

ПРИЛОЖЕНИЕ

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
ИМЕНИ В.Ф. УТКИНА»

Кафедра «Электронные вычислительные машины»

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ**

**Б1.В.15 «Технологии разработки программного обеспечения и проектирования  
систем ИИ»**

Направление подготовки  
09.03.01 Информатика и вычислительная техника

Профиль  
«Программно-аппаратное обеспечение вычислительных комплексов и систем  
искусственного интеллекта»

Уровень подготовки  
Бакалавриат

Квалификация выпускника – бакалавр

Форма обучения – очная

Рязань 2025

## 1. ОБЩИЕ ПОЛОЖЕНИЯ

Оценочные материалы – это совокупность учебно-методических материалов (контрольных заданий, описаний форм и процедур проверки), предназначенных для оценки качества освоения обучающимися данной дисциплины как части ОПОП.

Цель – оценить соответствие знаний, умений и владений, приобретенных обучающимся в процессе изучения дисциплины, целям и требованиям ОПОП в ходе проведения промежуточной аттестации.

Промежуточный контроль по дисциплине осуществляется путем проведения экзамена. Форма проведения экзамена – билеты с письменным ответом на два теоретических вопроса и одним практическим заданием. При необходимости, проводится устная беседа с обучаемым для уточнения оценки. Выполнение заданий на практических занятиях в течение семестра и заданий на самостоятельную работу является обязательным условием для допуска к экзамену.

## 2. ПАСПОРТ ОЦЕНОЧНЫХ МАТЕРИАЛОВ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

Контролируемые разделы (темы) дисциплины (результаты по разделам)	Код контролируемой компетенции (или её части)	Наименование оценочного средства
<b>Раздел 1.</b> Методологии и процессы разработки программного обеспечения. Базовые концепции и эволюция подходов. Современные парадигмы и инженерные практики в разработке программного обеспечения	ПК-10.1-З ПК-10.1-У ПК-10.1-В	Экзамен
<b>Раздел 2.</b> Инструменты и практики планирования и контроля в управлении ИТ-проектами	ПК-10.1-З ПК-10.1-У ПК-10.1-В	Экзамен
<b>Раздел 3.</b> Методология выявления требований и входных данных для проектирования ПО	ПК-10.1-З ПК-10.1-У ПК-10.1-В	Экзамен
<b>Раздел 4.</b> Проектирование архитектуры системы	ПК-10.1-З ПК-10.1-У ПК-10.1-В	Экзамен
<b>Раздел 5.</b> Современная разработка на платформе .NET с использованием языка С#	ПК-10.1-З ПК-10.1-У ПК-10.1-В	Экзамен
<b>Раздел 6.</b> Тестирование программных продуктов	ПК-10.1-З ПК-10.1-У ПК-10.1-В	Экзамен
<b>Раздел 7.</b> Современные тренды к разработке ПО	ПК-10.2-З ПК-10.2-У ПК-10.2-В	Экзамен

### **3. ОПИСАНИЕ ПОКАЗАТЕЛЕЙ И КРИТЕРИЕВ ОЦЕНИВАНИЯ КОМПЕТЕНЦИЙ**

Сформированность компетенции (или ее части) в рамках освоения данной дисциплины оценивается по трехуровневой шкале:

- 1) Пороговый (базовый) уровень является обязательным для всех обучающихся по завершении освоения дисциплины;
- 2) Продвинутый уровень характеризуется превышением минимальных характеристик сформированности компетенций по завершении освоения дисциплины;
- 3) Эталонный (экспертный) уровень характеризуется освоением компетенций на уровне выше среднего и является важным качественным ориентиром для самосовершенствования.

**Уровень освоения компетенций, формируемых дисциплиной:**

**Описание критериев и шкалы оценивания экзаменационного билета:**

<b>Шкала оценивания</b>	<b>Критерий</b>
5 баллов (эталонный уровень)	выставляется студенту, который дал полные ответы на вопросы, показал глубокие систематизированные знания, смог привести примеры, решил практическую задачу, ответил на дополнительные вопросы преподавателя
4 балла (продвинутый уровень)	выставляется студенту, который дал преимущественно полные ответы на вопросы, решил практическую задачу, но на некоторые дополнительные вопросы преподавателя ответил только с помощью наводящих вопросов
3 балла (пороговый уровень)	выставляется студенту, который дал неполные ответы на вопросы в билете, показал в основном верный ход решения задачи и смог ответить на дополнительные вопросы только с помощью преподавателя
2 балла	выставляется студенту, который не смог ответить на вопросы, а также решить практическую задачу

### **4. ТИПОВЫЕ КОНТРОЛЬНЫЕ ЗАДАНИЯ ИЛИ ИНЫЕ МАТЕРИАЛЫ**

#### **4.1. Промежуточная аттестация**

<b>Коды компетенций</b>	<b>Результаты освоения ОПОП Содержание компетенций (код и содержание индикатора)</b>
<b>ПК-10:</b>	<i>Способен применять знания об истории развития и трендах современного ИИ для формулирования корректных постановок задач и поиска перспективных способов решения проблем с помощью ИИ</i>  ПК-10.1: Позиционирует собственную задачу в заданной области знания с точки зрения трендов современного искусственного интеллекта ПК-10.2: Определяет тенденции развития, оценивает новизну и практическую значимость своих решений с точки зрения современного искусственного интеллекта

***Тестовые вопросы закрытого типа для оценки уровня освоения компетенций:***

**Тема 1. Методологии и процессы разработки программного обеспечения. Базовые концепции и эволюция подходов. Современные парадигмы и инженерные практики в разработке программного обеспечения**

1. Какая модель жизненного цикла является итерационной разновидностью каскадной модели:  
каскадная;  
+с промежуточным контролем;  
спиральная;  
инкрементальная.
2. Какая модель жизненного цикла основана на постепенном наращивании функционала с повторными уточнениями задач:  
каскадная;  
с промежуточным контролем;  
+спиральная;  
инкрементальная.
3. Какую модель жизненного цикла предпочтительнее использовать при большом количестве итераций:  
каскадная;  
с промежуточным контролем;  
спиральная;  
+инкрементальная.
4. Какая модель жизненного цикла подразумевает выполнение проекта без возможности возврата на предыдущие этапы:  
+каскадная;  
с промежуточным контролем;  
спиральная;  
инкрементальная.
5. Какой принцип является ключевым для гибкой методологии разработки?  
Жёсткое следование первоначальному плану  
Полное проектирование системы перед кодированием  
+Регулярная адаптация к изменениям требований  
Чёткое разделение на аналитиков, разработчиков и тестировщиков
6. Что характерно для каскадной (водопадной) модели?  
Итеративное прототипирование  
Параллельное выполнение этапов  
+ Последовательное прохождение этапов  
Разработка короткими спринтами
7. Что такое «спринт» в Scrum?  
Собрание для планирования релиза

Документ с техническим заданием

+ Фиксированный по времени цикл разработки

Отчёт об обнаруженных ошибках

8. Какая модель предполагает создание работающего прототипа на ранних стадиях?

V-образная модель

+ Итеративная модель

Каскадная модель

Сpirальная модель

9. Что из перечисленного НЕ является ценностью Agile-манифеста?

Люди и взаимодействие важнее процессов и инструментов

+ Следование плану важнее, чем реагирование на изменения

Работающее ПО важнее исчерпывающей документации

Сотрудничество с заказчиком важнее согласования условий контракта

10. К какому подходу относится практика парного программирования?

Waterfall

RUP

+ Экстремальное программирование (XP)

Критический путь

11. Технология экстремального программирования предназначена для:

работы в экстремальных условиях;

+ для работы в условиях неопределенных или быстро меняющихся требований;

для сокращения объема программного кода;

для уменьшения времени на разработку технического задания;

для сокращения количества тестовых вариантов.

12. Основными целями технология экстремального программирования являются:  
повышение качества интерфейса;

+ максимальное сокращение сроков выпуска работоспособной программы;

минимизация ошибок;

повышение быстродействия программы;

оптимизация программного кода.

13. Итерации экстремального программирования выполняются на основе:  
вариантов использования;

+ пользовательских историй;

анализа исходного текста программы;

результатов тестирования программы;

анализа быстродействия программы.

## **Тема 2. Инструменты и практики планирования и контроля в управлении ИТ-проектами**

1. Какой инструмент чаще всего используется для визуализации календарного плана проекта?

Диаграмма Ишикавы

+ Диаграмма Гантта

SWOT-анализ

Мозговой штурм

2. Что такое «критический путь» в проекте?

Наиболее затратная часть проекта

Путь с наибольшим количеством задач

+ Последовательность задач, определяющая минимальную длительность проекта

Список самых рискованных задач

3. Какая триада ограничений лежит в основе управления проектами?

Качество, Риск, Стоимость

+ Содержание, Время, Стоимость

Команда, Технологии, Бюджет

План, Факт, Отклонение

4. Какой метод используется для оценки длительности задач на основе оптимистичного, пессимистичного и реалистичного сценариев?

Диаграмма Парето

Метод цепей Маркова

+ Метод PERT

Анализ ROI

5. Что из себя представляет «бэклог продукта» (Product Backlog)?

Список завершённых задач

Отчёт о потраченных ресурсах

+ Приоритизированный список всей работы, которую необходимо сделать

Реестр рисков проекта

6. Основная цель фазы «Мониторинг и контроль» в проекте?

Написание итогового отчёта

+ Сравнение фактических показателей с плановыми и принятие корректирующих действий

Распределение бюджета

Формирование команды

7. Какое собрание в Scrum проводится в начале спринта для отбора задач из бэклога?

Ежедневный стендал

+ Планирование спринта

Обзор спринта

Ретроспектива спринта

8. Что такое «риск» в контексте управления проектами?  
Гарантированная проблема, которая обязательно произойдёт  
+ Неопределённое событие, которое может повлиять на цели проекта  
Отклонение от бюджета более чем на 10%

Конфликт внутри команды

9. Что такое «спринт»?  
+Фиксированный отрезок времени, когда команда разрабатывает и поставляет работающий инкремент продукта

Распределение бюджета

Формирование команды

Написание итогового отчёта

10. Что такое «Agile»?  
+ Гибкий подход к управлению проектами  
Отчёт о потраченных ресурсах

Список самых рискованных задач

Мозговой штурм

### **Тема 3. Методология выявления требований и входных данных для проектирования ПО**

1. Как называется документ, фиксирующий полный набор требований к системе?  
Бизнес-план

План тестирования

- + Техническое задание (ТЗ) / Спецификация требований

Руководство пользователя

2. Какой метод сбора требований предполагает прямое общение с будущими пользователями?

Анализ конкурентов

- + Интервью

Изучение документации

Мозговой штурм

3. Что такое нефункциональное требование?

Требование, которое реализуется не в коде, а в документации

- + Требование, описывающее свойства системы (производительность, безопасность)

Требование, от которого можно отказаться в процессе разработки

Требование, сформулированное не техническим специалистом

4. Какой стандарт в РФ часто используется для оформления Технического задания на ПО?

ISO 9001

- + ГОСТ 34.602-89

IEEE 830

RFC 2119

5. Какой инструмент визуализации лучше всего подходит для описания взаимодействия пользователя с системой?

Диаграмма классов

+ Диаграмма вариантов использования (Use Case)

Диаграмма развёртывания

Диаграмма Гантта

6. Что такое «User Story» (Пользовательская история)?

Подробное техническое описание функции

+ Краткое описание требования с точки зрения конечного пользователя

История изменений в коде

Сценарий тестирования

7. Как называется процесс приоритизации требований по критериям стоимости и важности?

Верификация

Валидация

+ Моделирование MoSCoW

Декомпозиция

8. Что является основной целью фазы анализа требований?

Написание кода

+ Достижение общего понимания того, что должно быть построено

Составление графика проекта

Тестирование системы

9. Какой термин описывает конфликт или несоответствие между различными требованиями?

Синергия

+ Противоречивость требований

Избыточность требований

Инкапсуляция требований

10. Какой метод используется для уточнения требований через создание упрощённой версии системы?

Бенчмаркинг

+ Прототипирование

Реинжиниринг

Рефакторинг

#### **Тема 4. Проектирование архитектуры системы**

1. Диаграмма деятельности предназначена:

для определения условий перехода между состояниями;

для описания внутренних функций состояний;

+ для детализации выполняемых системой действий;

для задания времени нахождения в состоянии;

для задания дорожек ответственности классов.

2. Для описания поведения моделируемой системы используется:

диаграмма вариантов использования;

диаграмма компонентов;

диаграмма кооперации;

+диаграмма состояний;

диаграмма деятельности.

3. Диаграмма компонентов языка UML показывает:

иерархию классов предметной области;

результаты объектной декомпозиции;

+ физическую структуру разрабатываемого программного обеспечения;

функции, выполняемые системой;

потоки данных в системе.

4. Диаграмма развертывания языка UML показывает:

иерархию классов предметной области;

результаты объектной декомпозиции;

аппаратную конфигурацию узлов системы;

+ аппаратную конфигурацию узлов системы с программными компонентами;

функции, выполняемые системой.

5. Для каких целей служит вариант использования на диаграмме вариантов использования:

- представляет класс предметной области;

+ описывает действия, совершаемые системой под воздействием актера;

- представляет объект заданного класса;

- описывает события в системе;

- передает сообщение между объектами системы.

6. Диаграмма классов используется для:

- описания функций системы;

+ представления классов системы и статических связей между ними;

- описания взаимодействия системы с внешними объектами;

- задания сервисов системы для актеров;

- описания последовательности событий в системе.

7. Как на диаграмме последовательности отображается время существования объекта в системе:

- функцией отсчета времени;

- фокусом активности;

+ линией жизни;

- временным интервалом между сообщениями;

- типом объекта.

8. Диаграмма кооперации показывает:

- совокупность объектов предметной области;

+ потоки данных между объектами;

- операции объектов;

- атрибуты объектов;

- наследование объектов.

9. Триггерный переход между состояниями срабатывает:

- при завершении do-деятельности;

- при завершении указанного интервала времени;

+ при наступлении события, внешнего по отношению к исходному состоянию;

- при выполнении заданного условия;

- при совпадении имени внешнего события и внутреннего действия.

10. Нетриггерный переход между состояниями срабатывает:

+ при завершении do-деятельности;

- при завершении указанного интервала времени;

- при наступлении события, внешнего по отношению к исходному состоянию;

- при выполнении заданного условия;

- при совпадении имени внешнего события и внутреннего действия.

## **Тема 5. Современная разработка на платформе .NET с использованием языка C#**

1. Архитектура объектно-ориентированной программы представляет собой:  
набор процедур и функций;

иерархию классов предметной области;

+ множество объектов, обменивающихся сообщениями;

обработчики внутренних и внешних событий;

множество экранных форм.

2. Класс - это:

+ тип данных, включающий описание свойств и функций объектов;

множество объектов с общим состоянием и поведением;

типа данных, включающий описание функций с возможностью их повторного использования;

множество объектов, имеющих общие свойства и функции.

3. Объект - это:

абстрактный тип данных, включающий состояние и поведение;

набор данных и функций работы с ними;

+ экземпляр класса;

типа данных, реализующий функции класса и хранящий его состояние.

4. Свойство системы, позволяющее описать новый класс на основе уже существующего с частично или полностью заимствующейся функциональностью называется:  
инкапсуляцией;  
виртуализацией;  
полиморфизмом;  
+наследованием.

5. Microsoft.Net является:  
+ платформой;

- языком программирования;
- системой управления базами данных;
- прикладной программой.

6. Под CASE – средствами понимают  
+ программные средства, поддерживающие процессы создания и сопровождения программного обеспечения;  
– языки программирования высокого уровня;  
– среды для разработки программного обеспечения;  
– прикладные программы.

7. Традиционным методом организации информационных систем является:  
– клиент-клиент;  
+ клиент-сервер;  
– сервер-сервер;  
– размещение всей информации на одном компьютере.

8. Что такое CLR (Common Language Runtime) в .NET?  
- Библиотека стандартных классов

- Среда разработки  
+ Исполнительная среда, управляющая выполнением кода  
- Система управления пакетами

9. Какой модификатор доступа является наиболее ограничительным в C#?  
- public

-protected  
-internal  
+ private

10. Какой модификатор доступа является наименее ограничительным в C#?  
+ public

-protected  
-internal  
-private

11. Технология визуального программирования это:  
способ создания экранных форм;

+способ создания программы путём манипулирования графическими объектами вместо написания её текста;  
построение графических образов;  
технология улучшения визуального качества изображений.

## **Тема 6. Тестирование программных продуктов**

1. Какой вид тестирования проверяет отдельный модуль или функцию?  
Системное тестирование

Интеграционное тестирование

+ Модульное (юнит) тестирование

Приёмочное тестирование

2. Что такое тестирование «чёрного ящика»?

Тестирование, основанное на знании внутренней структуры кода

+ Тестирование, основанное на спецификациях и требованиях

Тестирование производительности системы

Автоматизированное тестирование интерфейса

3. Какой артефакт содержит шаги для проверки конкретной функции?

План тестирования

Отчёт об ошибке

+ Тест-кейс

Чек-лист

4. Что проверяет регрессионное тестирование?

Новый функционал впервые

+ Что новые изменения не сломали существующий функционал

Соответствие системы нефункциональным требованиям

Интеграцию всех модулей

5. Какой уровень тестирования проверяет взаимодействие между несколькими модулями?

Модульное

+ Интеграционное

Системное

Альфа-тестирование

6. Как называется документ, описывающий общий подход, цели и объём тестирования?

Тест-кейс

+ План тестирования

Отчёт о тестировании

Спецификация требований

7. Что такое «баг-репорт»?

План по исправлению ошибок

+ Документ, описывающий обнаруженный дефект

График количества найденных ошибок

Ежедневный отчёт тестировщика

8. Какой тип тестирования проводится силами самой команды разработки?

Бета-тестирование

+ Альфа-тестирование

Приёмочное тестирование

Пользовательское тестирование

9. Какой принцип тестирования гласит, что исчерпывающее тестирование невозможно?

Принцип раннего тестирования

+ Парадокс пестицида

Кластеризация дефектов

Тестирование зависит от контекста

10. Что проверяет нагрузочное тестирование (Load Testing)?

Логику работы приложения

+ Поведение системы под ожидаемой нагрузкой

Удобство использования интерфейса

Безопасность системы

11. Тестирование белого ящика предназначено:

для проверки правильности выполняемых функций программы;

+ для проверки внутренней логики программы;

для проверки допустимых комбинаций входных данных;

для проверки допустимых комбинаций выходных данных.

12. Как представляется программа при тестировании базового пути:

множеством объектов;

списком операторов;

+ потоковым графом;

списками входных и выходных параметров;

командами языка ассемблера.

13. Количество регионов потокового графа показывает:

+ количество тестовых вариантов в способе тестирования базового пути;

количество циклов в программе;

количество модулей программы;

количество переходов в программе;

количество входных данных программы.

14. Тестирование черного ящика предназначено:

для проверки допустимых комбинаций входных данных;

для проверки допустимых комбинаций выходных данных;  
+ исследования функции программы на всей области определения;  
определения экстремумов функции;  
проверки точности вычисления функций программы.

## Тема 7. Современные тренды в разработке ПО

1. Какой подход объединяет разработку и эксплуатацию для ускорения выпуска обновлений?

Agile

+ DevOps

Waterfall

Lean

2. Что такое «Контейнеризация»?

Виртуализация на уровне операционной системы

+ Упаковка приложения со всеми зависимостями в изолированный контейнер

Разделение кода на независимые библиотеки

Способ хранения данных

3. Какой тренд предполагает использование облачных сервисов по требованию?

a) Микросервисы

+ «Как услуга» (XaaS: IaaS, PaaS, SaaS)

Непрерывная интеграция

Low-Code разработка

4. Для чего используется инструмент Docker?

Для управления виртуальными машинами

+ Для создания и управления контейнерами

Для оркестрации кластеров

Для мониторинга приложений

5. Что такое «Low-Code/No-Code платформы»?

Платформы для разработки низкоуровневого кода

+ Платформы, позволяющие создавать приложения с минимальным ручным кодированием

Фреймворки с открытым исходным кодом

Языки программирования с упрощённым синтаксисом

6. Какой практикой является «Инфраструктура как код» (IaC)?

Ручная настройка серверов

+ Управление и настройка инфраструктуры через конфигурационные файлы

Написание кода для аппаратного обеспечения

Использование шаблонов проектирования

7. Что является ключевой идеей «Serverless-архитектуры»?

Отказ от использования серверов

+ Разработчик сосредотачивается на коде, а облачный провайдер управляет серверами

Использование только физических серверов

Работа без подключения к сети

8. Какой подход делает безопасность неотъемлемой частью жизненного цикла разработки?

Agile Security

+ DevSecOps

Secure Coding

Penetration Testing

9. Что изучает «Data Science» в контексте разработки ПО?

Проектирование баз данных

+ Извлечение знаний и insights из данных

Написание SQL-запросов

Шифрование информации

10. Какой тренд связан с созданием цифровых двойников (Digital Twins) физических объектов?

Виртуальная реальность

+ Интернет вещей (IoT)

Блокчейн

Квантовые вычисления

***Тестовые вопросы открытого типа для оценки уровня освоения компетенций:***

**Тема 1. Методологии и процессы разработки ПО**

1. Как называется подход, где разработка ведётся короткими циклами? (Agile)
2. Как называется основной документ с требованиями в каскадной модели? (Техническое задание)
3. Как называется собрание в начале спринта в Scrum? (Планирование спринта)
4. Какая модель похожа на водопад, но с возвратами на предыдущие этапы? (Итеративная)
5. Как называется практика частого слияния кода в общую ветку? (Непрерывная интеграция)
6. Как называется список всех задач проекта в Agile? (Бэклог продукта)
7. Как называется встреча для обсуждения итогов спринта? (Ретроспектива)
8. Какая роль в Scrum отвечает за удаление препятствий? (Скрам-мастер)
9. Какой принцип Agile говорит, что важнее рабочий продукт? (Работающее ПО)
10. Как называется цикл разработки в Scrum? (Спринт)

**Тема 2. Управление ИТ-проектами**

1. Как называется график задач в виде столбцов? (Диаграмма Ганнта)
2. Как называется самая длинная цепочка задач в проекте? (Критический путь)
3. Как называется список задач на текущий спринт? (Бэклог спринта)
4. Что измеряет CPI (Cost Performance Index)? (Эффективность затрат)
5. Какая триада определяет основные ограничения проекта? (Содержание, время, стоимость)
6. Как называется событие, отмечающее ключевое завершение этапа? (Веха)
7. Как называется неопределённое событие, влияющее на проект? (Риск)
8. Какой метод оценивает длительность задач по трём сценариям? (PERT)
9. Как называется документ, разрешающий начало проекта? (Устав проекта)
10. Как называется процесс сравнения плана с фактом? (Контроль)

**Тема 3. Выявление требований и входные данные**

1. Как называется основной документ с требованиями к системе? (Техническое задание)
2. Какой метод сбора требований предполагает личную беседу? (Интервью)
3. Как называются требования к скорости и надёжности системы? (Нефункциональные)
4. Какой стандарт в РФ часто используют для ТЗ на ПО? (ГОСТ 34)
5. Как называется краткое описание функции с точки зрения пользователя? (Пользовательская история)
6. Как называется диаграмма взаимодействия пользователя с системой? (Вариантов использования)
7. Как называется метод приоритизации Must Have, Should Have? (MoSCoW)
8. Как называется упрощённая версия системы для уточнения требований? (Прототип)
9. Как называется конфликт между разными требованиями? (Противоречивость)
10. Кто является основным источником требований? (Заказчик/Пользователь)

## Тема 4. Проектирование архитектуры системы

1. Как называется язык для графического моделирования? (UML)
2. Какая диаграмма показывает классы и их связи? (Диаграмма классов)
3. Как называется принцип «один класс — одна ответственность»? (SRP)
4. Какой шаблон гарантирует один экземпляр класса? (Одиночка / Singleton)
5. Какая диаграмма показывает порядок вызовов методов? (Последовательностей)
6. Как называется архитектура из независимых сервисов? (Микросервисная)
7. Как называется минимальная зависимость между модулями? (Слабая связанность)
8. Какая диаграмма показывает размещение компонентов на серверах? (Развёртывания)
9. Как называется общий план построения системы? (Архитектура)
10. Какой шаблон создаёт объекты через общий интерфейс? (Фабрика)

## Тема 5. Разработка на .NET с использованием C#

1. Как называется среда выполнения кода .NET? (CLR)
2. Какой оператор обрамляет код, где может быть ошибка? (try)
3. Какое ключевое слово делает член общим для всех объектов класса? (static)
4. Как называется динамический массив в C#? (List<T>)
5. Какой модификатор доступа самый открытый? (public)
6. Как называется технология запросов к данным в C#? (LINQ)
7. Как называется файл .dll или .exe в .NET? (Сборка / Assembly)
8. Какой оператор безопасно приводит типы? (as)
9. Какой атрибут отмечает точку входа в программу? ([STAThread])
10. Какой интерфейс нужно реализовать для освобождения ресурсов? (IDisposable)

## Тема 6. Тестирование программных продуктов

1. Как называется проверка отдельного метода? (Модульное тестирование)
2. Какой вид тестирования не учитывает внутреннее устройство? (Чёрный ящик)
3. Как называется сценарий для проверки функции? (Тест-кейс)
4. Что проверяют, чтобы новое не сломало старое? (Регрессия)
5. Какой уровень проверяет взаимодействие модулей? (Интеграционный)
6. Как называется документ с общим подходом к тестированию? (План тестирования)
7. Как называется отчёт об ошибке? (Баг-репорт)
8. Какое тестирование имитирует работу пользователя? (Приёмочное / UAT)
9. Какой принцип гласит, что все ошибки найти невозможно? (Исчерпывающее тестирование)
10. Какое тестирование проверяет скорость работы? (Нагрузочное)

## Тема 7. Современные тренды в разработке ПО

1. Какой подход объединяет разработку и эксплуатацию? (DevOps)
2. Как называется упаковка приложения со всеми зависимостями? (Контейнеризация)
3. Какой инструмент используется для работы с контейнерами? (Docker)
4. Как называются небольшие независимые сервисы? (Микросервисы)

5. Как называется модель, где провайдер управляет серверами? (Serverless)
6. Как называется практика управления инфраструктурой через код? (IaC)
7. Какой фреймворк часто используют для SPA? (Angular / React / Vue)
8. Какой тренд использует машинное обучение в разработке? (AI/ML)
9. Как называются платформы с визуальным программированием? (Low-Code)
10. Какой подход встраивает безопасность в процесс разработки? (DevSecOps)

#### **Типовые теоретические вопросы:**

1. Понятие жизненного цикла ИС.
2. Жизненный цикл и этапы разработки ПО.
3. Стандартизация процессов жизненного цикла ИС.
4. Виды процессов жизненного цикла.
5. Эволюция моделей жизненного цикла ПО.
6. Эволюция моделей жизненного цикла ПО. Каскадная модель ЖЦ
7. Эволюция моделей жизненного цикла ПО. Модель ЖЦ с промежуточным контролем.
8. Эволюция моделей жизненного цикла ПО. Спиральная модель ЖЦ ПО.
9. Эволюция моделей жизненного цикла ПО. Инкрементальная модель.
10. Выбор процесса жизненного цикла процесса разработки ПО.
11. Построение моделей ИС
12. Классификация моделей разрабатываемых ИС.
13. Язык UML. Диаграммы вариантов использования.
14. Язык UML. Диаграммы классов.
15. Язык UML. Отношения между классами.
16. Язык UML. Диаграммы состояний.
17. Язык UML. Диаграммы деятельности.
18. Язык UML. Диаграммы взаимодействия.
19. Язык UML. Диаграммы компонентов.
20. Язык UML. Диаграммы размещения.
21. Проблемы разработки сложных ИС.
22. Технология экстремального программирования.
23. Технология RUP.
24. Технология визуального программирования.

## **Типовые практические задания:**

### **Задание 1**

В соответствии с вариантом разработать для заданной предметной области диаграмму вариантов использования и описательную спецификацию.

### **Критерии выполнения задания 1**

Задание считается выполненным, если обучающийся определил варианты использования создаваемой системы и отношения между ними, а также назначил актеров, определяющих внешние подсистемы по отношению к создаваемой.

### **Задание 2**

В соответствии с вариантом разработать для заданной предметной области диаграмму классов.

### **Критерии выполнения задания 2**

Задание считается выполненным, если обучающийся выделил части предметной области задачи и представил каждую из них соответствующим классом с необходимыми атрибутами и операциями, а также определил отношения в иерархии классов.

### **Задание 3**

В соответствии с вариантом разработать для заданной предметной области диаграмму классов и последовательностей.

### **Критерии выполнения задания 3**

Задание считается выполненным, если обучающийся определил сообщения необходимых типов между экземплярами классов системы и представил результаты в графическом виде на языке UML.

### **Задание 4**

В соответствии с вариантом разработать для заданной предметной области диаграмму классов и кооперации.

### **Критерии выполнения задания 4**

Задание считается выполненным, если обучающийся определил потоки данных и представил результаты в графическом виде на языке UML.

### **Задание 5**

В соответствии с вариантом разработать для заданной предметной области диаграмму компонентов.

### **Критерии выполнения задания 5**

Задание считается выполненным, если обучающийся выделил типы программных компонент создаваемой системы, взаимодействие между ними и представил результаты в графическом виде на языке UML.

### **Задание 6**

В соответствии с вариантом разработать для заданной предметной области диаграмму размещения.

### **Критерии выполнения задания 6**

Задание считается выполненным, если обучающийся выделил узлы, соответствующие аппаратной конфигурации создаваемой системы, каналы их взаимодействия и привязал программные компоненты к соответствующим узлам.

### **Задание 7**

В соответствии с вариантом разработать для заданной предметной области диаграмму состояния для всей системы.

### **Критерии выполнения задания 7**

Задание считается выполненным, если обучающийся выделил состояния создаваемой системы, события, вызывающие переходы между ними и представил результаты в графическом виде на языке UML.

### **Задание 8**

В соответствии с вариантом разработать для заданной предметной области диаграмму деятельности для всей системы.

### **Критерии выполнения задания 8**

Задание считается выполненным, если обучающийся выделил виды деятельности создаваемой системы и их взаимодействия, и представил результаты в графическом виде на языке UML.

## **Варианты предметных областей для практических заданий 1-8**

### **Вариант 1**

Система поддержки управления библиотекой должна обеспечивать операции (добавление, удаление и изменение) над данными о читателях. В регистрационном списке читателей хранятся следующие сведения: фамилия, имя и отчество читателя; номер его читательского билета и дата выдачи билета. Наряду с регистрационным списком системой должен поддерживаться каталог библиотеки, где хранится информация о книгах: название, список авторов, библиотечный шифр, год и место издания, название издательства, общее количество экземпляров книги в библиотеке и количество экземпляров, доступных в текущий момент. Система обеспечивает добавление, удаление и изменение данных каталога, а также поиск книг в каталоге на основании введенного шифра или названия книги. В системе осуществляется регистрация взятых и возвращенных читателем книг. Про каждую выданную книгу хранится запись о том, кому и когда была выдана книга, и когда она будет возвращена. При возврате книги в записи делается соответствующая пометка, а сама запись не удаляется из системы. Система должна выдавать следующую справочную информацию:

- какие книги были выданы за данный промежуток времени;
- какие книги были возвращены за данный промежуток времени;
- какие книги находятся у данного читателя;
- имеется ли в наличии некоторая книга.

Вариант задания предусматривает разработку схемы базы данных, хранящей список читателей, каталог книг и записи о выдаче книг.

### **Вариант 2**

Интернет-магазин позволяет делать покупки с доставкой на дом. Клиенты магазина при помощи программы-браузера имеют доступ к каталогу продаваемых товаров, поддержку которого осуществляет Интернет-магазин. В каталоге товары распределены по разделам. О каждом товаре доступна полная информация (название, вес, цена, изображение, дата изготовления и срок годности). Для удобства клиентов предусмотрена система поиска товаров

в каталоге. Заполнение каталога информацией происходит автоматически вначале рабочего дня, информация берется из системы автоматизации торговли.

При отборе клиентами товаров поддерживается виртуальная «торговая корзина». Любое наименование товара может быть добавлено в «корзину» или изъято в любой момент по желанию покупателя с последующим пересчетом общей стоимости покупки. Текущее содержимое «корзины» постоянно показывается клиенту.

По окончании выбора товаров производится оформление заказа и регистрация покупателя. Клиент указывает в регистрационной форме свою фамилию, имя и отчество, адрес доставки заказа и телефон, по которому с ним можно связаться для подтверждения сделанного заказа. Заказы передаются для обработки в систему автоматизации торговли. Проверка наличия товаров на складе и их резервирование Интернет-магазином не производятся. Дополнительно требуется разработать схему базы данных, хранящей заказы.

При выполнении этого варианта задания рекомендуем ознакомиться с работой [Коналлен-2001]. Следует определиться, по какому архитектурному шаблону будет строиться Web-приложение («тонкий клиент» или «толстый клиент»). В соответствии с выбранным шаблоном следует построить модели клиентской части магазина и серверной части, промоделировать связи между частями приложения. Для Web-приложений типичными являются следующие классы:

- клиентская Web-страница;
- серверная Web-страница (например, CGI-скрипт);
- HTML-форма;
- объект JavaScript.

Дополнительные связи между классами Web-приложений:

- link - ссылка с одной страницы на другую;
- build - связь между CGI-скриптом и клиентской страницей, генерируемой при его выполнении;
- submit - связь между формой и серверной Web-страницей, принимающей данные из формы.

Типичные компоненты:

- Web-страница (HTML-файл),
- Active Server Page (ASP),
- Java Server Page (JSP),
- сервлет,
- библиотека скриптов (например, подключаемый файл с Javascript-функциями).

### **Вариант 3**

WWW-конференция представляет собой хранилище сообщений в сети Интернет, доступ к которому осуществляется при помощи браузера. Для каждого сообщения конференции хранятся значения следующих полей: номер сообщения, автор, тема, текст сообщения, дата добавления сообщения, ссылка на родительское сообщение. Начальной страницей конференции является иерархический список сообщений. Верхний уровень иерархии составляют сообщения, открывающие новые темы, а подуровни составляют сообщения, полученные в ответ на сообщения верхнего уровня. Сообщение-ответ всегда имеет ссылку на исходное сообщение. В списке отображаются только темы сообщений, их авторы и даты добавления. Просматривая список, пользователь выбирает сообщение и по гиперссылке открывает страницу с текстом сообщения. Помимо текста на этой странице отображается список (иерархический) сообщений являющихся ответами, ответами на ответы и т.д. Для удобства пользователей необходимо предусмотреть поиск сообщений по автору или по ключевым словам в теме или тексте сообщения.

Сообщения добавляются в конференцию зарегистрированными пользователями, которые при отправке сообщения должны указать своё имя и пароль. Регистрирует новых пользователей модератор конференции - её ведущий. При регистрации пользователь

заполняет специальную форму, содержимое которой затем пересыпается модератору и запоминается в базе пользователей. Модератор решает, регистрировать пользователя или нет, и отправляет свой ответ.

При добавлении сообщений пользователь имеет возможность начать новую тему или ответить на ранее добавленные сообщения. После добавления сообщения оно доступно для чтения всем пользователям (даже незарегистрированным), и список сообщений обновляется.

Модератор имеет право по тем или иным причинам удалять сообщения любых авторов. Он также может наказывать пользователей, нарушающих правила поведения в конференции, лишая на некоторое время пользователя возможности добавлять и редактировать сообщения.

Вариант задания включает в себя разработку схемы базы данных для хранения сообщений конференции и информации об её участниках.

Выполняющим это задание полезно ознакомиться с заключительным замечанием к варианту «Интернет-магазин». Наиболее подходящей архитектурой для WWW-конференции является «тонкий клиент», поскольку клиентская часть практически не содержит «бизнес-логики». Единственным её элементом, который может выполняться на стороне клиента, является проверка правильного заполнения полей формы, перед отправкой её содержимого на сервер.

#### **Вариант 4**

В каталоге хранится следующая информация о ресурсах: название ресурса, уникальный локатор ресурса (URL), раздел каталога, в котором содержится ресурс, список ключевых слов, краткое описание, дата последнего обновления, контактная информация.

Доступ пользователей к каталогу осуществляется при помощи браузера. Пользователи каталога могут добавлять новые ресурсы, информация о которых не была внесена ранее. Ресурсы в каталоге классифицируются по разделам. Полный список ресурсов каждого раздела должен быть доступен пользователям. Пользователям каталога должны быть предоставлены возможности по поиску ресурсов. Поиск осуществляется по ключевым словам. Если пользователь не доволен результатами поиска, он может уточнить запрос (осуществить поиск среди результатов предыдущего поиска). Должна быть возможность выдавать результаты поиска в разной форме (вывод всей информации о ресурсах или частичной). Пользователь может отсортировать список ресурсов по релевантности (соответствию ключевым словам из запроса) или по дате обновления.

Поскольку содержание ресурсов Интернет со временем изменяется необходимо следить за датой последнего обновления, периодически опрашивая Web-сайты, URL которых хранятся в каталоге.

Вариант задания включает в себя разработку схемы базы данных для хранения сообщений конференции и информации об её участниках.

Выполняющим это задание полезно ознакомиться с заключительным замечанием к варианту «Интернет-магазин». Как и в варианте «WWW-конференция» самой подходящей архитектурой для каталога является «тонкий клиент», поскольку клиентская часть практически не включает в себя функций «бизнес-логики» кроме проверки содержимого форм перед пересылкой на сервер.

#### **Вариант 5**

На экране будильника постоянно отображается текущее время (часы и минуты, например: 12 : 00), двоеточие между числом часов и числом минут зажигается и гаснет с интервалом в полсекунды.

Управление будильником осуществляется следующими кнопками:

- кнопкой режима установки времени,
- кнопкой режима установки времени срабатывания,
- двумя отдельными кнопками для установки часов и минут,

- кнопкой сброса сигнала «СБРОС».

На будильнике имеется переключатель режима работы со следующими положениями: «ВЫКЛ», «ВКЛ», «РАДИО» и «ТАЙМЕР».

Для установки текущего времени нужно нажать на кнопку режима установки и, при нажатой кнопке, нажимать на кнопки установки часов и минут. При каждом нажатии на кнопки, устанавливаемое значение увеличивается на одну единицу (один час или одну минуту соответственно). При достижении максимального значения производится сброс. Для установки времени срабатывания будильника нужно нажать на кнопку режима установки времени срабатывания и, держа кнопку нажатой, нажимать на кнопки установки часов и минут. Когда переключатель режима работы находится в положении «ВКЛ», при достижении времени срабатывания происходит подача звукового сигнала в течение одной минуты. Сигнал можно прервать, нажав на кнопку «СБРОС». При этом сигнал должен быть возобновлен через пять минут. При установке переключателя в положение «ВЫКЛ» звуковой сигнал не подается.

Когда переключатель находится в положении «РАДИО» работает радиоприемник. При переводе переключателя в положение «ТАЙМЕР» включается радиоприемник на тридцать минут, а затем часы переходят в состояние будильника (аналогично положению «ВКЛ»). При нажатии на кнопку режима установки времени, будильник должен отображать время срабатывания.

### **Вариант 6**

Система хранит сведения о персонах (Ф.И.О., пол, дата рождения, дата смерти, биография) и о родственных связях между ними. Связи бывают только трех видов: «мужья-женены», «дети-родители» и «братья-сестры». Система обеспечивает возможность добавления данных о новых персонах и родственных связях, изменение введенных данных и удаление ненужных данных. Система следит за непротиворечивостью вводимых данных. Например, недопустимо, чтобы человек был собственным предком или потомком.

Разработанная модель должна содержать схему базы данных для хранения генеалогических деревьев.

Пользователи системы могут осуществлять поиск полезной информации по дереву:

- находить для указанного члена семьи его детей;
- находить для указанного члена семьи его родителей;
- находить для указанной персоны братьев и сестер, если таковые есть;
- получать список всех предков персоны;
- получать список всех потомков персоны;
- получать список всех родственников персоны;
- прослеживать цепочку родственных связей от одной персоны до другой (например, если Петр является шурином Ивана, то на запрос о родственных связях между Петром и Иваном выдается такой результат: «Петр - брат Ольги, Ольга - жена Ивана»).

### **Вариант 7**

В телевизоре имеются следующие устройства: приемник телевизионного сигнала, устройство отображения картинки, память каналов, память настроек, управляющие кнопки, пульт дистанционного управления (ДУ). Управление телевизором осуществляется при помощи кнопок на корпусе (их четыре: «ВКЛ/ВЫКЛ», «-», «+», кнопка начальной установки) и пульта ДУ. Кнопка «ВКЛ/ВЫКЛ» позволяет включать и выключать телевизор. После включения телевизора на экран отображается передача, идущая по каналу №1, при этом используются параметры изображения и значение громкости, сохраненные в памяти настроек.

Память каналов телевизора хранит до 60 каналов. Каналы нумеруются, начиная с нуля. Последовательное переключение каналов осуществляется при помощи кнопок «-» и «+». Нажатие на «+» переключает телевизор на канал с номером, на единицу большим (с 59-го

канала телевизор переключается на 0-ой). Нажатие на «-» переключает телевизор на канал с номером, на единицу меньшим (с 0-го канала телевизор переключается на 59-ый).

При нажатии на кнопку начальной установки очищается память каналов телевизора, после чего осуществляется поиск передач и сохранение их частот в памяти каналов. Поиск начинается с нижней границы рабочего диапазона телевизора. На экран телевизора выводится «синий экран». Рабочая частота постепенно увеличивается до тех пор, пока приемник не обнаружит телевизионный сигнал. Найденная передача выводится на экран в течение 10 секунд. Также отображается номер, под которым найденный канал будет сохранен в памяти (начиная с 1).

## **Типовые практические задания:**

**Задание 1.** Задана программа на псевдокоде, которую необходимо протестировать с помощью способа тестирования базового пути. При этом следует выполнить следующие шаги:

1. Построить потоковый граф программы.
2. Вычислить цикломатическую сложность тремя разными способами.
3. Определить базовое множество независимых путей графа.
4. Разработать тестовые варианты для каждого пути.

### ***Критерии выполнения задания 1***

Задание считается выполненным, если обучающийся правильно построил потоковый граф программы, разработал тестовые варианты для каждого пути из базового множества независимых путей графа.

#### **Варианты для задания 1**

##### **Вариант 1**

С помощью способа тестирования базового пути разработать тестовые варианты для приведенной ниже программы на псевдокоде:

```
{процедура «Снять деньги со счета через банкомат»}  
если (банкомат готов)  
    то нач  
        вставить карту  
        ввести PIN-код  
        если (время ожидания < 5 минут) и (PIN-код правильный)  
            то нач  
                ввести сумму  
                если (денег на счете достаточно)  
                    то выдать деньги  
                иначе выдать сообщение «Денег недостаточно»  
            кон  
        иначе выдать сообщение «Время вышло или PIN-код  
                                неправильный»  
    кон
```

##### **Вариант 2**

```
{процедура «Выполнение операций с банкоматом»}  
читать карту  
пока (не «Выход из меню»)  
    нач  
        запросить PIN-код  
        если (карта принадлежит банку) и (PIN-код правильный)  
            то нач  
                ввести код операции  
                если (операция = «Снять деньги со счета»)  
                    то выдать требуемую сумму  
                иначе  
                    если (операция = «Просмотреть баланс»)  
                        то вывести баланс счета
```

кон

иначе выдать сообщение «Карта не принадлежит банку  
или PIN-код неправильный»

кон

### **Вариант 3**

{процедура «Продажа товара через торговый автомат»}

если (торговый аппарат свободен)

то

нач

внести сумму

выбрать товар

если (время ожидания не вышло)

то

если (товар в наличии) и (денег достаточно)

то выдать товар

иначе выдать сообщение «Нет товара или денег  
недостаточно»

иначе

нач

выдать сообщение «Товар не выбран»

вернуть деньги

кон

кон

### **Вариант 4**

{процедура «Оплата мобильной связи через банкомат»}

если (банкомат свободен)

то

нач

вставить карту

ввести номер телефона

если (время ожидания < 5 минут) и (номер существует)

то

нач

ввести сумму оплаты

если (денег на карте достаточно)

то оплатить мобильную связь

иначе выдать сообщение «Денег недостаточно»

кон

иначе выдать сообщение «Время вышло

или номер неправильный»

кон

**Задание 2.** Необходимо протестировать программу с помощью способа разбиения на классы эквивалентности. При этом следует выполнить следующие шаги:

1. По каждому входному параметру определить допустимые и недопустимые классы эквивалентности и представить их в виде таблицы.
2. Построить дерево разбиений.
3. Разработать тестовые варианты для каждого конечного листа дерева.

***Критерии выполнения задания 2***

Задание считается выполненным, если обучающийся правильно определил допустимые и недопустимые классы эквивалентности по каждому входному параметру и разработал для них тестовые варианты.

**Задание 3.** Необходимо протестировать программу с помощью способа анализа граничных значений. Для этого требуется:

1. По каждому параметру выявить граничные значения.
2. Разработать тестовые варианты, проверяющие параметры на каждой границе.

***Критерии выполнения задания 3***

Задание считается выполненным, если обучающийся правильно определил граничные значения по каждому входному параметру и разработал для них тестовые варианты.

**Задание 4.** Необходимо протестировать программу с помощью способа анализа причинно-следственных связей. Для этого требуется:

1. Выявить причины и следствия.
2. Построить логические выражения, определяющие каждое следствие через причины.
3. Разработать граф причинно-следственных связей.
4. Построить таблицу решений.
5. Для каждого столбца таблицы решений разработать отдельный тестовый вариант.

***Критерии выполнения задания 3***

Задание считается выполненным, если обучающийся правильно определил причины и следствия, разработал граф причинно-следственных связей, построил таблицу решений и разработал все тестовые варианты.

**Варианты для задания 2-4**

**Вариант 1**

Требуется протестировать программу начисления стипендии студентам младших курсов. Средний балл может принимать значения в диапазоне от 12 до 20. Известно, что стипендия определяется согласно следующей таблице:

Курс	Средний балл	Стипендия
1	$\geq 14$ $< 14$	Повышенная Обычная
2	$\geq 16$ $< 16$	Повышенная Обычная
3	$\geq 18$ $< 18$	Повышенная Обычная

### **Вариант 2**

Требуется протестировать программу оценивания знаний студентов 1, 2, 3-го курса по дисциплине «История». Известно, что результат определяется согласно следующей таблице:

Курс	% решенных заданий	Результат
1	[90-100] [0-90)	Удовлетворительно Неудовлетворительно
2	[50-100] [0-50)	Удовлетворительно Неудовлетворительно
3	[30-100] [0-30)	Удовлетворительно Неудовлетворительно

### **Вариант 3**

Требуется протестировать программу начисления премии сотрудникам банка в зависимости от стажа работы и должности. Стаж может принимать значения в диапазоне от 1 до 25.

Должность	Стаж	Премия
Администрация	$\geq 5$	20 %
	<5	10 %
Менеджеры	$\geq 10$	20 %
	<10	10 %
Операторы	$\geq 20$	20 %
	<20	10 %

### **Вариант 4**

Требуется протестировать программу расчета транспортного налога в зависимости от типа автомобиля и мощности двигателя. Известно, что результат определяется согласно следующей таблице:

Тип машины	Мощность двигателя, л.с.	Налог, руб.
Легковая	$\geq 100$	5000
	<100	1000
Грузовая	$\geq 150$	10000
	<150	5000

### **Задание 5**

В соответствии с вариантом задания необходимо: задать все основные сведения о проекте, определить список основных задач и составляющих их подзадач, требуемых для успешного выполнения проекта.

Задание должно быть реализовано с помощью программного средства Microsoft Project.

### ***Критерии выполнения задания 5***

Задание считается выполненным, если обучающийся разработал файл проекта с расширением \*.mpp, используя программу Microsoft Project, при этом верно определил все сведения о проекте, правильно перечислил основные задачи проекта и выделил подзадачи.

## **Задание 6**

В соответствии с вариантом задания необходимо: задать все основные сведения о проекте, определить список и параметры задач всех уровней, требуемых для успешного выполнения проекта, структурировать список задач, задать вехи.

Задание должно быть реализовано с помощью программного средства Microsoft Project.

### ***Критерии выполнения задания 6***

Задание считается выполненным, если обучающийся разработал файл проекта с расширением \*.mpp, используя программу Microsoft Project, верно определил все основные сведения о проекте, список и параметры задач всех уровней, требуемых для успешного выполнения проекта, правильно структурировал список задач и задал вехи.

## **Задание 7**

В соответствии с вариантом задания необходимо: задать пул ресурсов, определить конкретные материальные и трудовые ресурсы, требуемые для успешного завершения проекта, задать ручное выравнивание ресурсов.

Задание должно быть реализовано с помощью программного средства Microsoft Project.

### ***Критерии выполнения задания 7***

Задание считается выполненным, если обучающийся разработал файл проекта с расширением \*.mpp, используя программу Microsoft Project, правильно определил пул ресурсов, конкретные материальные и трудовые ресурсы, требуемые для успешного завершения проекта, задал ручное выравнивание ресурсов.

## **Задание 8**

В соответствии с вариантом задания необходимо: выполнить назначение ресурсов задачам проекта; определить стандартные ставки для ресурсов проекта, определить стоимость каждого из ресурсов; определить стоимости отдельных задач и проекта в целом.

Задание должно быть реализовано с помощью программного средства Microsoft Project.

### ***Критерии выполнения задания 8***

Задание считается выполненным, если обучающийся разработал файл проекта с расширением \*.mpp, используя программу Microsoft Project, правильно выполнил назначение ресурсов задачам проекта; верно определил стандартные ставки для ресурсов проекта, стоимость каждого из ресурсов и стоимости отдельных задач и проекта в целом.

## **Задание 9**

В соответствии с вариантом задания необходимо: определить стандартные ставки для ресурсов проекта, определить стоимость каждого из ресурсов; определить стоимости отдельных задач и проекта в целом; выполнить анализ и выравнивание загрузки ресурсов.

Задание должно быть реализовано с помощью программного средства Microsoft Project.

### ***Критерии выполнения задания 9***

Задание считается выполненным, если обучающийся разработал файл проекта с расширением \*.mpp, используя программу Microsoft Project, правильно определил

стандартные ставки для ресурсов проекта, стоимость каждого из ресурсов, стоимости отдельных задач и проекта в целом, выполнил анализ и выравнивание загрузки ресурсов.

#### ***Варианты предметных областей для заданий 5-10.***

##### ***Вариант 1. Разработка системы для учета заграничных командировок в государственной компании с высоким уровнем секретности***

Государственной кампании с высоким уровнем секретности требуется разработать систему для учета заграничных командировок, в которые часто отправляются её сотрудники. Система должна обладать повышенным уровнем безопасности, а также должна быть полностью защищена от утечки информации.

Доступ к системе будет иметь определенная группа лиц, работающих в этой кампании.

В системе должны ежедневно обновляться данные о командировках сотрудников: их длительности, месте, цели и конкретных адресах, которые были посещены за время пребывания в другой стране, а также другая более подробная информация о поездке.

При необходимости кампания будет отправлять данные о командировках другим организациям, но только при наличии специального запроса и после прохождения всех проверок.

Система создается на длительное время, поэтому должна быть рассчитана на большие объемы информации.

Государственная кампания планирует нанимать специалистов для внедрения и сопровождения системы.

Система должна быть разработана в течение двух лет.

##### ***Вариант 2. Разработка системы для управления курьерской доставкой товаров***

Служба курьерской доставки товаров открыла новые офисы в нескольких городах России и планирует расширять зоны доставки как малогабаритных, так и крупногабаритных грузов в пределах страны.

Для этого требуется мощная система управления данной службой. Необходимо вести учет товара на складах одновременно в нескольких городах, контролировать перемещение товаров по стране, вести прием заказов и подсчет стоимости доставки на конкретное расстояние. С помощью этой системы также планируется начисление заработной платы сотрудникам, мониторинг состояния транспортных средств компании и т.д.

Служба курьерской доставки планирует запуск этой системы не позднее, чем через 8 месяцев.

Сопровождение системы не требуется.

##### ***Вариант 3. Разработка системы для учета товаров на складе небольшой фирмы***

Недавно появившаяся фирма арендовала складское помещение для хранения товаров.

Количество товаров ежедневно меняется, происходит отгрузка имеющихся и поступление новых партий товаров от различных фирм.

В один день может произойти и отгрузка и поступление одновременно.

Требуется система для учета товаров на складе: даты их закупки и продажи, их количества и цены.

Причем, если запасы какого-либо товара на складе близки к нулю, система должна информировать пользователя об этом.

Фирма молодая и не обладает крупным капиталом для реализации системы, потому её представители готовы обсудить детали и возможные планы сокращения стоимости проекта. Максимальный срок ожидания системы - 6 месяцев.

Сопровождение не требуется.

##### ***Вариант 4. Разработка системы для проведения рекламных компаний***

Компания проводит акции, обзванивая потенциальных клиентов, проживающих в одном городе.

Требуется разработать систему автоматического набора телефонных номеров. Система должна соединять оператора с клиентами. Каждому сотруднику должна предоставляться учетная запись для входа в систему. Должна поддерживаться одновременная работа нескольких операторов. Во время проведения каждой акции клиентам должны присваиваться различные статусы для слежения за ходом кампании. Система должна взаимодействовать с существующей в компании системой учета клиентов.

Срок разработки системы – 1 год. Планируется ввести продукт в рабочий цикл компании в два этапа: сначала часть с базой клиентов и подсистему статистики, затем – автоматизации работы и контроля сотрудников.

Оператор ЭДО ООО "Компания "Тензор"

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ

СОГЛАСОВАНО

**ФГБОУ ВО "РГРТУ", РГРТУ**, Костров Борис Васильевич,  
Заведующий кафедрой ЭВМ

**12.12.25 12:51 (MSK)**

Простая подпись