спецификации языка программирования. Синтаксис языка программирования. Стиль программирования
20. Основные конструкции выбранного языка программирования. Описание переменных, организация ввода-вывода, реализация типовых алгоритмов
21. Создание сетевого сервера и сетевого клиента.
22. Разработка графического интерфейса пользователя.
23. Отладка приложений. Организация обработки исключений.
24. Виды, цели и уровни интеграции программных модулей.
25. Выбор источников и приемников данных, сопоставление объектов данных.
26. Транспортные протоколы. Стандарты форматирования сообщений.
27. Организация файлового ввода-вывода.
28. Процесс отладки. Отладочные классы.
29. Спецификация настроек типовой ИС.
30. Виды модулей. Контекст модуля формы. Форма как программный объект
31. Процедуры- обработчики событий в модуле формы. Сервер и клиенты.
32. Компиляция общих модулей. Директивы компиляции.
33. Виды регистров накопления. Движения по регистрам накопления

## **МДК 05.03. Тестирование кода информационных систем**

### **Вопросы к дифференцированному зачету**

1. Организация тестирования в команде разработчиков
2. Виды и методы тестирования (в том числе автоматизированные)
3. Тестовые сценарии, тестовые варианты.
4. Разработка тестового сценария проекта

5. Разработка тестовых пакетов
6. Оформление результатов тестирования
7. Инструментарии анализа качества программных продуктов в среде разработке.
8. Обработка исключительных ситуаций.
9. Анализ и обеспечение обработки исключительных ситуаций
10. Методы и способы идентификации сбоев и ошибок.
11. Выявление ошибок системных компонентов.
12. Использование инструментария анализа качества
13. Функциональное тестирование
14. Тестирование безопасности
15. Нагрузочное тестирование, стрессовое тестирование
16. Тестирование интеграции
17. Конфигурационное тестирование
18. Реинжиниринг бизнес-процессов в информационных системах.

### **3. Критерии оценивания.**

#### ***Критерии оценивания подготовки и защиты доклада и презентации по нему***

*Оценка «отлично»* – тема раскрыта в полном объеме, оформление доклада соответствует требованиям, предъявляемым в образовательной организации, доклад грамотный, презентация соответствует всем требованиям.

*Оценка «хорошо»* – незначительные недочеты в оформлении презентации и подготовки доклада.

*Оценка «удовлетворительно»* – незначительные недочеты в оформлении, тема раскрыта, но есть замечания по докладу и презентации.

*Оценка «неудовлетворительно»* – тема не раскрыта, оформление презентации не соответствует требованиям, предъявляемым в образовательной организации, доклад не готов.

#### ***Критерии оценивания выполнения заданий практических занятий***

*Оценка "отлично"* – задание выполнено в полном объеме, даны правильные ответы на контрольные вопросы, сделаны логически точные выводы.

*Оценка "хорошо"* – задание выполнено в полном объеме, даны правильные ответы на контрольные вопросы, не все выводы логически точны и правильны.

*Оценка "удовлетворительно"* – задание выполнено в полном объеме, есть ошибки в ответах на контрольные вопросы, не все выводы правильные.

*Оценка "неудовлетворительно"* – задание не выполнено, ответов нет, выводов нет.

#### ***Критерии оценивания выполнения курсовой работы***

*Оценка «отлично»* – работа выполнена в полном объеме и полностью соответствует техническому заданию, оформление доклада соответствует требованиям, предъявляемым в образовательной организации, доклад грамотный, презентация соответствует всем требованиям.

*Оценка «хорошо»* – работа выполнена в полном объеме, имеются небольшие недоработки по техническому заданию, незначительные недочеты в оформлении презентации и подготовки доклада.

*Оценка «удовлетворительно»* – работа не выполнена в полном объеме, имеются недоработки по техническому заданию, недочеты в оформлении, есть замечания по докладу и презентации.

*Оценка «неудовлетворительно»* – работа выполнена менее чем на 30%, оформление презентации не соответствует требованиям, предъявляемым в образовательной организации, доклад не готов.

### **Критерии оценивания выполнения и защиты проектной работы**

*Оценка «отлично»* – проект выполнен в полном объеме и полностью соответствует техническому заданию, оформление доклада соответствует требованиям, предъявляемым в образовательной организации, доклад грамотный, презентация соответствует всем требованиям.

*Оценка «хорошо»* – проект выполнен в полном объеме, имеются небольшие недоработки по техническому заданию, незначительные недочеты в оформлении презентации и подготовки доклада.

*Оценка «удовлетворительно»* – проект не выполнен в полном объеме, имеются недоработки по техническому заданию, недочеты в оформлении, есть замечания по докладу и презентации.

*Оценка «неудовлетворительно»* – проект выполнен менее чем на 30%, оформление презентации не соответствует требованиям, предъявляемым в образовательной организации, доклад не готов.

### **Критерии оценивания дифференцированного зачета**

Оценка *"отлично"* –

1. Глубокое и прочное усвоение программного материала.
2. Знание пакетов прикладных программ.
3. Знание основных принципов построения пакетов прикладных программ.
4. Знание основных задач прикладных программ.
5. Свободное владение пакетами прикладных программ.
6. Точность и обоснованность выводов.
7. Безошибочное выполнение практического задания.
8. Точные, полные и логичные ответы на дополнительные вопросы.

Оценка *"хорошо"* –

1. Хорошее знание программного материала.
2. Недостаточно полное изложение теоретического вопроса экзаменационного билета.
3. Наличие незначительных неточностей в употреблении терминов, классификаций.
4. Знание основных пакетов прикладных программ.
5. Неполнота представленного иллюстративного материала.
6. Точность и обоснованность выводов.
7. Логичное изложение вопроса, соответствие изложения научному стилю.
8. Негрубая ошибка при выполнении практического задания.

Оценка *"удовлетворительно"* –

1. Поверхностное усвоение программного материала.
2. Недостаточно полное изложение теоретического вопроса экзаменационного билета.
3. Затруднение в приведении примеров, подтверждающих теоретические положения.
4. Наличие неточностей в употреблении терминов, классификаций.
5. Неумение четко сформулировать выводы.
6. Отсутствие навыков научного стиля изложения.
7. Грубая ошибка в практическом задании.
8. Неточные ответы на дополнительные вопросы.

Оценка *"неудовлетворительно"* –

1. Незнание значительной части программного материала.
2. Неспособность привести примеры пакетов прикладных программ
3. Неумение выделить главное, сделать выводы и обобщения.
4. Грубые ошибки при выполнении практического задания.
5. Неправильные ответы на дополнительные вопросы.

### **Критерии оценивания экзамена**

Оценка *"отлично"* –

9. Глубокое и прочное усвоение программного материала.
10. Знание пакетов прикладных программ.
11. Знание основных принципов построения пакетов прикладных программ.
12. Знание основных задач прикладных программ.
13. Свободное владение пакетами прикладных программ.
14. Точность и обоснованность выводов.
15. Безошибочное выполнение практического задания.
16. Точные, полные и логичные ответы на дополнительные вопросы.

Оценка *"хорошо"* –

9. Хорошее знание программного материала.
10. Недостаточно полное изложение теоретического вопроса экзаменационного билета.
11. Наличие незначительных неточностей в употреблении терминов, классификаций.
12. Знание основных пакетов прикладных программ.
13. Неполнота представленного иллюстративного материала.
14. Точность и обоснованность выводов.
15. Логичное изложение вопроса, соответствие изложения научному стилю.
16. Негрубая ошибка при выполнении практического задания.

Оценка *"удовлетворительно"* –

9. Поверхностное усвоение программного материала.
10. Недостаточно полное изложение теоретического вопроса экзаменационного билета.
11. Затруднение в приведении примеров, подтверждающих теоретические положения.
12. Наличие неточностей в употреблении терминов, классификаций.
13. Неумение четко сформулировать выводы.
14. Отсутствие навыков научного стиля изложения.
15. Грубая ошибка в практическом задании.
16. Неточные ответы на дополнительные вопросы.

Оценка *"неудовлетворительно"* –

6. Незнание значительной части программного материала.
7. Неспособность привести примеры пакетов прикладных программ
8. Неумение выделить главное, сделать выводы и обобщения.
9. Грубые ошибки при выполнении практического задания.
10. Неправильные ответы на дополнительные вопросы.