

МИНОБРНАУКИ РОССИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»  
(ФГБОУ ВО «ВГУ»)

**УТВЕРЖДАЮ**

Заведующий кафедрой  
программирования и информационных технологий



проф. Махортов

С.Д.

подпись, расшифровка подписи

05.03.2024 г.

## РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

Б1.В.08 Технологии программирования

**1. Код и наименование направления подготовки/специальности:**

09.03.02 Информационные системы и технологии

**2. Профиль подготовки/специализация/магистерская программа:**

Инженерия информационных систем и технологий

**3. Квалификация (степень) выпускника:** бакалавр

**4. Форма обучения:** очная

**5. Кафедра, отвечающая за реализацию дисциплины:**

программирования и информационных технологий

**6. Составители программы:** Тарасов Вячеслав Сергеевич

**7. Рекомендована:** НМС ф-та компьютерных наук, протокол № 5 от 05.03.2024 г.

**8. Учебный год:** 2026/2027

**Семестр(ы):** 6

**9. Цели и задачи учебной дисциплины:** Целью изучения дисциплины является формирование теоретических и практических навыков в области создания надежного и качественного программного обеспечения с применением современных технологий программирования, методов и средств коллективной разработки.

Основные задачи дисциплины:

- освоение теоретических основ и современных технологий анализа, проектирования и разработки программного обеспечения;
- овладение практическими навыками проектирования и разработки различных видов программного обеспечения на основе объектно-ориентированного подхода;
- приобретение опыта разработки программных средств средней сложности;
- знакомство с библиотеками классов и инструментальными средствами, используемыми при разработке программного обеспечения.

**10. Место учебной дисциплины в структуре ООП:** Дисциплина относится к Блоку 1, базовая часть. Для ее изучения требуются входные знания из курсов: «Языки и системы программирования», «Объектно-ориентированное программирование», «Алгоритмы и структуры данных», «Базы данных», «Информационные системы». Является основной дисциплиной профессионального цикла. Данная дисциплина является предшествующей для ряда дисциплин профессионального цикла: «Конструирование программного обеспечения», «Тестирование программного обеспечения», «Разработка и анализ требований», «Управление программными проектами».

**11. Планируемые результаты обучения по дисциплине/модулю (знания, умения, навыки), соотнесенные с планируемыми результатами освоения образовательной программы (компетенциями) и индикаторами их достижения:**

Код	Название компетенции	Код(ы)	Индикатор(ы)	Планируемые результаты обучения
ПК-2	Способен выполнять интеграцию программных модулей и компонент, выполнять верификацию программных продуктов	ПК-2.1	Знает технические спецификации программных компонентов и способы их взаимодействия	<p>знать: технические спецификации программных компонентов и способы их взаимодействия</p> <p>уметь: анализировать и интерпретировать технические спецификации программных компонентов для определения способов их взаимодействия</p> <p>владеть (иметь навык(и)): инструментами анализа технических спецификаций и проектирования системного взаимодействия программных компонентов</p>
		ПК-2.2	Знает что такое формализация и алгоритмизация. Имеет навыки формализации и алгоритмизации задача для ПО.	<p>знать: основные концепции формализации и алгоритмизации, а также их применение в контексте разработки программного обеспечения</p> <p>уметь: формализовывать задачи для программного обеспечения и разрабатывать алгоритмы для их решения, учитывая требования и стандарты</p> <p>владеть (иметь навык(и)): инструментами и методами формализации задач и разработки алгоритмов для программного обеспечения</p>
		ПК-2.3	Имеет навыки программирования,	знать: основные принципы программирования, отладки и

			отладки и тестирования программного обеспечения	<p>тестирования программного обеспечения, а также методы и инструменты, используемые в этих процессах</p> <p>уметь: писать программный код, отлаживать его и проводить тестирование для обеспечения корректности и надежности программного обеспечения</p> <p>владеть (иметь навык(и)): языками программирования, средствами отладки и инструментами тестирования программного обеспечения</p>
--	--	--	---	--

**12. Объем дисциплины в зачетных единицах/час. — 3/108.**

**Форма промежуточной аттестации - экзамен.**

### 13. Виды учебной работы

Вид учебной работы	Трудоемкость			
	Всего	По семестрам		
		6 семестр		...
Аудиторные занятия	48	48		
в том числе: лекции	32	32		
практические	-	-		
лабораторные	16	16		
Самостоятельная работа	24	24		
Форма промежуточной аттестации (зачет – 0 час. / экзамен – __ час.)	36	36		
Итого:	108	108		

#### 13.1. Содержание дисциплины

п/п	Наименование раздела дисциплины	Содержание раздела дисциплины
<b>1. Лекции</b>		
1.1	Основные понятия технологии программирования	Индустриальный подход к разработке программного обеспечения. Методы и средства программной инженерии. Жизненный цикл программного продукта. Этапы процесса разработки. Понятие качества программного продукта, основные критерии качества
1.2	Модели процесса разработки программных средств (ПС): прогностические и адаптивные модели	Стратегии разработки и модели процесса разработки. Прогностические и адаптивные модели. Особенности прогностических моделей. Каскадная, инкрементная и спиральная модели процесса разработки ПС. RUP-процесс. Особенности адаптивных моделей. XP-модель и принципы экстремального программирования. Scrum-модель.
1.3	Моделирование предметной области; структурный и объектно-ориентированный анализ	Анализ и моделирование предметной области как основа для разработки требований к ПО. Метод функционального моделирования SADT. Состав функциональной модели. IDEF- и DFD-диаграммы. Принципы и средства объектного моделирования систем. Унифицированный язык моделирования UML. Виды диаграмм. Концептуальный и логический уровни моделирования.
1.4	Проектирование программных средств; CASE-	Архитектурное и детальное проектирование. Основные виды архитектур программных систем. Задачи этапа

	средства архитектурного и детального проектирования	архитектурного проектирования: выявление подсистем и интерфейсов, формирование архитектурных уровней, проектирование структуры потоков управления. Уровень логического (детального) проектирования.
1.5	Особенности объектно-ориентированного проектирования программных систем. Паттерны проектирования.	Проектирование объектно-ориентированных ПС. Проектирование классов и интерфейсов. Шаблоны проектирования. Объектно-ориентированное CASE средство Rational Rose.
1.6	Современные языки программирования: особенности и тенденции развития. Техники написания эффективного программного кода.	Классификация языков программирования: процедурные, объектно-ориентированные и декларативные. Критерии сравнительного анализа языков. Проблемы совместимости компонент, написанных на различных языках программирования.
1.7	Тестирование и отладка программных средств; виды тестирования	Тестирование и отладка программных средств. Виды тестирования. Тестовые наборы и тестовые процедуры. Технологии разработки, ведомые тестированием. Автоматизация процесса тестирования модулей. Инструментальное средство NUnit.
1.8	Проблема контроля версий в процессе разработки ПС	Понятие версии ПС и контроля версий. Автоматизация контроля версий. Утилита Subversion.
1.9	Стандартизация в сфере программной инженерии	Стандартизация в сфере программной инженерии. Национальные и международные стандарты. Стандарты группы ISO/IEC: стандарты на базовые процессы и стандарты оценки уровня зрелости.
1.10	Управление программным проектом	Цели и задачи управления проектом. Планирование проектных задач и распределения работ. Риски, анализ и управление рисками. LOC- и FP-метрики. Оценка проекта на основе метрик.

## 2. Практические занятия

*нет*

## 3. Лабораторные работы

3.1	Основные понятия технологии программирования	Индустриальный подход к разработке программного обеспечения. Методы и средства программной инженерии. Жизненный цикл программного продукта. Этапы процесса разработки. Понятие качества программного продукта, основные критерии качества
3.2	Модели процесса разработки программных средств (ПС): прогностические и адаптивные модели	Стратегии разработки и модели процесса разработки. Прогностические и адаптивные модели. Особенности прогностических моделей. Каскадная, инкрементная и спиральная модели процесса разработки ПС. RUP-процесс. Особенности адаптивных моделей. XP-модель и принципы экстремального программирования. Scrum-модель.
3.3	Моделирование предметной области; структурный и объектно-ориентированный анализ	Анализ и моделирование предметной области как основа для разработки требований к ПО. Метод функционального моделирования SADT. Состав функциональной модели. IDEF- и DFD-диаграммы. Принципы и средства объектного моделирования систем. Унифицированный язык моделирования UML. Виды диаграмм. Концептуальный и логический уровни моделирования.
3.4	Проектирование программных средств; CASE-средства архитектурного и детального проектирования	Архитектурное и детальное проектирование. Основные виды архитектур программных систем. Задачи этапа архитектурного проектирования: выявление подсистем и интерфейсов, формирование архитектурных уровней, проектирование структуры потоков управления. Уровень логического (детального) проектирования.
3.5	Особенности объектно-ориентированного проектирования программных систем. Паттерны проектирования.	Проектирование объектно-ориентированных ПС. Проектирование классов и интерфейсов. Шаблоны проектирования. Объектно-ориентированное CASE средство Rational Rose.
3.6	Современные языки	Классификация языков программирования: процедурные,

	программирования: особенности и тенденции развития. Техники написания эффективного программного кода.	объектно-ориентированные и декларативные. Критерии сравнительного анализа языков. Проблемы совместимости компонент, написанных на различных языках программирования.
3.7	Тестирование и отладка программных средств; виды тестирования	Тестирование и отладка программных средств. Виды тестирования. Тестовые наборы и тестовые процедуры. Технологии разработки, ведомые тестированием. Автоматизация процесса тестирования модулей. Инструментальное средство NUnit.
3.8	Проблема контроля версий в процессе разработки ПС	Понятие версии ПС и контроля версий. Автоматизация контроля версий. Утилита Subversion.
3.9	Стандартизация в сфере программной инженерии	Стандартизация в сфере программной инженерии. Национальные и международные стандарты. Стандарты группы ISO/IEC: стандарты на базовые процессы и стандарты оценки уровня зрелости.
3.10	Управление программным проектом	Цели и задачи управления проектом. Планирование проектных задач и распределения работ. Риски, анализ и управление рисками. LOC- и FP-метрики. Оценка проекта на основе метрик.

### 13.2. Темы (разделы) дисциплины и виды занятий

№ п/п	Наименование темы (раздела) дисциплины	Виды занятий (часов)				Всего
		Лекции	Практические	Лабораторные	Самостоятельная работа	
1	Основные понятия технологии программирования	2	-	1	2	5
2	Модели процесса разработки программных средств (ПС): прогностические и адаптивные модели	4	-	2	3	9
3	Моделирование предметной области; структурный и объектно-ориентированный анализ	4	-	2	3	9
4	Проектирование программных средств; CASE-средства архитектурного и детального проектирования	4	-	2	3	9
5	Особенности объектно-ориентированного проектирования программных систем. Паттерны проектирования.	3	-	2	2	7
6	Современные языки программирования: особенности и тенденции развития. Техники написания эффективного программного кода.	3	-	2	2	7
7	Тестирование и отладка программных средств; виды тестирования	4	-	2	3	9
8	Проблема контроля версий в процессе разработки ПС	3	-	2	2	7
9	Стандартизация в сфере программной инженерии	3	-		2	5
10	Управление программным проектом	2	-	1	2	5
	Итого:	32	-	16	24	72

#### 14. Методические указания для обучающихся по освоению дисциплины

Рекомендуется работа с конспектами лекций, презентационным материалом, выполнение практических проектов, тестов, заданий текущей аттестации. Учебные и методические материалы по дисциплине представлены на сетевом диске внутренней сети факультета.

#### 15. Перечень основной и дополнительной литературы, ресурсов интернет, необходимых для освоения дисциплины

а) основная литература:

№ п/п	Источник
1.	Орлов С.А. Технологии разработки программного обеспечения. Разработка сложных программных систем: учебник для вузов /С.А. Орлов. – СПб.: Питер, 2012. – 527 с.

б) дополнительная литература:

№ п/п	Источник
2.	Кулямин В.В. Технологии программирования. Компонентный подход: учебное пособие/В.В. Кулямин. – М. Бином. Лаборатория знаний. 2007. – 463 с.

в) информационные электронно-образовательные ресурсы (официальные ресурсы интернет)\*:

№ п/п	Ресурс
3.	В.А. Петрухин, Е.М. Лаврищева Методы и средства инженерии программного обеспечения. <a href="http://www.intuit.ru/department/se/swebok/0/">http://www.intuit.ru/department/se/swebok/0/</a>
4.	<a href="http://www.lib.vsu.ru">www.lib.vsu.ru</a> ЗНБ ВГУ

#### 16. Перечень учебно-методического обеспечения для самостоятельной работы

№ п/п	Источник
1.	Контрольные задания и тесты в электронном варианте
2.	Сетевой диск внутренней сети факультета

#### 17. Информационные технологии, используемые для реализации учебной дисциплины, включая программное обеспечение и информационно-справочные системы (при необходимости)

На лабораторных и практических занятиях допускается использовать студентами любой язык программирования и любую среду программирования.

#### 18. Материально-техническое обеспечение дисциплины:

Мультимедийная лекционная аудитория, компьютерные классы факультета компьютерных наук.

## 19. Фонд оценочных средств:

### 19.1. Перечень компетенций с указанием этапов формирования и планируемых результатов обучения

№ п/п	Наименование раздела дисциплины (модуля)	Компетенция(и)	Индикатор(ы) достижения компетенции	Оценочные средства
1	Основные понятия технологии программирования	ПК-2	ПК-2.1, ПК-2.2, ПК-2.3	Лабораторные работы
2	Модели процесса разработки программных средств (ПС): прогностические и адаптивные модели	ПК-2	ПК-2.1, ПК-2.2, ПК-2.3	Лабораторные работы
3	Моделирование предметной области; структурный и объектно-ориентированный анализ	ПК-2	ПК-2.1, ПК-2.2, ПК-2.3	Лабораторные работы
4	Проектирование программных средств; CASE-средства архитектурного и детального проектирования	ПК-2	ПК-2.1, ПК-2.2, ПК-2.3	Лабораторные работы
5	Особенности объектно-ориентированного проектирования программных систем. Паттерны проектирования.	ПК-2	ПК-2.1, ПК-2.2, ПК-2.3	Лабораторные работы
6	Современные языки программирования: особенности и тенденции развития. Техники написания эффективного программного кода.	ПК-2	ПК-2.1, ПК-2.2, ПК-2.3	Лабораторные работы
7	Тестирование и отладка программных средств; виды тестирования	ПК-2	ПК-2.1, ПК-2.2, ПК-2.3	Лабораторные работы
8	Проблема контроля версий в процессе разработки ПС	ПК-2	ПК-2.1, ПК-2.2, ПК-2.3	Лабораторные работы
9	Стандартизация в сфере программной инженерии	ПК-2	ПК-2.1, ПК-2.2, ПК-2.3	Лабораторные работы
10	Управление программным проектом	ПК-2	ПК-2.1, ПК-2.2, ПК-2.3	Лабораторные работы
Промежуточная аттестация форма контроля – экзамен				Курсовая работа, вопросы к экзамену

## 19.2 Описание критериев и шкалы оценивания компетенций (результатов обучения) при промежуточной аттестации

Для оценивания результатов обучения на зачете используются следующие показатели:

- 1) знание учебного материала и владение понятийным аппаратом разработки приложений;
- 2) умение связывать теорию с практикой;
- 3) умение иллюстрировать ответ примерами, фактами, хорошими практиками разработки;
- 4) умение применять методы проектирования, реализации и тестирования приложений, решать задачи тестирования производительности приложений;
- 5) владение способами разрешения противоречий при прохождении жизненного цикла системы.

Для оценивания качества курсовой работы используется 4-балльная шкала: «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

Для оценивания результатов обучения на экзамене используется 4-балльная шкала: «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

Соотношение показателей, критериев и шкалы оценивания результатов обучения.

Критерии оценивания компетенций	Уровень сформированности компетенций	Шкала оценок
Обучающийся в полной мере владеет теоретическими основами дисциплины, способен иллюстрировать ответ примерами, применять теоретические знания для решения практических задач.	<i>Повышенный уровень</i>	<i>Отлично</i>
Ответ на контрольно-измерительный материал не вполне соответствует двум из перечисленных показателей, но обучающийся дает правильные ответы на дополнительные вопросы. Недостаточно продемонстрировано владение системным подходом, или содержатся отдельные пробелы при описании терминологии дисциплины и ее практик.	<i>Базовый уровень</i>	<i>Хорошо</i>
Ответ на контрольно-измерительный материал не соответствует любым трем из перечисленных показателей, обучающийся дает неполные ответы на дополнительные вопросы. Демонстрирует лишь частичные знания, или не умеет связывать теорию с практикой, или имеет неполное представление о системном подходе, допускает существенные ошибки.	<i>Пороговый уровень</i>	<i>Удовлетворительно</i>
Ответ на контрольно-измерительный материал не соответствует любым четырем из перечисленных показателей. Обучающийся демонстрирует отрывочные, фрагментарные знания, допускает грубые ошибки.	–	<i>Неудовлетворительно</i>

## 19.3 Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующие этапы формирования компетенций в процессе освоения образовательной программы

### 19.3.1 Перечень вопросов к экзамену:

1	<i>Что такое технология программирования? Методы и средства разработки программных продуктов?</i>
2	<i>Понятие качества программных продуктов. Критерии качества.</i>
3	<i>Определение качества ПО в стандарте ISO 9126. Аспекты качества, их взаимное влияние.</i>



4	Многоуровневая модель качества ПО в стандарте ISO 9126.
5	Модель жизненного цикла программных средств. Фазы жизненного цикла. Этапы фазы разработки, их содержание.
6	Анализ предметной области: цели и задачи. Модели предметной области. Формальные определения. Классификация моделей.
7	Методология IDEF0, синтаксис IDEF0-моделей.
8	Диаграммы потоков данных (DFD-диаграммы), их использование при моделировании предметной области.
9	Модели процесса разработки программных средств и реализуемые ими стратегии. Прогностические модели (процессы) разработки. Каскадная модель, ее характеристика.
10	Модели процесса разработки программных средств и реализуемые ими стратегии. Прогностические модели (процессы) разработки. Инкрементная модель процесса разработки, ее характеристика.
11	Модели процесса разработки программных средств и реализуемые ими стратегии. Прогностические модели (процессы) разработки. Спиральная модель процесса разработки, ее характеристика.
12	RUP-модель процесса разработки программного средства: принципы и структура.
13	Адаптивные процессы разработки программных средств. XP-модель процесса разработки. Принципы экстремального программирования.
14	Адаптивные процессы разработки программных средств. Scrum-модель процесса разработки.
15	Задачи этапа объектно-ориентированного анализа предметной области. Методика определения границ системы и ключевых абстракций. Пример проведения анализа.
16	Функциональные требования к системе. Способ их представления в виде UML-диаграммы. Пример диаграммы с использованием отношений «расширяет» и «включает».
17	Понятие прецедента и сценария. Пример прецедента, основного и дополнительного сценариев.
18	Нефункциональные требования к системе, их виды. Примеры нефункциональных требований.
19	Концептуальная модель системы: концептуальные классы, системные события и системные операции. Способ их представления в виде UML-диаграмм. Пример концептуального описания прецедента.
20	Диаграммы взаимодействия как элементы концептуальной модели. Синтаксис диаграмм взаимодействия. Примеры диаграмм взаимодействия
21	Цели и задачи этапа проектирования. Понятие модели проектирования, ее отличия от концептуальной модели.
22	Стадии проектирования, их краткая характеристика.
23	Задачи, решаемые на стадии эскизного проектирования.
24	Понятие архитектуры ПС. Проблема выбора архитектуры. Влияние архитектуры на качественные характеристики ПС.
25	Понятие модуля и модульного программирования. Преимущества модульного подхода к разработке ПО.
26	Задачи, решаемые на стадии детального проектирования.
27	Цели и задачи проектирования пользовательского интерфейса.
28	Особенности объектно-ориентированного проектирования.
29	Идентификация методов программных классов. Диаграммы классов, способы отображения отношений ассоциации и зависимости. Пример диаграммы классов.
30	Обязанности программных классов, виды обязанностей. Визуализация распределения обязанностей посредством UML-диаграмм. Примеры диаграмм.
31	Шаблоны проектирования, их классификация. Правила описания шаблонов, примеры шаблонов с их описаниями.
32	Шаблоны анализа (analysis patterns), их классификация.
33	Архитектурные шаблоны (architectural patterns), их классификация.
34	Шаблоны проектирования (design patterns), их примеры.
35	Шаблоны распределения обязанностей, их назначение. Примеры применения.
36	Структурные шаблоны, их назначение. Примеры структурных шаблонов с их описаниями.
37	Тестирование программного средства. Стадии тестирования и их характеристика.
39	Основные принципы тестирования.
40	Тесты и тестовые наборы. Понятие тестового покрытия. Отладочное тестирование. Соотношение структурного и функционального подходов.
41	Структурный подход к формированию тестовых наборов. Пример реализации структурного подхода.
42	Функциональный подход к формированию тестовых наборов. Пример реализации функционального подхода.
43	Интеграционное тестирование. Виды интеграционного тестирования. Критерии полноты тестовых наборов.
44	Регрессионное тестирование. Критерии завершения отладочного тестирования.
45	Восходящая и нисходящая стратегии интеграционного тестирования, механизмы их

	<i>реализации.</i>
46	<i>Системное тестирование. Виды системного тестирования. Критерии полноты тестовых наборов.</i>
47	<i>Особенности объектно-ориентированного тестирования. Расширение области применения тестирования. Критерии тестирования моделей.</i>
48	<i>Особенности методики модульного тестирования объектно-ориентированных систем. Тестирование классов.</i>
49	<i>Особенности методики интеграционного тестирования объектно-ориентированных систем. Тестирование кластеров и потоковое тестирование.</i>
50	<i>Понятие автоматизированного тестирования. Автотесты. Достоинства и недостатки автоматизированного тестирования.</i>
51	<i>Типы автоматизированного тестирования, их цели. Средства автоматизированного тестирования.</i>
52	<i>Утверждения, параметры утверждений. Группы утверждений, классическая и закрытая модель утверждений.</i>
56	<i>Система конкурирующих версий CVS, ее достоинства и недостатки.</i>
57	<i>Система Subversion, ее архитектура; достоинства и недостатки системы.</i>
58	<i>Хранилище, его структура, правки. Команды SVN для работы с хранилищем.</i>
59	<i>Понятия рабочей копии и служебного каталога. Команды SVN для работы с рабочими копиями.</i>
60	<i>Сценарий объединения правок. Конфликты и способы их разрешения.</i>
61	<i>Понятие сборки, манифест сборки. Сборка приложения, системы автоматизации сборки.</i>
62	<i>Утилита NAnt, файл сборки и его структура.</i>
63	<i>Цели, зависимость целей, описание целей.</i>
64	<i>Документирование процесса разработки. Типы документов управления.</i>
65	<i>Документирование программного продукта. Документация сопровождения, ее назначение и состав.</i>
66	<i>Документирование программного продукта. Пользовательская документация, ее назначение и состав.</i>
67	<i>Генератор документации Sandcastle, его назначение и принцип работы.</i>
68	<i>Руководство проектом и особенности проектной деятельности.</i>
69	<i>Проектная команда, группы и роли в проектной команде.</i>
70	<i>Критерии оценивания проектов, шкалы ценности проекта.</i>
71	<i>Риски, их ранжирование, управление рисками.</i>
72	<i>Базовое расписание проекта, точки контроля, распараллеливание работ.</i>
73	<i>Способы контроля хода выполнения проекта: меры и метрики. Виды метрик</i>

### **19.3.2 Перечень лабораторных работ**

Проектирование, реализация, тестирование и приложения для предметной области по выбору обучающегося, а также подготовка и защита курсовой работы.

### **19.4. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций**

Оценка знаний, умений и навыков, характеризующая этапы формирования компетенций в рамках изучения дисциплины, осуществляется в ходе текущей и промежуточной аттестаций.

Текущая аттестация проводится в соответствии с Положением о текущей аттестации

обучающихся по программам высшего образования Воронежского государственного университета. Текущая аттестация проводится в форме демонстрации стадий процесса создания практического проекта. Критерии оценивания приведены выше.

Промежуточная аттестация проводится в соответствии с Положением о промежуточной аттестации обучающихся по программам высшего образования. Контрольно-измерительные материалы промежуточной аттестации включают в себя теоретические вопросы, позволяющие оценить уровень полученных знаний.

При оценивании используются количественные и качественные шкалы оценок. Критерии оценивания приведены выше.

**20. Приведённые ниже задания рекомендуется использовать при проведении диагностических работ для оценки остаточных знаний по дисциплине**

12 вопросов с выбором ответа:

1. Какой этап процесса проектирования программного обеспечения включает определение функциональных и нефункциональных требований к продукту?

- a) Этап проектирования архитектуры
- b) Этап детального проектирования
- c) Этап программирования
- d) Этап анализа требований

Ответ: d) Этап анализа требований

2. Какая основная цель проектирования архитектуры на этапе проектирования программного обеспечения?

- a) Разработка спецификаций требований
- b) Определение функций и взаимодействия компонентов
- c) Создание детального плана работы
- d) Написание программного кода

Ответ: b) Определение функций и взаимодействия компонентов

3. Какой метод документирования наиболее часто используется на этапе проектирования программного обеспечения?

- a) UML (Unified Modeling Language)
- b) HTML (HyperText Markup Language)
- c) CSS (Cascading Style Sheets)
- d) JSON (JavaScript Object Notation)

Ответ: a) UML (Unified Modeling Language)

4. Какие методы документирования применяются на этапе детального проектирования при проектировании программного обеспечения?

- a) Текстовые, графические и нотационные
- b) Текстовые и устные
- c) Главные, побочные и дополнительные
- d) Качественные и количественные

Ответ: a) Текстовые, графические и нотационные

5. Какова цель оценки требований на этапе анализа при проектировании программного обеспечения?

- a) Определение трудоемкости и времени выполнения заданий
- b) Оценка рисков и возможных компромиссных решений
- c) Определение приоритетов и возможностей реализации требований
- d) Создание оптимального интерфейса пользователя

Ответ: c) Определение приоритетов и возможностей реализации требований

6. Какой метод документирования применяется на этапе проектирования архитектуры?

- a) Базы данных
- b) Программное обеспечение
- c) UML

d) Текстовые документы

Ответ: c) UML

7. Какой этап проектирования предшествует этапу программирования?

a) Этап анализа требований

b) Этап проектирования архитектуры

c) Этап детального проектирования

d) Этап сопровождения

Ответ: c) Этап детального проектирования

8. Какие основные этапы включает процесс проектирования программного обеспечения?

a) Анализирование, обновление, тестирование

b) Испытание, обновление, разработка

c) Развертывание, проектирование, коммуникация

d) Анализ требований, проектирование архитектуры, детальное проектирование, программирование, тестирование и сопровождение.

Ответ: d) Анализ требований, проектирование архитектуры, детальное проектирование, программирование, тестирование и сопровождение.

9. Какие методы документирования используются на этапе детального проектирования?

a) Текстовые и устные

b) Нотационные и графические

c) Системы контроля версий

d) HTML и CSS

Ответ: b) Нотационные и графические

10. Какова цель оценки требований на этапе анализа при проектировании программного обеспечения?

a) Определение трудоемкости и времени выполнения заданий

b) Создание оптимального интерфейса пользователя

c) Оценка рисков и возможных компромиссных решений

d) Определение приоритетов и возможностей реализации требований

Ответ: d) Определение приоритетов и возможностей реализации требований

12. При проектировании программного обеспечения, какой этап предшествует этапу анализа требований?

a) Этап детального проектирования

b) Этап программирования

c) Этап проектирования архитектуры

d) Этап сопровождения

Ответ: c) Этап проектирования архитектуры.

13. Какой метод документирования наиболее используется на этапе детального проектирования?

a) Графические диаграммы и схемы

b) Технические задания и планы

- c) UML
- d) Спецификации требований

Ответ: а) Графические диаграммы и схемы.

14. Какова цель проектирования архитектуры на этапе проектирования программного обеспечения?

- а) Формирование концепции продукта
- б) Определение деталей взаимодействия компонентов
- с) Написание и отладка кода
- д) Оценка рисков

Ответ: б) Определение деталей взаимодействия компонентов.

15. Какое основное назначение этапа сопровождения при проектировании программного обеспечения?

- а) Определение технических и экономических возможностей
- б) Разработка тестового плана
- с) Устранение выявленных ошибок и дефектов
- д) Определение функциональных и нефункциональных требований

Ответ: с) Устранение выявленных ошибок и дефектов.

3 вопроса с кратким ответом:

1. Что такое технология программирования и каковы ее основные этапы развития? Какие проблемы могут возникать при разработке сложных программных систем?

Ответ: Технология программирования является системой методов, правил и приемов, используемых при создании программного обеспечения. Основными этапами развития технологии программирования являются этап низкоуровневого программирования, этап структурированного программирования и этап объектно-ориентированного программирования. Проблемы, которые могут возникать при разработке сложных программных систем, включают управление проектом, анализ требований, интеграцию различных модулей и сопровождение программного обеспечения.

2. Какие модели разработки программного обеспечения существуют? Какие преимущества и недостатки свойственны каждой из этих моделей?

Ответ: Существуют различные модели разработки программного обеспечения, такие как каскадная, с промежуточным контролем, спиральная, RAD-технологии и т.д. Каждая модель имеет свои преимущества и недостатки. Например, каскадная модель позволяет упорядоченно выполнять каждую фазу разработки, но может привести к необходимости начать проект заново, если в процессе работы обнаружатся ошибки. Спиральная модель предполагает последовательное повышение уровня сложности проекта, что может помочь в контроле рисков, но может привести к повышению цены и срокам выполнения проекта.

3. Как выполняется тестирование программного обеспечения и каким образом оценивается его качество?

Ответ: Тестирование программного обеспечения предполагает запуск программы для проверки работоспособности, соответствия требованиям и выявления ошибок.

Тестирование включает проведение модульных, интеграционных, системных и приемочных тестов. Для оценки качества ПО используются различные метрики, такие как количество найденных ошибок, время реакции программы, количество памяти, используемой программой, и т.д.

3 вопроса с развёрнутым ответом:

1) Какие основные этапы включает процесс проектирования программного обеспечения?

Процесс проектирования программного обеспечения включает следующие основные этапы: анализ требований, проектирование архитектуры, детальное проектирование, программирование, тестирование и сопровождение. На этапе анализа требований определяются функциональные и нефункциональные требования к программному продукту, на основе которых формируется концепция и список функций. На этапе проектирования архитектуры создается общая концепция построения системы, определяются ее компоненты и взаимодействие между ними. На этапе детального проектирования разрабатываются детали взаимодействия между компонентами системы, а также проектируется интерфейс пользователя. После этого происходит написание и отладка кода, тестирование продукта, а также его сопровождение.

2) Какие методы документирования используются на этапе проектирования программного обеспечения?

На этапе проектирования программного обеспечения используются различные методы документирования, включая текстовые, графические и нотационные. Для написания текстовых документов используются спецификации требований, технические задания, планы тестирования и т.д. Графические документы включают диаграммы, схемы, графики и т.д., которые используются для визуализации архитектуры системы, взаимодействия компонентов и т.д. Важной ролью в документировании являются нотационные методы, такие как UML (Unified Modeling Language), BPMN (Business Process Model and Notation) и т.д. Они позволяют описать систему на более высоком уровне абстракции и использовать единый стандарт при коммуникации между членами команды. Документирование на этапе проектирования играет важную роль в обеспечении качества продукта, упрощении процесса сопровождения и повышении эффективности командной работы.

3) Как оцениваются требования на этапе анализа при проектировании программного обеспечения?

Оценка требований на этапе анализа при проектировании программного обеспечения проводится в несколько этапов. Вначале производится анализ и классификация требований по их важности и приоритету. Затем определяются технические и экономические возможности и ограничения при разработке и реализации требований. Оценка также включает оценку трудоемкости и времени выполнения заданий, анализ стоимости реализации, рисков и возможных компромиссных решений. В конечном итоге, на основе совокупности данных, проводится принятие решения о приоритетах и возможностях реализации конкретных требований в контексте бизнес-целей организации.

Выполнение этого этапа позволяет улучшить понимание целей и требований заказчика, создание оптимального плана проекта и избежание неожиданных проблем в ходе разработки.