

АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ ОРГАНИЗАЦИЯ ВЫСШЕГО ОБРАЗОВАНИЯ  
САМАРСКИЙ УНИВЕРСИТЕТ ГОСУДАРСТВЕННОГО УПРАВЛЕНИЯ  
«МЕЖДУНАРОДНЫЙ ИНСТИТУТ РЫНКА»

УТВЕРЖДАЮ  
Проректор по учебной работе и  
качеству образования

\_\_\_\_\_ И. А. Долгова

16 апреля 2025 г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДИСЦИПЛИНЫ

---

**ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ**

---

Направление подготовки:	09.03.03 Прикладная информатика
Профиль подготовки:	Корпоративные информационные системы
Квалификация:	бакалавр
Форма обучения:	очная, очно-заочная
Год начала подготовки:	2025

# 1. ОЦЕНОЧНЫЕ СРЕДСТВА, СООТНЕСЁННЫЕ С ПЛАНИРУЕМЫМИ РЕЗУЛЬТАТАМИ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Код и наименование компетенции	Код и наименование индикатора достижения компетенции	Результаты обучения по дисциплине	Вид аттестации и оценочных средств
ОПК-4 Способен участвовать в разработке стандартов, норм и правил, а также технической документации, связанной с профессиональной деятельностью;	ОПК-4.И-1. Участвует в разработке стандартов, норм и правил, связанных с профессиональной деятельностью	ОПК-4.И-1.3-1. Знает основные нормативные документы в профессиональной сфере	Текущий контроль: устный опрос, лабораторная работа, промежуточный тест. Промежуточная аттестация: вопросы на экзамене.
		ОПК-4.И-1.У-1. Умеет применить знания основных нормативных документов при разработке новых стандартов, норм и правил в профессиональной сфере	
	ПК-4.И-2. Участвует в разработке технической документации, связанной с профессиональной деятельностью	ОПК-4.И-2.3-1. Знает основные приемы разработки технической документации, связанной с профессиональной деятельностью	Текущий контроль: устный опрос, решение задач/кейсов, доклад, реферат, лабораторная работа, промежуточный тест. Промежуточная аттестация: зачет (контрольное задание), вопросы на экзамене.
		ОПК-4.И-2.У-1. Умеет использовать основные приемы разработки технической документации, связанной с профессиональной деятельностью	

## 2. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ

### 2.1. Вопросы для подготовки к семинарским/практическим занятиям

Раздел 1. Программная инженерия: назначение, основные принципы и понятия

1. Стандарты в программной инженерии
2. Жизненный цикл ИС
3. Требования к программному обеспечению

Раздел 2. Управление программными проектами

4. Каноническое проектирование
5. Структурный анализ и проектирование в методологиях IDEF0, DFD
6. Проектирование пользовательского интерфейса
7. Тестирование и управление качеством

Раздел 3. Объектно-ориентированный подход к бизнес-моделированию

- 8.ООП: ведение в язык UML
- 9.UML: варианты использования
- 10.UML: диаграммы классов
11. UML: поведенческие диаграммы
12. UML: диаграммы реализации
13. Построение модели данных в нотации ER

Раздел 4. Оценка трудоемкости разработки ПО

14. Методология разработки ПО Rational Unified Process
- 15.Определение весовых показателей действующих лиц, вариантов использования ИС
16. Определение технической сложности ИС

#### Критерии оценки работы на практическом занятии

Критерии	Максимальное количество баллов за занятие
<b>Устный опрос, коллоквиум</b>	
Основные теоретические положения по вопросу раскрыты. Имеются элементы обоснования выводов. Имеются элементы систематизации информации, факты применения профессиональной терминологии. Очевидно использование источников рекомендованной литературы.	5 баллов

## 2.2. Темы лабораторных работ

### Раздел 1. Программная инженерия: назначение, основные принципы и понятия

Лабораторная работа №1. Разработка и анализ требований к информационной системе (ИС)

*Цель работы:* отработка ключевых понятий жизненного цикла (ЖЦ) ИС и первой стадии ЖЦ - Требования к ИС

*Вопросы для самопроверки:*

1. Основные этапы разработки требований к ИС, уровни и свойства требований
2. Методы выявления требований
3. Управление требованиями

### Раздел 2. Управление программными проектами

Лабораторная работа №1. Структурный анализ и проектирование в методологиях IDEF0

*Цель работы:* Составление модели информационной системы в графической нотации IDEF0 (структурный анализ и проектирование)

*Вопросы для самопроверки:*

1. Процесс моделирования информационной системы в IDEF0
2. Что такое функциональное моделирование в IDEF0
3. Какие диаграммы содержит модель ИС в нотации IDEF0

Лабораторная работа №2. Структурный анализ и проектирование в методологиях, DFD

*Цель работы:* Составление модели информационной системы в графической нотации DFD (структурный анализ и проектирование)

*Вопросы для самопроверки:*

1. Процесс моделирования информационной системы в DFD
2. Что такое функциональное моделирование в DFD
3. Какие диаграммы содержит модель ИС в нотации DFD

### Раздел 3. Объектно-ориентированный подход к бизнес-моделированию

Лабораторная работа №1. Введение в язык UML

*Цель работы:* дать основные понятия и «строительные блоки» UML

*Вопросы для самопроверки:*

1. Основные этапы развития UML
2. Определение, документация и сфера применения UML
3. Ключевые правила языка UML

Лабораторная работа №2. UML: варианты использования

*Цель работы:* дать основные понятия и «строительные блоки» UML

*Вопросы для самопроверки:*

1. Какие диаграммы определены в терминах языка UML
2. Формирование требований с помощью диаграммы Use Case
3. Сценарии вариантов использования

Лабораторная работа №4. UML: диаграммы классов

*Цель работы:* дать основные понятия и «строительные блоки» UML

*Вопросы для самопроверки:*

1. Диаграммы классов: графическое представление, виды классов, атрибуты классов
2. Структурные отношения между классами
3. Сценарии вариантов использования

Лабораторная работа №5. UML: поведенческие диаграммы

*Цель работы:* дать основные понятия и «строительные блоки» UML

*Вопросы для самопроверки:*

1. Основные элементы диаграммы состояний
2. Основные элементы диаграммы деятельности
3. Сценарии вариантов использования

Лабораторная работа №6. UML: диаграммы реализации

*Цель работы:* дать основные понятия и «строительные блоки» UML

*Вопросы для самопроверки:*

1. Разновидности диаграммы реализации: диаграммы компонентов и диаграммы развертывания
2. Основные элементы и графическое представление диаграмм
3. Сценарии вариантов использования

Лабораторная работа №7. Построение ER – модели данных

*Цель работы:* дать основные понятия и «строительные блоки» UML

*Вопросы для самопроверки:*

1. Базовые понятия ER - модели
2. ER - диаграммы
3. Сценарии вариантов использования

Лабораторная работа №8. Проектирование пользовательского интерфейса

*Цель работы:* дать основные понятия и «строительные блоки» UML

*Вопросы для самопроверки:*

1. Основные принципы реализации интерфейса
2. Понятие о юзабилити, основные правила
3. Сценарии вариантов использования

Лабораторная работа №9. Оценка трудоемкости разработки ПО

*Цель работы:* дать практические навыки работы с методами оценки трудоемкости

*Вопросы для самопроверки:*

1. Характеристика методов оценки трудоемкости ПО
2. Этапы оценки
3. Методики измерения трудоемкости LOC и UC

## **Методические указания к проведению лабораторных работ**

### **Цели лабораторных занятий:**

1. Углубление и закрепление знания теоретического курса путем практического изучения в лабораторных условиях изложенных в лекциях методов и технологий;
2. Приобретение навыков в научном экспериментировании, анализе полученных результатов;
3. Формирование первичных навыков организации, планирования и проведения научных исследований.

### **Порядок проведения лабораторного занятия:**

1. Вводная часть:
  - входной контроль подготовки обучаемого;
  - вводный инструктаж (знакомство обучающихся с содержанием предстоящей работы, краткий анализ теоретических положений и выводов, демонстрация подходов к выполнению отдельных операций, напоминание о технике безопасности, предупреждение о возможных ошибках).
2. Основная часть:
  - проведение обучаемым лабораторной работы;
  - текущее индивидуальное консультирование обучаемого;

### 3. Заключительная часть:

- демонстрация результатов выполненного задания;
- заключительный инструктаж (подведение итогов выполнения учебных задач, разбор допущенных ошибок и выявление их причин, сообщение результатов работы каждого обучаемого, объявление о том, что необходимо повторить к следующему занятию).

### **Особенности подготовки к проведению лабораторного занятия**

Подготовка лабораторного занятия начинается с изучения теоретических положений, определения (уточнения) целей и задач данного занятия, времени, выделяемого обучаемым для подготовки.

В ходе подготовки к лабораторной работе необходимо пояснить проблематику, объем и содержание лабораторного занятия, определить, какие понятия, определения, теории могут быть иллюстрированы данным экспериментом, какие умения и навыки должны приобрести обучаемые в ходе занятия, какие знания углубить и расширить.

При этом преподавателю необходимо решить, на каком этапе обучения следует поставить задачу о подготовке к лабораторной работе, каким образом достигнуть активизации познавательной деятельности обучающихся. Задача на подготовку к лабораторной работе может быть поставлена на лекции, с таким временным расчетом, чтобы обучаемые смогли качественно подготовиться к ее проведению. Одновременно им выдаются учебно-методические материалы, иллюстрирующие круг вопросов, затрагиваемых в ходе выполнения лабораторного задания. Это могут быть методические указания по соответствующему курсу, презентации, ссылки на Интернет-источники и др. Эти материалы могут отражать учебные вопросы, краткие сведения по теории, программу выполнения работы, содержание отчета, вопросы для подготовки и литературу, рекомендуемую к изучению и т.д. В них также ставятся задачи, которые обучаемые должны решить при подготовке к работе, в процессе эксперимента и при обработке полученных результатов.

В ходе подготовки к лабораторной работе необходимо обратить внимание обучающегося на результат ее выполнения. Результат лабораторной работы должен быть четко сформулирован, приведены критерии его достижения, перечень материалов, его (результат) иллюстрирующих – файлы, графики, скриншоты и т.д. Учащийся должен уметь формулировать основные выводы, опираясь на полученный на лабораторной работе результат.

В отдельных случаях, на лабораторном занятии может быть предусмотрена защита выполненной работы.

### **Шкала и критерии оценки лабораторной работы**

<b>Критерии</b>	<b>Баллы</b>
Степень соответствия выполненного задания поставленным требованиям	25
Структурирование и комментирование лабораторной работы	25
Уникальность выполненной работы (отличие от работ коллег)	25
Ответы на контрольные вопросы	25

Лабораторная работа оценивается по 100 балльной шкале, баллы переводятся в оценки успеваемости следующим образом:

90 – 100 баллов – «отлично»;

70 – 89 баллов – «хорошо»;

50 – 69 баллов – «удовлетворительно»;

менее 50 баллов – «неудовлетворительно».

### 3. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

#### 3.1. Банк контрольных заданий (с указанием компетенции)

**1. ОПК-4.1 Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа.**

Для какой модели жизненного цикла верно следующее утверждение: на каждом этапе создается законченный набор документации?

- А) спиральной
- Б) каскадной
- В) инкрементной
- Г) итерационной

Ответ:

Обоснование:

**2. ОПК-4.1 Прочитайте текст, выберите правильные ответы и запишите аргументы, обосновывающие выбор ответов.**

Какие из требований имеют самый высокий уровень?

- А) бизнес-требования
- Б) пользовательские требования
- В) системные требования
- Г) функциональные требования

Ответ:

Обоснование:

**3. ОПК-4.1 Прочитайте текст, выберите правильные ответы и запишите аргументы, обосновывающие выбор ответов.**

Отметьте основные разделы технического задания:

- А) характеристика объекта информатизации
- Б) порядок контроля и приемки
- В) эскизный проект информационной системы
- Г) организация информационной базы

Ответ:

Обоснование:

**4. ОПК-4.1 Прочитайте текст, выберите правильные ответы и запишите аргументы, обосновывающие выбор ответов.**

К структурным диаграммам в UML относятся диаграммы:

- А) классов
- Б) пограничный класс
- В) интерфейсный класс
- Г) управляющий класс

Ответ:

Обоснование:

**5. ОПК-4.1 Прочитайте текст, выберите правильные ответы и запишите аргументы, обосновывающие выбор ответов.**

В модели бизнес-процессов UML к бизнес-актерам относятся:

- А) заказчики
- Б) клиенты
- В) менеджеры организации
- Г) партнеры

Ответ:

Обоснование:

**6. ОПК-4.1 Прочитайте текст, выберите правильные ответы и запишите аргументы, обосновывающие выбор ответов.**

При расчете трудоемкости методом вариантов использования весовой коэффициент варианта определяется:

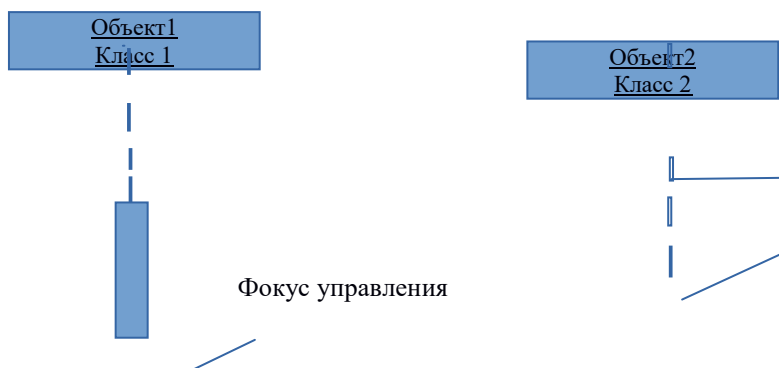
- А) сложностью
- Б) числом транзакций
- В) числом действующих лиц
- Г) числом операций, необходимых для его выполнения

Ответ:

Обоснование:

**7. ОПК 4.2 Прочитайте текст и запишите развёрнутый обоснованный ответ**

На рисунке изображен фрагмент диаграммы последовательности. Что обозначает пунктирная линия?



Ответ:

**8. ОПК 4.2 Прочитайте текст и запишите развёрнутый обоснованный ответ**

На рисунке вопроса 6 изображен фрагмент диаграммы последовательности. Что обозначает сплошная горизонтальная линия?

Ответ:

**9. ОПК4.2 Прочитайте текст и запишите развёрнутый обоснованный ответ**



Как называется диаграмма, на которой представлены узлы выполнения программных компонентов реального времени

Ответ:

**10. ОПК4.2 Прочитайте текст и запишите развёрнутый обоснованный ответ**

Дайте определение понятию «жизненный цикл программного обеспечения»

Ответ:

**11. ОПК-4.1 Прочитайте текст, выберите правильные ответы и запишите аргументы, обосновывающие выбор ответов.**

Единицами измерения размера программного продукта являются количество

- А) строк кода
- Б) операторов
- В) классов
- Г) функциональных точек
- Д) вариантов использования

Ответ:

Обоснование:

**12. ОПК-4.1 Прочитайте текст, выберите правильные ответы и запишите аргументы, обосновывающие выбор ответов.**

С учетом психофизиологии восприятия цветов общий фон экрана следует делать

- А) красным
- Б) черным
- В) белым
- Г) зеленым
- Д) голубым

Ответ:

Обоснование:

**13. ОПК-4.1 Прочитайте текст, выберите правильные ответы и запишите аргументы, обосновывающие выбор ответов.**

Пусть два класса А и В связаны направленной ассоциацией от А к В. Тогда:

- А) класс А может управлять операциями класса В
- Б) класс В может управлять операциями класса А
- В) класс А и класс В не связаны друг с другом
- Г) класс В всегда наследует от класса А

Ответ:

Обоснование:

**14. ОПК4.2 Прочитайте текст и запишите развёрнутый обоснованный ответ**

На диаграмме состояний факт смены одного состояния другим изображается как переход. Переходы бывают триггерными и нетриггерными. Для триггерного перехода необходимо наличие только одного обязательного элемента. Напишите название этого элемента и его функции

Ответ:

**15. ОПК-4.1 Прочитайте текст, выберите правильные ответы и запишите аргументы, обосновывающие выбор ответов.**

Отметьте известные вам принципы структурного программирования

- А) наследование
- Б) полиморфизм
- В) исходящее проектирование
- Г) инкапсуляция
- Д) программирование без goto

Ответ:

Обоснование:

### 13.2. Ключи к контрольным заданиям

№ задания	Верный ответ
1	<p><b>В</b></p> <p>Данная модель применяется при разработке информационных систем, для которых в самом начале разработки можно достаточно точно и полно сформулировать все требования:</p> <ul style="list-style-type: none"> <li>- на каждой стадии формируется законченный набор документации, программного и аппаратного обеспечения, отвечающий критериям полноты и согласованности;</li> <li>- выполняемые в четкой последовательности стадии позволяют уверенно планировать сроки выполнения работ и соответствующие ресурсы (денежные, материальные и людские).</li> </ul>
2	<p><b>А</b></p> <p>Требования разделяются по уровням. Уровни требований связаны, с одной стороны, с уровнем абстракции системы, с другой – с уровнем управления на предприятии. Обычно выделяют три уровня требований. На верхнем уровне представлены так называемые бизнес-требования (business requirements). Примеры бизнес-требования: система должна сократить срок оборачиваемости обрабатываемых на предприятии заказов в три раза. Бизнес-требования обычно формулируются топ-менеджерами, либо акционерами предприятия.</p> <p>Следующий уровень – уровень требований пользователей (user requirements). Пример требования пользователя: система должна представлять диалоговые средства для ввода исчерпывающей информации о заказе, последующей фиксации информации в базе данных и маршрутизации информации о заказе к сотруднику, отвечающему за его планирование и исполнение. Требования пользователей часто бывают плохо структурированными, дублирующимися, противоречивыми. Поэтому для создания системы важен третий уровень, в котором осуществляется формализация требований.</p> <p>Третий уровень – функциональный (functional requirements). Пример функциональных требований (или просто функций) по работе с электронным заказом: заказ может быть создан, отредактирован, удалён и перемещён с участка на участок.</p>
3	<p><b>А Б В</b></p> <p>Написание технического задания — один из первых этапов работы над проектом. Он предваряет разработку самой системы. В техническом задании мы описываем предметную область, существующую инфраструктуру Заказчика, требования к создаваемому функционалу, а также нефункциональные требования. Получившийся документ необходим как бизнес-пользователю для того, чтобы он убедился в том, что все его пожелания к будущей системе учтены, так и нам, чтобы оценить стоимость разработки системы.</p> <p>Структура ТЗ</p> <ol style="list-style-type: none"> <li>1. Общие положения</li> <li>2. Назначение и цели создания (развития) системы</li> <li>3. Характеристика объекта автоматизации</li> <li>4. Требования к системе</li> <li>5. Состав и содержание работ по созданию системы</li> <li>6. Порядок контроля и приемки системы</li> <li>7. Требования к составу и содержанию работ по подготовке объекта</li> </ol>

	<p>автоматизации к вводу системы в действие</p> <p>8. Требования к документированию</p> <p>9. Источники разработки</p>
4	<p><b>А</b></p> <p>Поскольку многие проекты опираются на объектно-ориентированное программирование, где от разработчика требуется описание типов возможных функций, диаграммы классов — самый популярный вариант UML. На диаграмме классов приводится статичная структура системы, включая классы, их атрибуты, поведение и взаимосвязи.</p> <p>Класс обозначается прямоугольником из трех блоков, расположенных один над другим: в верхнем блоке обязательно указывается имя класса, а в остальных двух — дополнительная информация: атрибуты, операции и поведение.</p>
5	<p><b>Б Г</b></p> <p>Бизнес-актер (business actor) – индивидуум, группа, организация, компания или система, которые взаимодействуют с моделируемой бизнес-системой, но не входят в нее, т.е. не являются частью моделируемой системы.</p> <p>Примерами бизнес-актеров являются клиенты, покупатели, поставщики, партнеры. Общее свойство бизнес-актеров состоит в том, что они являются инициаторами или клиентами бизнес-процессов моделируемой системы.</p>
6	<b>А Б В</b>
7	<p>Линия жизни объекта (object lifeline) изображается пунктирной вертикальной линией, ассоциированной с единственным объектом на диаграмме последовательности. Линия жизни служит для обозначения периода времени, в течение которого объект существует в системе и, следовательно, может потенциально участвовать во всех ее взаимодействиях. Если объект существует в системе постоянно, то и его линия жизни должна продолжаться по всей плоскости диаграммы последовательности от самой верхней ее части до самой нижней.</p>
8	<p>Сплошная горизонтальная линия на диаграмме последовательности в UML обозначает линию жизни объекта, представляющую время существования объекта во время взаимодействия. Она показывает период, в течение которого объект активен и участвует в взаимодействии.</p> <p>Обычно сообщения изображаются горизонтальными стрелками, соединяющими линии жизни или фокусы управления двух объектов на диаграмме последовательности. При этом неявно предполагается, что время передачи сообщения достаточно мало по сравнению с процессами выполнения действий объектами. Считается также, что за время передачи сообщения с соответствующими объектами не может произойти никаких событий. Другими словами, состояния объектов остаются без изменения.</p>
9	<p>Диаграмма, на которой представлены узлы выполнения программных компонентов реального времени, называется диаграммой развертывания (Deployment diagram). Она показывает, как программные</p>

	компоненты ("артефакты") расположены и взаимодействуют на физических узлах (например, аппаратных компонентах) в системе реального времени.
10	<p>Жизненный цикл программного обеспечения (ПО) – это последовательность этапов, которые проходит программное обеспечение от момента зарождения идеи до полного вывода из эксплуатации. Он включает в себя разработку, тестирование, внедрение, поддержку и обслуживание ПО.</p> <p>Этапы жизненного цикла ПО:</p> <p>В разных моделях жизненного цикла эти этапы могут быть представлены по-разному, но в целом включают:</p> <ul style="list-style-type: none"> <li>• Анализ требований: определение требований к ПО, его функциональности и целевой аудитории.</li> <li>• Проектирование: разработка архитектуры и структуры ПО.</li> <li>• Кодирование: написание кода, реализующего функциональность ПО.</li> <li>• Тестирование: проверка ПО на соответствие требованиям и выявлению ошибок.</li> <li>• Внедрение: развертывание ПО в эксплуатационную среду.</li> <li>• Поддержка и сопровождение: исправление ошибок, обновление ПО и предоставление консультаций пользователям.</li> <li>• Вывод из эксплуатации: прекращение поддержки и использования ПО.</li> <li>• Модели жизненного цикла:</li> </ul> <p>Существуют различные модели жизненного цикла ПО, такие как каскадная модель, итеративная модель, спиральная модель и другие. Выбор модели зависит от конкретного проекта и его требований.</p> <ul style="list-style-type: none"> <li>• Цель жизненного цикла ПО:</li> </ul> <p>Упростить процесс разработки и поддержания ПО, повысить его качество, снизить риски и затраты, а также обеспечить более эффективное использование ресурсов.</p>
11	<p><b>А Г</b></p> <p>Единицами измерения размера программного продукта могут быть различные показатели, например, количество функциональных точек, строки кода, или количество бит/байт, занимаемых программой в памяти. Кроме того, можно использовать более сложные метрики, такие как функциональные точки.</p> <p>Функциональные точки (FP). Это единицы измерения, которые отражают функциональную сложность программного продукта. Они основаны на анализе функций, которые система должна выполнять, и данных, которые она обрабатывает.</p> <p>Строки кода (LOC). Это простой способ измерения размера программного продукта, он измеряет количество строк в исходном коде. Хотя LOC может быть полезен для оценки объема работы, его использование имеет ограничения, так как он не учитывает сложность кода или его функциональность.</p> <p>Бит/байт. Это базовые единицы измерения информации, используемые в компьютерных системах. 1 байт равен 8 битам. Они часто используются для измерения размера файлов, программ и данных, хранящихся в памяти компьютера.</p> <p>Килобайт (KB), Мегабайт (MB), Гигабайт (GB), Терабайт (TB). Производные единицы измерения информации, которые используются для более крупных объемов данных. Например, 1 Кбайт = 1024 байта, 1 Мбайт = 1024 Кбайт, и так далее.</p> <p>Другие метрики. Существуют и другие метрики, например, количество пользовательских интерфейсов, количество модулей, количество используемых библиотек, и т.д.</p>

12	<p><b>В Г Д</b></p> <p>С учетом психофизиологии восприятия цветов, общий фон экрана рекомендуется делать не слишком ярким и насыщенным, а также в нейтральных тонах, таких как светлые оттенки серого, белого или голубого. Зеленый цвет также может быть хорошим выбором, так как он считается одним из самых комфортных для глаз. Почему это важно:</p> <p>Снижение зрительной усталости: Яркие и насыщенные цвета могут вызывать зрительную усталость при длительной работе за компьютером.</p> <p>Повышение концентрации: Нейтральные цвета помогают лучше концентрироваться и снижают рассеянность внимания.</p> <p>Успокаивающий эффект: Нейтральные тона могут способствовать созданию более спокойной и комфортной рабочей среды.</p> <p>Улучшение восприятия других элементов: Нейтральный фон не отвлекает от основного контента и позволяет другим элементам (тексту, изображениям) выделяться.</p> <p>Зеленый цвет: Воспринимается непосредственно сетчаткой без рефракции, что делает его комфортным для глаз. Примеры:</p> <p>Светло-серый: Классический и нейтральный вариант, подходит для большинства типов контента.</p> <p>Светлый бежевый или кремовый: Более теплые варианты, которые могут создать более приятную атмосферу.</p> <p>Светло-голубой: Создает ощущение прохлады и спокойствия.</p> <p>Приглушенный зеленый: Способен успокоить и освежить.</p> <p>Важно: Выбор фона должен быть индивидуальным и зависеть от личных предпочтений и типа работы.</p> <p>Необходимо учитывать цветовые контрасты между фоном и другим контентом для обеспечения читаемости.</p>
13	<p><b>А</b></p> <p>При направленной ассоциации от А к В, класс А может управлять операциями класса В. Объяснение: направленная ассоциация между классами А и В подразумевает, что класс А может взаимодействовать с объектами класса В, возможно, даже вызывая или запрашивая выполнение операций класса В. Эта связь указывает, что класс А имеет некоторую степень контроля над классом В, хотя и не обязательно полное владение или контроль. Другие варианты ответов и почему они неверны:</p> <p>Б) класс В может управлять операциями класса А: Это неверно. Направленная ассоциация указывает на обратное: А может управлять В, а не наоборот.</p> <p>В) класс А и класс В не связаны друг с другом: Это неверно, поскольку ассоциация по определению является связью между классами.</p> <p>Г) класс В всегда наследует от класса А:</p>

	<p>Наследство и ассоциация - разные типы связей. Наследство указывает на иерархию классов, где один класс является подклассом другого, а ассоциация описывает взаимодействие между классами.</p>
14	<p>Обязательным элементом для триггерного перехода в диаграмме состояний (UML) является имя события. Это имя события, которое инициирует переход из одного состояния в другое. Дополнительно, могут быть указаны параметры события, сторожевые условия и действия, которые выполняются при переходе. Более детально:</p> <p>Имя события: Это строка текста, которая обозначает событие, которое триггирует переход. Например, "click", "receive", "error" и т.д.</p> <p>Параметры события: Дополнительные данные, которые передаются вместе с событием. Они могут быть указаны в скобках после имени события.</p> <p>Сторожевые условия: Условия, которые должны быть выполнены, чтобы событие привело к переходу. Если условие не выполняется, переход не произойдет. Сторожевые условия записываются в квадратных скобках после имени события.</p> <p>Действия: Действия, которые выполняются после того, как произошел переход. Они могут быть указаны после имени события.</p> <p>Пример: event(parameter1, parameter2) [guard] action В этом примере: event - это имя события. parameter1, parameter2 - параметры события. [guard] - сторожевое условие. action - действие, выполняемое после перехода.</p>
15	<p><b>А Б Г</b> Ключевые принципы структурного программирования.</p> <p>Модульность: Одной из основ является деление программы на модули или блоки. Каждый блок решает отдельную задачу, что облегчает понимание и тестирование. Модульность способствует разделению обязанностей, что увеличивает эффективность разработки и упрощает отладку.</p> <p>Использование подпрограмм: Подпрограммы или функции являются краеугольным камнем. Они позволяют реализовывать отдельные кусочки логики, которые можно вызывать многократно, что уменьшает дублирование кода и повышает его реюзабилити.</p> <p>Логическая структура: Важным принципом является строгое соблюдение логической структуры программы. Использование условных операторов, циклов и других управляющих конструкций позволяет строить код, логика которого легко прослеживается и анализируется.</p> <p>Последовательность: Программы должны выполняться с точной последовательностью операций. Это способствует предсказуемости и упрощает понимание алгоритмов.</p> <p>Инкапсуляция: Принцип инкапсуляции предполагает сокрытие деталей реализации внутри модулей. Внешний код взаимодействует с модулем через</p>

	определённый интерфейс, что повышает безопасность и защищённость данных.
--	--

### Шкала и критерии оценки текущего тестирования

Число правильных ответов	Оценка
90-100% правильных ответов	Оценка «отлично»
70-89% правильных ответов	Оценка «хорошо»
50-69% правильных ответов	Оценка «удовлетворительно»
Менее 50% правильных ответов	Оценка «неудовлетворительно»



### 23.3. Перечень тем для проверки образовательных результатов на знания (вопросы к экзамену)

#### Тема. Инженерия ПО

1. Почему произошел первый кризис программирования?
2. Отметьте известные вам принципы структурного программирования
3. Почему возникло объектно-ориентированное программирование?
4. Какова главная цель программной инженерии?
5. Что такое программное обеспечение?
6. Выберите верное определение программной инженерии:

#### Тема. Жизненный цикл ПО

1. Жизненный цикл ПО – это:
2. Отметьте известные вам этапы жизненного цикла ПО:
3. Отметьте известные вам модели жизненного цикла ПО:
4. Для какой модели верно следующее утверждение: на каждом этапе создается законченный набор документации?
5. Для какой модели верно следующее утверждение: на каждом витке выпускается работоспособная версия системы?
6. На каком этапе жизненного цикла стоимость внесения изменений в систему самая высокая? 7. Для какой модели жизненного цикла стоимость внесения изменений в систему на поздних стадиях самая высокая?
8. Для каких моделей жизненного цикла характерны трудности с внедрением, такие как невозможность или неудобство внедрения неполной функциональности?

#### Тема. Требования

1. Каков правильный подход к определению понятия «требование»?
2. Отметьте известные вам стадии процесса разработки требований:
3. На каком этапе жизненного цикла разработки программного обеспечения должны быть разработаны требования к системе?
4. Расположите в хронологическом порядке этапы процесса разработки требований:
  - А) Формирование и анализ требований
  - Б) Анализ предметной области.
  - В) Анализ осуществимости.
  - Г) Детализация требований.
  - Д) Документирование требований.
5. Отметьте известные вам уровни требований.
6. Отметьте известные вам виды требований.
7. Какие требования и почему имеют самый высокий уровень?
8. Каково содержание функциональных требований?
9. Отметьте известные вам классы нефункциональных требований
10. Что такое спецификация требований?

#### Тема. Введение в UML

1. Отметьте основные принципы объектно-ориентированного программирования:
2. Выберите верное определение методологии объектно-ориентированного программирования.
3. Для какого понятия верно данное определение: это характеристика сущности, которая отличает ее от других сущностей. 2
4. Для какого понятия верно данное определение: это совокупность объектов, которые имеют общий набор свойств и обладают одинаковым поведением
5. Для какого понятия верно данное определение: это принцип, в соответствии с которым знание о наиболее общей категории разрешается применять для более частной категории
6. Для какого понятия верно данное определение: это сокрытие отдельных деталей внутреннего устройства классов от внешних по отношению к нему объектов или пользователей.

7. Для какого понятия верно данное определение: это принцип, согласно которому действия, выполняемые одноименными методами, могут различаться в зависимости от того, к какому из классов относится тот или иной метод.
8. Что такое Unified Modeling Language (UML)?
9. Для какого понятия верно данное определение: это проекция описания системы, рассматриваемой с определенной точки зрения, в которой высвечены детали, важные в данном аспекте, и опущены несущественные.
10. Отметьте известные вам виды системы:
11. Перечислите известные вам строительные блоки языка UML:
12. К какому типу сущностей относится сущность класс?
13. К какому типу сущностей относится сущность вариант использования?
14. К какому типу сущностей относится сущность компонент?
15. К какому типу сущностей относится сущность деятельность?
16. К какому типу сущностей относится сущность пакет?
17. К какому типу сущностей относится сущность примечание?
18. Отметьте диаграммы, относящиеся к структурным диаграммам UML
19. Отметьте диаграммы, относящиеся к поведенческим диаграммам UML
20. Перечислите диаграммы поведения UML, которые создаются при логическом моделировании системы.

#### Тема. Варианты использования

1. Выберите наиболее точное определение понятия вариант использования:
2. Чем различаются термины вариант использования и прецедент?
3. Включение – это такое отношение между вариантами использования, при котором:
4. Расширение – это такое отношение между вариантами использования, при котором:
5. Обобщение – это такое отношение между вариантами использования, при котором:
6. Отметьте известные вам цели создания модели вариантов использования:
7. Выберите правильное окончание следующей фразы: "Основной ход событий..."
8. Какой графический символ служит для изображения варианта использования (use case) в языке UML?
9. Какой графический символ служит для изображения примечания (note) в языке UML?
10. Выберите правильное окончание следующей фразы: "При разработке диаграммы вариантов использования в первую очередь необходимо..."

#### Тема. Классы 1.

В чем отличие понятий класс и объект?

2. Какие отношения между классами могут возникать при проектировании классов системы?
3. Отметьте известные вам секции прямоугольника, графически изображающего класс:
4. Для какого понятия верно это определение: служит для представления отдельного свойства или признака, который является общим для всех объектов данного класса. 3
5. Для какого понятия верно это определение: это сервис, предоставляемый каждым экземпляром или объектом класса по требованию своих клиентов, в качестве которых могут выступать другие объекты, в том числе и экземпляры данного класса.
6. Каким значком изображается квантор видимости "закрытый" (private) на диаграмме классов?
7. Какое определение конкретного класса (concrete class) правильно?
8. Укажите правильные записи кратности (multiplicity) конца ассоциации на диаграмме классов?
9. Как изображается отношение агрегации (aggregation) на диаграмме классов?
10. Как изображается отношение обобщения (generalization) на диаграмме классов?
11. Что такое интерфейс?
12. Какое утверждение верно для отношения наследования?
13. Какое отношение между классами показано на рисунке?
14. Чем отличаются отношения агрегации и композиции?

#### Тема. Диаграммы взаимодействия

1. Перечислите диаграммы поведения UML, которые создаются при логическом моделировании системы.
2. Диаграмма последовательности – это диаграмма взаимодействия, в которой основной акцент сделан на:
3. Отметьте элементы, из которых строится диаграмма последовательности:
4. Какой элемент диаграммы последовательности указан на рисунке?
5. Диаграмма коммуникации – это диаграмма взаимодействия, в которой основной акцент сделан на:
6. Имя объекта на диаграмме коммуникации представляет из себя следующую строку текста: Компьютер1/обработчик запросов: Сервер :: Серверы. Чем является подстрока обработчик запросов ?
7. . Имя объекта на диаграмме коммуникации представляет из себя следующую строку текста: Компьютер1/обработчик запросов: Сервер :: Серверы. Чем является подстрока Сервер ?
8. Активный объект в контексте UML – это объект, который:

#### Тема. Диаграмма состояний

1. Что моделирует диаграмма состояний?
2. Отметьте утверждения, верные по отношению к формализму обычного автомата:
3. К какому понятию относится данное определение: это ситуация в жизни объекта, на протяжении которой он удовлетворяет некоторому условию, выполняет определенную деятельность или ожидает какого-то события.
4. К какому понятию относится данное определение: это спецификация существенного факта, имеющего место в пространстве и во времени, некий стимул, инициирующий переход из одного состояния в другое.
5. К какому понятию относится данное определение: это поведение в терминах последовательности состояний, через которые проходит объект в течение своего жизненного цикла, отвечая на различные события, а также его реакций на эти события.
6. Перечислите известные вам фиксированные в UML метки действий, выполняемых при нахождении объекта в каком-либо состоянии.
7. Каким графическим символом обозначается состояние на диаграмме состояний?
8. Отметьте элементы, которые характеризуют переход.
9. Как графически обозначается составное состояние на диаграмме состояний?
10. Что такое реактивный объект? 4

#### Тема. Диаграмма деятельности

1. Закончите фразу: диаграмма деятельности может дополнить любой элемент UML модели:
2. Перечислите основные элементы диаграммы деятельности:
3. Для чего нужны плавательные дорожки (swimlines) на диаграмме деятельности?

#### Тема. Диаграммы реализации

1. К какому уровню моделей системы относится диаграмма компонентов?
2. К какому уровню моделей системы относится диаграмма узлов?
3. Отметьте элементы, из которых строится диаграмма компонентов.

#### Тема. Оценка трудоемкости создания программного обеспечения

1. Для решения каких задач применяются методы и модели оценки трудоемкости создания программного обеспечения (ПО)?
2. В чем состоит суть метода оценки трудоемкости под названием «алгоритмическое моделирование»?
3. В чем состоит суть метода оценки трудоемкости под названием «экспертные оценки»?
4. В чем состоит суть метода оценки трудоемкости под названием «оценка по аналогии»?
5. В чем состоит суть метода оценки трудоемкости под названием «оценка с целью выиграть контракт»?
6. Какой фактор оценки трудоемкости из ниже перечисленных является наиболее влиятельным?
7. В каком порядке выполняются этапы процедуры оценки трудоемкости?

8. Какие основные единицы измерения размера ПО вы знаете?
9. От чего зависит значение поправочного коэффициента Value Adjustment Factor?
10. В каком диапазоне изменяются значения общих характеристик системы (General System Characteristics)?
11. Какой объем в функциональных точках имеет проект со следующими характеристиками: 1 день работы 1-го разработчика.
12. Какой объем в функциональных точках имеет проект со следующими характеристиками: 1 месяц работы 1-го разработчика.
13. Какой объем в функциональных точках имеет проект со следующими характеристиками: 6 месяцев работы 1-го разработчика.
14. Какой объем в функциональных точках имеет проект со следующими характеристиками: 1 год работы, 10 разработчиков.
15. Какой объем в функциональных точках имеет проект со следующими характеристиками: 1,5-5 лет работы, 100 разработчиков.
16. Какой объем в функциональных точках имеет проект со следующими характеристиками: 5-8 лет работы, сотни разработчиков, это операционные системы и крупные военные приложения
17. Отметьте известные вам этапы методики расчета трудоемкости на основе вариантов использования:
18. Выберите тип действующего лица, для которого верно следующее определение: это внешняя система с четко определенным программным интерфейсом.
19. Выберите тип действующего лица, для которого верно следующее определение: это либо внешняя система, взаимодействующая с данной системой посредством протокола наподобие TCP/IP, либо личность, пользующаяся текстовым интерфейсом.
20. Выберите тип действующего лица, для которого верно следующее определение: это личность, пользующуюся графическим интерфейсом.
21. Какие способы определения весового показателя варианта использования в методике оценки трудоемкости по вариантам использования вы знаете:
22. В каком диапазоне изменяются значения показателей технической сложности проекта в методике оценки трудоемкости по вариантам использования?

#### Тема. Тестирование ПО

1. Принципы и уровни тестирования
2. Виды и методы тестирования
3. Тестирование требований
4. Тестовая документация – тест-кейс, баг-репорт...

#### Тема. Пользовательский интерфейс

1. Выберите правильное определение понятия «пользовательский интерфейс»
2. Какие показатели качества программы должен обеспечить пользовательский интерфейс?
3. Что такое «юзабилити» применительно к пользовательскому интерфейсу компьютерных программ?
4. Какова задача юзабилити?
5. Какие права человека уважает идеология юзабилити?
6. Для какой количественной характеристики юзабилити верно следующее определение: насколько легко пользователи могут выполнить основные задачи, впервые встретившись с незнакомым им интерфейсом?
7. Для какой количественной характеристики юзабилити верно следующее определение: после того, как пользователи ознакомятся с дизайном, как быстро они могут выполнить то же задание?
8. Для какой количественной характеристики юзабилити верно следующее определение: после того, как пользователь снова вернется к интерфейсу через определенный промежуток времени, насколько легко ему будет восстановить свои навыки работы с данным интерфейсом?
9. Для какой количественной характеристики юзабилити верно следующее определение: насколько пользователю приятно пользоваться данным интерфейсом?

10. Для какого понятия верно следующее определение: эта характеристика программы описывает функциональность дизайна и отвечает на вопрос - является ли интерфейс тем, что пользователю нужно?
11. Для какого понятия верно следующее определение: это качественный признак программы, который определяет, насколько интерфейс пользователя легок в использовании?
12. Какая доля бюджета разработки Web-сайта должна тратиться на юзабилити?
13. Какой метод проверки юзабилити интерфейса является самым действенным?

#### Тема. Управление ИТ-проектами

1. Принципы гибкой методологии разработки ПО Agile.
2. Для какого типа систем согласно модели Кеневин целесообразно использовать методологию разработки SCRUM?
3. Для каких продуктов/проектов эффективно использование SCRUM?
4. Главные особенности Scrum:
5. Выберите верные утверждения для команды разработки SCRUM Team.
6. Что такое bus-factor команды?
7. Стадии развития команды согласно модели Такмана.
8. Обязательные события SCRUM-спринта, которые происходят ежедневно:
9. Отметьте обязательные события SCRUM-спринта, которые происходят в конце итерации:
10. Обязанности SCRUM-мастера: 6
11. Человек, отвечающий за разработку продукта и принимающий важные окончательные решения в проекте, называется:
12. Инкремент продукта (product increment) это:
13. Как называется документ, который содержит перечень рабочих задач проекта, расположенных в порядке важности, для команды разработчиков.
14. Как называется документ, который содержит перечень рабочих задач текущей итерации, расположенных в порядке важности, для команды разработчиков.
15. Как называется документ, который содержит описание требований к продукту на языке, понятном для всех заинтересованных лиц.
16. Вопросы, которые обсуждаются на Daily scrum (15-минутная ежедневная встреча):
17. На каком собрании проводится демонстрация результатов очередной итерации:
18. На каком собрании проводится анализ очередной итерации в плане взаимодействия, инструментов и качества SCRUM-процессов:
19. Понятие Story Points:
20. Понятие Velocity (скорость работы команды)
21. Что показывает диаграмма сгорания (burndown chart)?
22. Технический долг – это: