

Автономная некоммерческая организация высшего образования «Университет информационных технологий и инноваций»

(АНО ВО УИТИ)

Утверждаю:

Ректор АНО ВО УИТИ Хутинаева С.З.

Сведения об электронной подписи  
Подписано: Хутинаева Светлана Зураповна  
Должность: ректор  
Пользователь: skhutinaeva

Протокол заседания Учёного совета АНО ВО УИТИ № 01 от 26.02.2026 г.

Утверждено на заседании кафедры информатики

Протокол № 01/ ИТ от 24.02.2026 г.

<b>РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ</b>
<b>Б1.О.04 МОДУЛЬ ОБЩЕПРОФЕССИОНАЛЬНОЙ ПОДГОТОВКИ</b>
<b>Б1.О.04.05 ТЕОРИЯ И ТЕХНОЛОГИЯ ПРОГРАММИРОВАНИЯ</b>
<b>Для направления подготовки: 27.03.03 Системный анализ и управление (уровень бакалавриат)</b>
<b>Типы задач профессиональной деятельности: проектно-технологический, научно-исследовательский, эксплуатационно-технологический</b>
<b>Направленность (профиль): Системный анализ и управление бизнес-процессами</b>
<b>Форма обучения: очная</b>

г. Владикавказ, 2026

## СОДЕРЖАНИЕ

1. ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ (МОДУЛЮ) .....	3
Перечень компетенций, формируемых дисциплиной (модулем) в процессе освоения образовательной программы .....	3
2. РЕЗУЛЬТАТЫ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ) .....	3
3. ОБЪЕМ ДИСЦИПЛИНЫ И РАСПРЕДЕЛЕНИЕ ВИДОВ УЧЕБНОЙ РАБОТЫ	ПО
СЕМЕСТРАМ .....	3
5. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПО ДИСЦИПЛИНЕ .....	5
6. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ .....	5
6.1. Рекомендуемая литература .....	5
7. ИНФОРМАЦИОННО-СПРАВОЧНОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ .....	5
7.1 Программное обеспечение Университета – часть электронной информационно-образовательной среды: .....	5
8. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ .....	6
9. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ .....	6

## 1. ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

Перечень компетенций, формируемых дисциплиной (модулем) в процессе освоения образовательной программы

Код и наименование компетенции	Индикаторы достижения компетенции	Результаты обучения
ОПК-10. Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности	ОПК-10.2. Использует принципы работы и достижения современных информационных технологий для решения профессиональных задач	<b>Знает:</b> принципы работы и достижения современных информационных технологий для решения профессиональных задач <b>Умеет:</b> применять принципы работы и достижения современных информационных технологий для решения профессиональных задач <b>Владеет:</b> навыком применения принципов работы и достижений современных информационных технологий для решения профессиональных задач

## 2. РЕЗУЛЬТАТЫ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)

**Цель:** Освоение фундаментальных принципов и понимание базовых концепций программирования (алгоритмы, структуры данных, парадигмы программирования, архитектура программ).

### **Задачи:**

- Изучение современных технологий и инструментов программирования, средств разработки, контроля версий и коллективной работы.
- Развитие алгоритмическое мышление, формализацию задач, проектирования эффективных алгоритмов и анализ их сложности.
- Приобретение навыков проектирования и разработки ПО — умение создавать надёжные, масштабируемые и поддерживаемые программные системы, используя стандарты и лучшие практики индустрии.
- Понимание методологий разработки - гибких и традиционных подходов к управлению проектами.
- Подготовка к решению практических задач применения полученных знаний в реальных условиях, включая отладку, тестирование, оптимизацию и документирование кода

## 3. ОБЪЕМ ДИСЦИПЛИНЫ И РАСПРЕДЕЛЕНИЕ ВИДОВ УЧЕБНОЙ РАБОТЫ ПО СЕМЕСТРАМ

Общая трудоемкость дисциплины «Теория и технология программирования» составляет: 3 з.е. / 108 час.

Вид учебной работы								
Аудиторные занятия				Самостоятельная работа		Промежуточная аттестация		
Аудиторные занятия в том числе:	Лекции	Практические занятия	Лабораторные работы	Самостоятельная работа в том числе:	часы на выполнение КР / КП	Вид	Семестр	Трудоемкость (час.)
Всего число часов и (или) зачетных единиц (по формам обучения)								
Очная форма обучения								
36	18	18	-	72	-	Зачет с оценкой	2	-
Общая трудоемкость з.е. / час.: 3 з.е. / 108 час.								

#### 4. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

Тема 1. Знакомство со средой программирования, синтаксисом языка Python. Место и значимость языка Python среди других языков программирования	Лекции ч.	Практические занятия ч.	Лабораторные работы ч.	Самостоят. работа ч.
	2	2	-	14
Среда для Python, содержит инструменты для написания и отладки кода, контроля версий, создания виртуальных сред и тестирования приложений. то один из самых простых языков кодирования для изучения, простой в использовании и общепринятый.				
Тема 2. Виды циклов	Лекции ч.	Прак зан ч.	Лаб раб ч.	Самост раб ч.
	4	4	-	14
Цикличность является естественным законом природы. В природе все повторяется: день сменяется ночью, лето осенью, а затем зимой. В программировании же циклы — «краеугольный камень» любого языка. Благодаря им программист может реализовать многократное выполнение определенных действий, попутно избежав дублирования кода.				
Тема 3. Структуры данных. Массивы и матрицы в языке Python. Файловый ввод и вывод	Лекции ч.	Прак зан ч.	Лаб раб ч.	Самост раб ч.
	4	4	-	15
Способы ввода двумерных массивов. Какими способами можно задавать данные в массиве. Понятие двумерные массивы, создание и управление.				
Тема 4. Объектно-ориентированное	Лекции ч.	Прак зан ч.	Лаб раб ч.	Самост раб ч.
	4	4	-	15

программирование. Классы и задачи. Подключение и использование библиотек	Подход к созданию программ, основанный на представлении программы в виде совокупности взаимодействующих объектов. ООП помогает структурировать код так, чтобы у каждого элемента были определённые свойства и действия. Классы, набор полей, определение алгоритмов и наследование.
--	---

Тема 5. Динамическое программирование. Игровые модели	Лекции ч.	Прак зан ч.	Лаб раб ч.	Самост раб ч.
	4	4	-	14
Алгоритмы моделирования кратчайших путей между вершинами (построение дерева решений, алгоритм Дейкстры, метод динамического программирования). Элементы теории игр (выигрышной стратегией).				

## 5. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПО ДИСЦИПЛИНЕ

Примерный фонд оценочных средств представлен в Приложении 1.

## 6. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

### Рекомендуемая литература

1. Терехов, А. Н. Технология программирования: учебное пособие / А. Н. Терехов. — 5-е изд. — Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2024. — 148 с. — ISBN 978-5-4497-3349-8. — Текст: электронный // Цифровой образовательный ресурс IPR SMART: [сайт]. — URL: <https://www.iprbookshop.ru/142308.html>

2. Роганов, Е. А. Основы информатики и программирования: учебное пособие / Е. А. Роганов. — 4-е изд. — Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2025. — 390 с. — ISBN 978-5-4497-0908-0. — Текст: электронный // Цифровой образовательный ресурс IPR SMART: [сайт]. — URL: <https://www.iprbookshop.ru/146361.html>

3. Непейвода, Н. Н. Стили и методы программирования: учебное пособие / Н. Н. Непейвода. — 4-е изд. — Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2025. — 295 с. — ISBN 978-5-4497-0938-7. — Текст: электронный // Цифровой образовательный ресурс IPR SMART: [сайт]. — URL: <https://www.iprbookshop.ru/146400.html>

4. Городняя, Л. В. Основы функционального программирования: учебное пособие / Л. В. Городняя. — 4-е изд. — Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2025. — 246 с. — ISBN 978-5-4497-0932-5. — Текст: электронный // Цифровой образовательный ресурс IPR SMART: [сайт]. — URL: <https://www.iprbookshop.ru/146377.html>

## 7. ИНФОРМАЦИОННО-СПРАВОЧНОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

### 7.1 Лицензионное программное обеспечение

- Microsoft Windows 10/11.
- Modelio / StarUML (Средства моделирования систем (UML));
- ELMA365 (Отечественное ПО для автоматизации бизнес-процессов);
- RStudio / Anaconda (Python) (Среды для системного анализа данных);
- PyCharm Community Edition (Среда разработки (версия Community)).

## 7.2 Свободно распространяемое программное обеспечение:

- Astra Linux Common Edition (отечественное ПО)
- LibreOffice (свободно распространяемое ПО (Open Source))
- Яндекс.Браузер (отечественное ПО)
- 7-Zip
- PostgreSQL/pgAdmin [Система управления базами данных; свободно распространяемое ПО]

## 7.3 Перечень современных профессиональных баз данных, информационных справочных систем и ресурсов сети Интернет:

1. <https://ro-edu.ru/> - Медиалпортал «Российское образование»
2. <http://www.iprbookshop.ru> - Электронно-библиотечная система IPRSmart (ЭБС IPRSmart) –электронная библиотека по всем отраслям знаний
3. <https://www.elibrary.ru/> - электронно-библиотечная система eLIBRARY.RU, крупнейшая в России электронная библиотека научных публикаций
4. <https://cyberleninka.ru/> - научная электронная библиотека КиберЛенинка
5. <https://gufo.me/> - справочная база энциклопедий и словарей
6. <http://www.consultant.ru/> - справочная правовая система КонсультантПлюс
7. <https://www.garant.ru/> - справочная правовая система Гарант
8. <https://rosstat.gov.ru/emiss> Единая межведомственная информационно-статистическая система (ЕМИСС) Государственная база статистических данных
9. <https://minfin.gov.ru/ru/performance/audit/standarts/international/documents/?ysclid=mn6p22hks7190904011> - База данных международных стандартов аудита (МСА) [Профессиональный ресурс на сайте IFAC;
10. [https://sroaas.ru/auditor/pravila\\_i\\_standarty/standarty-audita/](https://sroaas.ru/auditor/pravila_i_standarty/standarty-audita/) - База данных международных стандартов аудита (МСА) на сайте МФБ (ifac.org) — первоисточники для аудиторской деятельности

## 8. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

При реализации образовательной программы для освоения учебной дисциплины используются следующие компоненты материально-технической базы Университета:

1. Аудиторный фонд.
2. Материально-технический фонд.
3. Библиотечный фонд.

Аудиторный фонд представляет собой аудитории для проведения учебных занятий, в том числе, лекционных занятий, практических занятий/лабораторных работ.

Материально-технический фонд представлен учебной мебелью и соответствующим оборудованием, обеспечивающим освоение учебной дисциплины.

Библиотечный фонд обеспечивает доступ каждого обучающегося к электронно-библиотечной системе, современным профессиональным базам, информационно-справочным системам, информационным ресурсам сети Интернет, указанным в рабочей программе дисциплины.

#### **Перечень материально-технического обеспечения по дисциплине:**

##### **Аудитория для проведения учебных занятий:**

Комплект специализированной учебной мебели, отвечающий всем установленным нормам и требованиям: столы, стулья. Персональные компьютеры с программным обеспечением, с возможностью подключения к сети «Интернет». Шкаф книжный, стеллаж, шкаф книжный, стеллаж, доска передвижная поворотная магнитная (маркерная), тумба, доска передвижная магнитная (маркерная).

Рабочее место преподавателя: стол, стул, персональный компьютер с программным обеспечением, с возможностью подключения к сети «Интернет».

##### **Помещение для самостоятельной работы обучающихся:**

Комплект специализированной учебной мебели, отвечающий всем установленным нормам и требованиям: столы, стулья.

Персональные компьютеры с программным обеспечением, с возможностью подключения к сети «Интернет».

##### **Аудитория для проведения учебных занятий для обучающихся с ограниченными возможностями здоровья, инвалидов:**

Комплект специализированной мебели, отвечающий всем установленным нормам и требованиям для обучающихся с ограниченными возможностями здоровья, инвалидов: столы, стулья, инвалидное кресло-коляска. Гарнитура, информационная система «Исток» - для слабослышащих, клавиатура Брайля, шкаф книжный.

Рабочее место преподавателя: стол, стул, тумба, персональный компьютер с программным обеспечением, с возможностью подключения к сети «Интернет».

## **9. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ**

Обучение по дисциплине предполагает освоение учебного материала на аудиторных занятиях и в ходе самостоятельной работы. Аудиторные занятия проходят в форме лекций и практических занятий/лабораторных работ.

Для успешного освоения дисциплины рекомендуется придерживаться системного подхода к учебному процессу. Просматривать все лекции, так как они формируют теоретический каркас дисциплины и помогают выстроить логику взаимосвязи ключевых понятий. Рекомендуется вести конспект лекции, с выделением основных идей, вопросов для уточнения и собственных ассоциаций — это поможет в подготовке к активной работе на практических занятиях. На семинарских и практических занятиях целесообразно участвовать в дискуссиях, аргументируя свою позицию и анализируя позиции коллег.

При подготовке к работе во время проведения практических/ лабораторных занятий следует обратить внимание на следующие моменты: на процесс предварительной подготовки, на работу во время занятия, обработку полученных результатов, исправление полученных замечаний.

Предварительная подготовка к практическому/лабораторному занятию заключается в изучении теоретического материала в отведенное для самостоятельной работы время, ознакомление с инструктивными материалами с целью осознания задач практического занятия/лабораторной работы, техники безопасности при работе с оборудованием.

Самостоятельная работа является равноправной частью обучения: целесообразно изучать рекомендованную литературу, дополняя лекционный материал аналитическими источниками и современными исследованиями. Рекомендуется выделять время на систематизацию знаний — составление схем, таблиц, глоссария терминов значительно облегчит подготовку к промежуточной аттестации.

При выполнении самостоятельных заданий целесообразно сфокусироваться на глубине проработки темы и умении применять знания к анализу конкретных ситуаций. Рекомендуется использовать цифровые образовательные ресурсы, современные профессиональные базы, электронные библиотечные системы и информационно-справочные системы для расширения информационной базы.

Рекомендуется регулярно проводить самодиагностику: формулировать ответы на ключевые вопросы без опоры на конспекты, чтобы выявить слабые места. Целесообразно готовиться к занятиям заранее, знакомясь с темой — это позволяет участвовать в учебном процессе на уровне диалога, а не пассивного восприятия.

Успешное освоение дисциплины возможно только при синтезе всех форм работы: лекции задают направление, практические занятия/лабораторные работы развивают умения и навыки, а самостоятельная работа формирует устойчивые компетенции. Необходимо подходить к обучению как к осознанному проектированию собственного интеллектуального развития, а не как к формальному выполнению требований учебного плана.

Автономная некоммерческая организация высшего образования «Университет информационных технологий и инноваций»

(АНО ВО УИТИ)

<b>ФОНД ОЦЕНОЧНЫХ СРЕДСТВ</b>
Текущего контроля и промежуточной аттестации по дисциплине (модулю)
<b>Б1.О.04.05 ТЕОРИЯ И ТЕХНОЛОГИЯ ПРОГРАММИРОВАНИЯ</b>
Для направления подготовки: 27.03.03 Системный анализ и управление (уровень бакалавриат)
Типы задач профессиональной деятельности: проектно-технологический, научно-исследовательский, эксплуатационно-технологический
Направленность (профиль): Системный анализ и управление бизнес-процессами
Форма обучения: очная

г. Владикавказ, 2026

## ТИПОВЫЕ КОНТРОЛЬНЫЕ ЗАДАНИЯ ТЕКУЩЕГО КОНТРОЛЯ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ

### Примерные темы для практических занятий

- 1.. Среда для Python, содержит инструменты для написания и отладки кода, контроля версий, создания виртуальных сред и тестирования приложений. Это один из самых простых языков кодирования для изучения, простой в использовании и общепринятый.
2. Цикличность является естественным законом природы. В природе все повторяется: день сменяется ночью, лето осенью, а затем зимой. В программировании же циклы — «краеугольный камень» любого языка. Благодаря им программист может реализовать многократное выполнение определенных действий, попутно избежав дублирования кода.
3. Способы ввода двумерных массивов. Какими способами можно задавать данные в массиве. Понятие двумерные массивы, создание и управление.
4. Подход к созданию программ, основанный на представлении программы в виде совокупности взаимодействующих объектов. ООП помогает структурировать код так, чтобы у каждого элемента были определённые свойства и действия. Классы, набор полей, определение алгоритмов и наследование.
5. Алгоритмы моделирования кратчайших путей между вершинами (построение дерева решений, алгоритм Дейкстры, метод динамического программирования). Элементы теории игр (выигрышной стратегией).

### Примерные темы рефератов

1. Архитектурные стили веб-приложений.
2. Базы данных в современной разработке.
3. Методологии разработки программного обеспечения. Эволюция подходов к управлению проектами
4. Принципы SOLID и чистой архитектуры Роберта Мартина. Практическая реализация и выгоды для проекта.
5. Принципы реактивного программирования и их реализация в современных фреймворках
6. Проблемы параллельного и многопоточного программирования. Механизмы синхронизации: мьютексы, семафоры, мониторы.
7. Статическая и динамическая типизация: "война" языков. Теория типов данных и ее практическое значение.
8. Теория сложности алгоритмов: классы P, NP, NP-полные задачи. Практическая значимость для разработчика.
9. Технологии и инструменты контроля версий: глубокий разбор Git и системы GitHub/GitLab
10. Эволюция языков программирования: от машинных кодов до языков высокого уровня. (Исторический обзор и ключевые парадигмы)

### Примеры тестовых заданий

1. Архитектурный стиль, при котором приложение разбивается на множество мелких, слабо связанных сервисов, называется:

- а) Монолитная архитектура
- б) Микросервисная архитектура
- в) Событийно-ориентированная архитектура
- г) Многослойная архитектура

Ответ: б

2. Что из перечисленного является основным принципом функционального программирования?

- а) Изменяемое состояние
- б) Наследование классов
- в) Чистые функции и иммутабельность данных
- г) Использование циклов for и while

Ответ: в

3. Что проверяет «Принцип подстановки Лисков» (LSP)?

- а) Наследующий класс должен дополнять, а не замещать функциональность базового класса
- б) Объекты в программе должны быть заменяемыми на экземпляры их подтипов без изменения правильности программы
- в) Класс должен иметь только одну причину для изменения
- г) Интерфейсы должны быть узкоспециализированными

Ответ: б

4. Инструмент, который автоматически анализирует исходный код на предмет соответствия стандартам, потенциальных ошибок и «запахов кода», называется:

- а) Компилятор
- б) Интерпретатор
- в) Статический анализатор кода
- г) Профайлер

Ответ: в

5. Методология разработки, основанная на коротких итерациях (спринтах) и регулярном планировании, называется:

- а) Waterfall (Водопадная модель)
- б) Scrum
- в) RUP (Rational Unified Process)
- г) Spiral (Спиральная модель)

Ответ: б

6. Что из перечисленного НЕ является этапом в конвейере CI/CD?

- а) Непрерывное развертывание
- б) Непрерывное тестирование
- в) Непрерывное документирование
- г) Непрерывная интеграция

Ответ: в

7. Какая из перечисленных структур данных работает по принципу «первым пришел — первым вышел»?

- а) Стек
- б) Очередь
- в) Множество
- г) Куча

Ответ: б

8. Алгоритмическая сложность  $O(\log n)$  характерна для:

- а) Линейного поиска в неотсортированном массиве
- б) Пузырьковой сортировки
- в) Бинарного поиска в отсортированном массиве
- г) Вычисления факториала числа

Ответ: в

9. Какой инструмент используется для изоляции приложения и его зависимостей в единый переносимый пакет?

- а) Jenkins
- б) Docker
- в) Kubernetes
- г) Ansible

Ответ: б

10. Какой протокол/подход НЕ используется для создания веб-API?

- а) REST
- б) GraphQL
- в) SOAP
- г) FTP

Ответ: г

### Примеры вопросов для зачета с оценкой

1. Алгоритмическая сложность. Объясните разницу между временной сложностью  $O(n)$  и  $O(n^2)$ . Приведите конкретный пример алгоритма для каждого случая и опишите, как будет меняться время его выполнения при увеличении объема входных данных в 10 раз.
2. Сетевое взаимодействие. Сравните архитектурные стили REST и GraphQL для построения API. В чем основное концептуальное различие в подходе к данным? Приведите сценарий, где GraphQL был бы более эффективным выбором.
3. Контейнеризация. Объясните, какую фундаментальную проблему решает технология контейнеризации. Чем контейнер принципиально отличается от виртуальной машины с точки зрения архитектуры и производительности?
4. Многопоточность. Что такое состояние гонки в многопоточном программировании? Объясните на простом примере (например, инкремент счетчика). Какой механизм синхронизации вы бы использовали для его предотвращения и почему?
5. Непрерывная интеграция и поставка (CI/CD). Опишите типичный пайплайн CI/CD. Какие основные этапы он включает какую цель преследует каждый из них?
6. Безопасность. Что такое SQL-инъекция? Объясните механизм этой уязвимости на примере строки кода. Каковы основные методы защиты от нее?

7. Функциональное программирование. Что такое чистая функция? Назовите два ее обязательных свойства. Какие преимущества дает использование чистых функций с точки зрения тестирования и надежности кода?
8. Проектирование интерфейсов. Раскройте суть Принципа разделения интерфейса. Приведите пример «толстого» интерфейса, который его нарушает, и покажите, как его правильно разделить.
9. Поиск и сортировка. Опишите алгоритм быстрой сортировки. Какова его средняя и худшая временная сложность? От чего зависит выбор опорного элемента и как он влияет на производительность?
10. Системы контроля версий. Объясните, в чем ключевое отличие централизованных и распределенных систем контроля версий. Какие преимущества дает распределенная модель для командной разработки и workflow?
11. Парадигмы программирования. Сравните императивную и функциональную парадигмы программирования. Назовите по два ключевых признака каждой, их сильные и слабые стороны. Приведите пример задачи, для которой лучше подходит функциональный подход.
12. Паттерны проектирования. Опишите назначение и механизм работы паттерна «Наблюдатель». Приведите пример его практического использования в разработке пользовательских интерфейсов или в любом известном вам фреймворке.
13. Тестирование ПО. Дайте определения модульному и интеграционному тестированию. Чем они принципиально отличаются по цели и объекту тестирования? Почему оба типа важны в современной разработке?
14. Архитектура приложений. Опишите ключевые характеристики микросервисной архитектуры. Назовите три основных преимущества и три сложности/проблемы, которые она introduces по сравнению с монолитной архитектурой.
15. Базы данных. Объясните, что такое транзакция в контексте реляционных СУБД. Раскройте смысл акронима ACID. Почему эти свойства критически важны для банковских или финансовых систем?

#### **Критерии оценивания результатов текущего контроля**

1. Оценка прохождения практических занятий производится по шкале «зачтено» / «не зачтено».
2. Оценка подготовки реферата производится по шкале «зачтено» / «не зачтено».
3. Оценка выполнения тестовых заданий формируется следующим образом:
  - оценка «отлично» - 85-100% правильных ответов;
  - оценка «хорошо» - 70-84% правильных ответов;
  - оценка «удовлетворительно» - 40-69% правильных ответов;
  - оценка «неудовлетворительно» - менее 39% правильных ответов.

#### **Критерии оценивания результатов при проведении промежуточной аттестации**

Знания обучающихся оцениваются по 4-балльной шкале при проведении экзаменов и зачетов с оценкой:

*«отлично»,*  
*«хорошо»,*  
*«удовлетворительно»*  
*«неудовлетворительно»)*

или 2-балльной шкале при проведении зачета:

*«зачтено»,*  
*«не зачтено»*

#### **Описание критериев оценивания:**

##### **1. «Отлично» или «зачтено»**

- а) Обоснованные объемные ответы на вопросы. Обучающийся иллюстрирует выводы фактами, приводит данные из источников.
- б) Обучающийся успешно применяет знание теории для реализации практической части дисциплины. Выполненные задания соответствуют высокому уровню качества, включая использование правильных форматов, методологий и инструментов.
- в) Обучающийся умеет анализировать и оценивать нюансы тематики, демонстрируя способность к критическому мышлению и самостоятельному исследованию.

##### **2. «Хорошо» или «зачтено»**

- а) Обучающийся дает достаточно полные ответы на вопросы с учетом основных направлений темы. Ответы обучающегося имеют четкую структуру и логически связаны.
- б) Обучающийся применяет теоретические знания в практических заданиях. Выполнение задания в целом соответствует требованиям, допустимы некоторые недочеты или неточные выводы по полученным результатам.
- в) Обучающийся демонстрирует хорошее понимание вопроса, знает основные аспекты тематики. Ответы обучающегося содержат достаточно информации, но допустимы недостаточно глубокие суждения.

##### **3. «Удовлетворительно» или «зачтено»**

- а) Ответы на вопросы неполные, не охватывают все стороны тематики и не всегда структурированы или логически связаны. Обучающийся делает верные выводы, но они недостаточно аргументированы или основаны на поверхностном понимании предмета вопроса.
- б) Обучающийся способен использовать теоретические знания в практических заданиях, но недостаточно уверен в верности примененных методов и точности в их выполнении. Выполненное задание может содержать некоторые ошибки, недочеты или расхождения.
- в) Обучающийся охватывает большинство основных сторон темы вопроса, но демонстрирует неполное или поверхностное их понимание, дает недостаточно развернутые объяснения.

##### **4. «Неудовлетворительно» или «не зачтено»**

- а) Обучающийся отвечает на вопросы неполно, не раскрывает основных направлений темы. Ответы обучающегося не структурированы, не связаны с вопросом, отсутствует логика изложения. Выводы, представляют простые утверждения без анализа или четкой аргументации.

б) Обучающийся не умеет переносить теоретические знания в практическую плоскость и не способен применять их для выполнения задания. Выполненное задание содержит много ошибок, а его результаты не соответствуют поставленным требованиям и (или) неправильно интерпретируются.

в) Ответ обучающегося фрагментарный или отрывочный, не включает анализ рассматриваемого вопроса, пропущены важные детали и связи, поверхностный.