

**Федеральное государственное образовательное бюджетное
учреждение высшего образования
«Финансовый университет при Правительстве Российской Федерации»
(Финуниверситет)**

Владикавказский филиал Финуниверситета

Кафедра «Математика и информатика»

СОГЛАСОВАНО

ООО «Экспертно-аналитические
системы»

УТВЕРЖДАЮ

Директор филиала

Генеральный директор


И.В. Игнатович
«27» апреля 2026 г.


Т.А. Хубаев
«27» апреля 2026 г.

А.М. Кумаритов

Архитектура и дизайн программного обеспечения

Рабочая программа дисциплины

для студентов, обучающихся по направлению подготовки
09.03.04 Программная инженерия,
ОП «Технологии разработки программного обеспечения»

*Рекомендовано Ученым советом Владикавказского филиала
Финуниверситета*

(протокол от «15» апреля 2026 г. № ____)

*Одобрено на заседании кафедры «Математика и информатика»
(протокол от «10» апреля 2026 г. № 8)*

Владикавказ 2026

Содержание

1. Наименование дисциплины.....	3
2. Перечень планируемых результатов освоения образовательной программы (перечень компетенций) с указанием индикаторов их достижения и планируемых результатов обучения по дисциплине	3
3. Место дисциплины в структуре образовательной программы.....	6
4. Объем дисциплины (модуля) в зачетных единицах и в академических часах с выделением объема аудиторной (лекции, семинары) и самостоятельной работы обучающихся.....	7
5. Содержание дисциплины, структурированное по темам (разделам) дисциплины с указанием их объемов (в академических часах) и видов учебных занятий.....	7
5.1. Содержание дисциплины.....	7
5.2. Учебно-тематический план.....	8
5.3. Содержание семинаров, практических занятий.....	10
6. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине.....	12
6.1. Перечень вопросов, отводимых на самостоятельное освоение дисциплины, формы внеаудиторной самостоятельной работы.....	12
6.2. Перечень вопросов, заданий, тем для подготовки к текущему контролю	14
7. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине.....	16
8. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины	24
9. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины.....	25
10. Методические указания для обучающихся по освоению дисциплины.....	25
11. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень необходимого программного обеспечения и информационных справочных систем	31
12. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине.....	32

1. Наименование дисциплины

«Архитектура и дизайн программного обеспечения».

2. Перечень планируемых результатов освоения образовательной программы (перечень компетенций) с указанием индикаторов их достижения и планируемых результатов обучения по дисциплине

Код компетенции	Наименование компетенции	Индикаторы достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции
ПКП-1	Способность описывать, анализировать и проектировать интерфейс программных модулей с учетом требований к ним	1. Демонстрирует знания основных понятий интерфейсов программных модулей, понятие внешней и внутренней среды, читает и понимает готовую программную документацию в части описания интерфейсов.	Знать: основные понятия интерфейсов программных модулей, понятие внешней и внутренней среды Уметь: использовать на практике готовую программную документацию в части описания интерфейсов
		2. Понимает достоинства и недостатки различных архитектурных решений в области проектирования интерфейсов программных модулей, может критически анализировать существующие решения	Знать: достоинства и недостатки различных архитектурных решений в области проектирования интерфейсов программных модулей Уметь: критически анализировать существующие решения
		3. Описывает интерфейс программной системы в формализованном виде по определенным стандартам, демонстрирует знания общепринятых стандартов описания архитектуры программной системы	Знать: общепринятые стандарты описания архитектуры программной системы Уметь: описывать интерфейс программной системы в формализованном виде по определенным стандартам

		4.Проектирует интерфейс программного модуля с учетом требований к программной системе в целом и с учетом интеграции с другими программными модулями.	Знать: требования к программной системе в целом Уметь: проектировать интерфейс программного модуля
ПКП-2	Способность управлять изменениями в разрабатываемой программной системе	1.Понимает необходимость развития и изменения разрабатываемого программного продукта, имеет представление о процессах по внесению изменений в программный проект, владеет инструментальными средствами контроля версий.	Знать: необходимость развития и изменения разрабатываемого программного продукта Уметь: использовать на практике инструментальные средства контроля версий.
		2.Управляет качеством программного продукта в процессе внесения изменений, владеет методиками мониторинга качества, организывает процесс сопровождения программной системы.	Знать: методики мониторинга качества, процесс сопровождения программной системы Уметь: управлять качеством программного продукта в процессе внесения изменений
		3.Организовывает и автоматизирует процессы поддержки и сопровождения программных проектов, включая процессы миграции данных, периодические релизы, тестирование и непрерывное развертывание программных систем, демонстрирует знания соответствующих инструментальных средств.	Знать: соответствующие инструментальные средства Уметь: автоматизировать процессы поддержки и сопровождения программных проектов, включая процессы миграции данных, периодические релизы, тестирование и непрерывное развертывание программных систем

		4.Прогнозирует наиболее вероятное направление внесения изменений, выбирает те или иные архитектурные решения, облегчающие процесс внесения изменений в программную систему на этапе проектирования программной системы.	<p>Знать: процесс внесения изменений в программную систему на этапе проектирования программной системы</p> <p>Уметь: прогнозировать наиболее вероятное направление внесения изменений, выбирать те или иные архитектурные решения, облегчающие процесс внесения изменений в программную систему на этапе проектирования программной системы</p>
ПКП-4	Способность проектировать архитектуру и дизайн программной системы на основе требований к ней	1.Декомпозирует задачу, решаемую программной системой на более простые, элементарные подзадачи, реализует простые задачи с помощью алгоритмизации и в виде программного модуль на языке программирования.	<p>Знать: основные алгоритмы и структуры данных</p> <p>Уметь: использовать на практике простые структуры данных, оценивать сложность алгоритмов</p>
		2.Демонстрирует знания основ теории архитектуры и дизайна программного обеспечения, разделения задачи на уровни абстракции, подбирает те или иные архитектурные решения на основе анализа требований к программной системе в целом.	<p>Знать: основы теории архитектуры и дизайна программного обеспечения, разделение задачи на уровни абстракции</p> <p>Уметь: подбирать те или иные архитектурные решения на основе анализа требований к программной системе в целом</p>
		3.Проводит интеграцию программных модулей, описывает, анализирует и модифицирует интерфейс программных модулей, понимает принципы разделения ответственности, модульности и инверсии зависимостей применительно к созданию программного обеспечения.	<p>Знать: принципы разделения ответственности, модульности и инверсии зависимостей применительно к созданию программного обеспечения</p> <p>Уметь: проводить интеграцию программных модулей, описывать, анализировать и модифицировать интерфейс программных модулей</p>

		4.Проектирует пользовательское взаимодействие с программной системой, включая пользовательский интерфейс, сценарии использования, механизмы аутентификации и разделения ролей, в том числе, с учетом требований информационной безопасности.	<p>Знать: сценарии использования, механизмы аутентификации и разделения ролей, в том числе, с учетом требований информационной безопасности</p> <p>Уметь: проектировать пользовательское взаимодействие с программной системой, включая пользовательский интерфейс</p>
		5. Адаптирует архитектуру и интерфейс программной системы под известную внешнюю среду, включая интерфейсы внешних программных систем.	<p>Знать: интерфейсы внешних программных систем</p> <p>Уметь: адаптировать архитектуру и интерфейс программной системы под известную внешнюю среду, включая интерфейсы внешних программных систем</p>
		6.Учитывает нефункциональные требования к программной системе в ее архитектуре, в том числе требования к надежности, согласованности, скорости работы программной системы.	<p>Знать: архитектурные решения на основе нефункциональных требований к программной системе</p> <p>Уметь: собирать, формулировать, систематизировать и анализировать нефункциональные требования к программной системе</p>

3. Место дисциплины в структуре образовательной программы

Дисциплина «Архитектура и дизайн программного обеспечения» является дисциплиной профиля части, формируемой участниками образовательных отношений, учебного плана образовательной программы «Технологии разработки программного обеспечения» по направлению подготовки 09.03.04 Программная инженерия, профиль «Технологии разработки программного обеспечения».

4. Объем дисциплины (модуля) в зачетных единицах и в академических часах с выделением объема аудиторной (лекции, семинары) и самостоятельной работы обучающихся

Вид учебной работы по дисциплине	Всего (в з/е и часах)	Семестр 6 (в часах)
Общая трудоемкость дисциплины	5/180	180
<i>Контактная работа – Аудиторные занятия</i>	<i>68</i>	<i>68</i>
<i>Лекции</i>	<i>16</i>	<i>16</i>
<i>Семинары, практические занятия</i>	<i>52</i>	<i>52</i>
Самостоятельная работа	112	112
Вид текущего контроля	Проектная работа	Проектная работа
Вид промежуточной аттестации	Экзамен	Экзамен

5. Содержание дисциплины, структурированное по темам (разделам) дисциплины с указанием их объемов (в академических часах) и видов учебных занятий

5.1. Содержание дисциплины

Тема 1. Архитектуры программных систем

Архитектурные образцы, эталонные модели и эталонные варианты архитектур. Архитектурные структуры и представления. Архитектуры, основанные на уровнях абстракции. Архитектуры, основанные на портах. Архитектуры, основанные на потоках данных. Архитектуры независимых компонентов. Сервис ориентированные архитектуры (SOA).

Тема 2. Архитектурные представления программных систем

Архитектурный вид – структура многослойной программной системы. Архитектурный вид – размещение программной системы. Архитектурный вид – размещение программной системы, основанной на потоках данных. Архитектурный вид – распределение работ по группам разработчиков.

Тема 3. Модульность

Модули, модульно-интерфейсный подход, модельное программирование. Обоснование модульности. Внутренняя характеристика модуля – связанность (прочность). Сцепление модулей – внешняя характеристика модуля.

Тема 4. Сложность программной системы

Методы оценки сложности. Оценка сложности на основе связанности и сцепления модулей.

Тема 5. Представление архитектуры программных систем

Модульно-интерфейсный подход. Объектно-ориентированный подход. Компонентный подход.

Тема 6. Методы структурного проектирования

Методы восходящей разработки. Методы нисходящей разработки. Особенности структурного проектирования.

Тема 7. Формальное описание методики разработки модульной архитектуры программной системы

Проектирование «снизу вверх». Проектирование «снизу вверх». Проектирование архитектуры на основе объектно - ориентированной и компонентной методологии.

Тема 8. Рефакторинг программных систем

Потребность в архитектурном рефакторинге. Построение архитектуры программного средства по ее программному коду. Рефакторинг архитектуры многослойной иерархической программной системы. Возможные подходы к созданию программных средств. Представление созданной архитектуры программного средства. Анализ на соответствие послойной архитектуре (выделение слоев). Коррекция (трансформация) архитектуры в интересах ее рефакторинга. Рефакторинг архитектуры в интересах повышения производительности ИС.

5.2. Учебно – тематический план

№ п/ п	Наименование тем (разделов) дисциплины	Трудоемкость в часах					Формы текущего контроля успеваемости
		Всего	Контактная работа - Аудиторная работа			Самос- тоятельная работа	
			Общая, в т.ч.:	Лекции	Семинары, практические занятия		

1	Архитектуры программных систем	16	6	2	4	10	Опрос, собеседование по домашним заданиям самостоятельной работы, решение практико-ориентированных задач.
2	Архитектурные представления программных систем	18	8	2	6	10	Опрос, собеседование по домашним заданиям самостоятельной работы, решение практико-ориентированных задач.
3	Модульность	16	6	2	4	10	Опрос, собеседование по домашним заданиям самостоятельной работы, решение практико-ориентированных задач.
4	Сложность программной системы	18	8	2	6	10	Опрос, собеседование по домашним заданиям самостоятельной работы, решение практико-ориентированных задач.
5	Представление архитектуры программных систем	18	8	2	6	10	Опрос, собеседование по домашним заданиям самостоятельной работы, решение практико-ориентированных задач.

6	Методы структурного проектирования	18	8	2	6	10	Опрос, собеседование по домашним заданиям самостоятельной работы, решение практико-ориентированных задач.
7	Формальное описание методики разработки модульной архитектуры программной системы	20	10	2	8	10	Опрос, собеседование по домашним заданиям самостоятельной работы, решение практико-ориентированных задач.
8	Рефакторинг программных систем	56	14	2	12	42	Опрос, собеседование по домашним заданиям самостоятельной работы, решение практико-ориентированных задач.
	В целом по дисциплине	180	68	16	52	112	Согласно учебному плану: проектная работа
	Итого в %		38	24	76	62	

5.3. Содержание семинаров, практических занятий

Наименование тем (разделов) дисциплины	Перечень вопросов для обсуждения на семинарах, практических занятиях	Формы проведения занятий
Архитектуры программных систем	Методология решения задач проектирования по Г. Майерсу. Структуры “компонент и соединитель”	Интерактивная форма: опрос, собеседование по домашним заданиям самостоятельной работы, решение практико-ориентированных задач с

		последующим коллективным обсуждением их результатов
Архитектурные представления программных систем	Унифицированный язык моделирования UML. Использование базовых диаграмм UML при проектировании ПО: - основные понятия UML, - диаграммы прецедентов, - диаграммы классов, - диаграммы взаимодействий, - диаграммы последовательности действий, - диаграммы состояний, компонентные диаграммы.	Интерактивная форма: опрос, собеседование по домашним заданиям самостоятельной работы, решение практико-ориентированных задач с последующим коллективным обсуждением их результатов
Модульность	UML. Диаграммы пакетов	Интерактивная форма: опрос, собеседование по домашним заданиям самостоятельной работы, решение практико-ориентированных задач с последующим коллективным обсуждением их результатов
Сложность программной системы	Методы оценки	Интерактивная форма: опрос, собеседование по домашним заданиям самостоятельной работы, решение практико-ориентированных задач с последующим коллективным обсуждением их результатов
Представление архитектуры программных систем	UML. Диаграммы компонентов. Диаграммы развертывания	Интерактивная форма: опрос, собеседование по домашним заданиям самостоятельной работы, решение практико-ориентированных задач с последующим коллективным обсуждением их результатов
Методы структурного проектирования	SADT-модели. DFD-диаграммы	Интерактивная форма: опрос, собеседование по домашним заданиям самостоятельной работы, решение практико-ориентированных задач с последующим коллективным обсуждением их результатов
Формальное описание методики разработки модульной архитектуры программной системы	Проектирования многослойного программного продукта	Интерактивная форма: опрос, собеседование по домашним заданиям самостоятельной работы, решение практико-ориентированных задач с последующим коллективным обсуждением их результатов

Рефакторинг программных систем	Рефакторинг большого класса. Рефакторинг ленивого класса. Способы выявления проблемных участков кода. Методы рефакторинга: улучшения структуры методов в классах, организация данных	Занятия в интерактивной форме в виде дискуссий. Интерактивная форма: опрос, собеседование по домашним заданиям самостоятельной работы, решение практико-ориентированных задач с последующим коллективным обсуждением их результатов, защита проектной работы
--------------------------------	--	--

6. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине

6.1. Перечень вопросов, отводимых на самостоятельное освоение дисциплины, формы внеаудиторной самостоятельной работы

Наименование тем (разделов) дисциплины	Перечень вопросов, отводимых на самостоятельное освоение	Формы внеаудиторной самостоятельной работы
Архитектуры программных систем	Структуры «компонент и соединитель». Структуры распределения. Отношения между структурами.	Изучение материалов лекций и литературы, предложенной преподавателем, поиск и анализ информации, содержащейся в Интернет-ресурсах. Разбор вопросов, отводимых на самостоятельное освоение
Архитектурные представления программных систем	Архитектурное представление назначения заданий	Изучение материалов лекций и литературы, предложенной преподавателем, поиск и анализ информации, содержащейся в Интернет-ресурсах. Разбор вопросов, отводимых на самостоятельное освоение, выполнение домашних заданий самостоятельной работы
Модульность	Декомпозиционное представление модульной архитектуры и представление зависимостей	Изучение материалов лекций и литературы, предложенной преподавателем, поиск и анализ информации, содержащейся в Интернет-ресурсах. Разбор вопросов, отводимых на самостоятельное освоение, выполнение домашних заданий самостоятельной работы

Сложность программной системы	Минимизация сложности	Изучение материалов лекций и литературы, предложенной преподавателем, поиск и анализ информации, содержащейся в Интернет-ресурсах. Разбор вопросов, отводимых на самостоятельное освоение , выполнение домашних заданий самостоятельной работы
Представление архитектуры программных систем	Архитектурное представление назначения заданий	Изучение материалов лекций и литературы, предложенной преподавателем, поиск и анализ информации, содержащейся в Интернет-ресурсах. Разбор вопросов, отводимых на самостоятельное освоение , выполнение домашних заданий самостоятельной работы
Методы структурного проектирования	Методология решения задач проектирования по Майерсу. Уровни требований к программным системам	Изучение материалов лекций и литературы, предложенной преподавателем, поиск и анализ информации, содержащейся в Интернет-ресурсах. Разбор вопросов, отводимых на самостоятельное освоение , выполнение домашних заданий самостоятельной работы
Формальное описание методики разработки модульной архитектуры программной системы	Процесс анализа. Процесс синтеза	Изучение материалов лекций и литературы, предложенной преподавателем, поиск и анализ информации, содержащейся в Интернет-ресурсах. Разбор вопросов, отводимых на самостоятельное освоение , выполнение домашних заданий самостоятельной работы
Рефакторинг программных систем	Рефакторинг в СУБД. Инструментальные средства проведения рефакторинга	Изучение материалов лекций и литературы, предложенной преподавателем, поиск и анализ информации, содержащейся в Интернет-ресурсах. Разбор вопросов, отводимых на самостоятельное освоение, выполнение домашних заданий самостоятельной работы. Выполнение проектной работы.

6.2. Перечень вопросов, заданий, тем для подготовки к текущему контролю

Примерный перечень тем для подготовки к опросу, подготовки докладов, презентаций

Тема 1. Архитектуры программных систем. Архитектурные образцы, эталонные модели и эталонные варианты архитектур. Архитектурные структуры и представления. Архитектуры, основанные на уровнях абстракции. Архитектуры, основанные на портах. Архитектуры, основанные на потоках данных. Архитектуры независимых компонентов. Сервис ориентированные архитектуры (SOA).

Тема 2. Архитектурные представления программных систем. Архитектурный вид – структура многослойной программной системы. Архитектурный вид – размещение программной системы. Архитектурный вид – размещение программной системы, основанной на потоках данных. Архитектурный вид – распределение работ по группам разработчиков.

Тема 3. Модульность. Модули, модульно-интерфейсный подход, модельное программирование. Обоснование модульности. Внутренняя характеристика модуля – связанность (прочность). Сцепление модулей – внешняя характеристика модуля.

Тема 4. Сложность программной системы. Методы оценки сложности. Оценка сложности на основе связанности и сцепления модулей.

Тема 5. Представление архитектуры программных систем. Модульно-интерфейсный подход. Объектно-ориентированный подход. Компонентный подход.

Тема 6. Методы структурного проектирования. Методы восходящей разработки. Методы нисходящей разработки. Особенности структурного проектирования.

Тема 7. Формальное описание методики разработки модульной архитектуры программной системы. Проектирование «снизу вверх». Проектирование «снизу вверх». Проектирование архитектуры на основе объектно - ориентированной и компонентной методологии.

Тема 8. Рефакторинг программных систем. Потребность в архитектурном рефакторинге. Построение архитектуры программного средства по ее программному коду. Рефакторинг архитектуры многослойной иерархической программной системы.

Возможные подходы к созданию программных средств. Представление созданной архитектуры программного средства. Анализ на соответствие послойной архитектуре (выделение слоев). Коррекция (трансформация) архитектуры в интересах ее рефакторинга. Рефакторинг архитектуры в интересах повышения производительности ИС.

Примеры практико-ориентированных задач

1. Создать диаграммы классов и последовательностей для системы управления базами данных.
2. Разработать архитектуру для мобильного приложения, учитывая ограничения ресурсов и требования к производительности.
3. Спроектировать архитектуру для системы обработки больших данных, включая выбор подходящих технологий и алгоритмов.
4. Создать диаграмму компонентов и развертывания для системы электронной коммерции.
5. Создать диаграмму компонентов и развертывания для системы онлайн-бронирования билетов в театр.

Примерные задания для проектной работы

1. Спроектировать программы управления ИТ-проектами: задача подбора персонала.
2. Спроектировать программы управления ИТ-проектами: задача составления расписания выполнения проекта.
3. Спроектировать программы управления ИТ-проектами: задача прогнозирования себестоимости ИТ-проекта.
4. Спроектировать программное обеспечение учебного тренажера по выбранной теме.
5. Спроектировать программный комплекс для обработки данных на выбранную тематику.
6. Спроектировать программный комплекс поддержки принятия управленческих решений (разработка системы учета договоров и расчетов с

субподрядчиками)

9. Создать проект программного учёта материально-технических средств.

10. Создать проект клиент- серверного приложения для решения задачи по выбранной тематики.

Для выбранного варианта предметной области построить следующие диаграммы UML:

- диаграммы прецедентов,
- диаграммы классов,
- диаграммы взаимодействий,
- диаграммы последовательности действий,
- диаграммы состояний,
- диаграммы компонентов,
- диаграммы развертывания.

Критерии балльной оценки различных форм текущего контроля успеваемости содержатся в соответствующих методических рекомендациях кафедры «Математика и информатика».

7. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине

Перечень компетенций с указанием индикаторов их достижения в процессе освоения образовательной программы содержится в разделе 2. «Перечень планируемых результатов освоения образовательной программы (перечень компетенций) с указанием индикаторов их достижения и планируемых результатов обучения по дисциплине».

Типовые контрольные задания или иные материалы, необходимые для оценки индикаторов достижения компетенций, знаний и умений

Примерные вопросы для подготовки к экзамену

1. Архитектурные образцы, эталонные модели и эталонные варианты архитектур.

2. Архитектурные структуры и представления.
3. Архитектуры, основанные на уровнях абстракции.
4. Архитектуры, основанные на портах.
5. Архитектуры, основанные на потоках данных.
6. Архитектуры независимых компонентов.
7. Сервис ориентированные архитектуры (SOA).
8. Архитектурный вид – структура многослойной программной системы.
9. Архитектурный вид – размещение программной системы.
10. Архитектурный вид – размещение программной системы, основанной на потоках данных.
11. Архитектурный вид – распределение работ по группам разработчиков.
12. Модули, модульно-интерфейсный подход, модельное программирование.
13. Обоснование модульности.
14. Внутренняя характеристика модуля – связанность (прочность).
15. Сцепление модулей – внешняя характеристика модуля.
16. Методы оценки сложности.
17. Оценка сложности на основе связанности и сцепления модулей.
18. Модульно-интерфейсный подход.
19. Объектно-ориентированный подход.
20. Компонентный подход.
21. Методы восходящей разработки.
22. Методы нисходящей разработки.
23. Особенности структурного проектирования.
24. Проектирование «снизу вверх».
25. Проектирование «снизу вверх».
26. Проектирование архитектуры на основе объектно - ориентированной и компонентной методологии.
27. Потребность в архитектурном рефакторинге.
28. Построение архитектуры программного средства по ее программному

коду.

29. Рефакторинг архитектуры многослойной иерархической программной системы.
30. Возможные подходы к созданию программных средств.
31. Представление созданной архитектуры программного средства.
32. Анализ на соответствие послойной архитектуре (выделение слоев).
33. Коррекция (трансформация) архитектуры в интересах ее рефакторинга.
34. Рефакторинг архитектуры в интересах повышения производительности ИС.

Примерные задания для подготовки к экзамену

1. Создать диаграммы классов и последовательностей для системы управления базами данных.
2. Разработать архитектуру для мобильного приложения, учитывая ограничения ресурсов и требования к производительности.
3. Спроектировать архитектуру для системы обработки больших данных, включая выбор подходящих технологий и алгоритмов.
4. Создать диаграмму компонентов и развертывания для системы электронной коммерции.
5. Создать диаграмму компонентов и развертывания для системы онлайн-бронирования билетов в театр.

Пример экзаменационного билета

**Федеральное государственное образовательное бюджетное
учреждение высшего образования
«Финансовый университет при Правительстве Российской Федерации»
(Финансовый университет)**

Кафедра: **Математика и информатика**

Дисциплина: **Архитектура и дизайн программного обеспечения**

Филиал: **Владикавказский**; Форма обучения: **Очная**

Семестр: **6** Направление: **09.03.04 Программная инженерия**

Профиль: **Технологии разработки программного обеспечения**

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 1

1. Сервис ориентированные архитектуры (SOA) (20 баллов)
2. Проектирование архитектуры на основе объектно - ориентированной и компонентной методологии (20 баллов)
3. Создать диаграммы классов и последовательностей для системы управления базами данных (20 баллов).

Подготовил: _____

На основе перечня теоретических вопросов и практико-ориентированных заданий, утвержденного на заседании кафедры «Математика и информатика» протокол № ____ от _____.2026 г.

Утверждаю:

Заведующий кафедрой _____

Дата _____.2026г.

Примеры оценочных средств для проверки индикаторов достижения компетенций, формируемых дисциплиной

Наименование компетенции	Наименование индикаторов достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции	Типовые контрольные задания
Способность описывать, анализировать и проектировать интерфейс программных модулей с учетом требований к ним (ПКП-1)	1. Демонстрирует знания основных понятий интерфейсов программных модулей, понятие внешней и внутренней среды, читает и понимает готовую программную документацию в части описания интерфейсов.	Уметь использовать на практике готовую программную документацию в части описания интерфейсов Знать основные понятия интерфейсов программных модулей, понятие внешней и внутренней среды	Спроектировать компонент подключения по API к производителю датчик температуры, для обновления прошивок и передачи диагностических данных Спроектировать адаптеры интерфейсов в виде компонент (подключение к БД, шлюзы к интерфейсу, шлюзы к третьим источникам)

	2.Понимает достоинства и недостатки различных архитектурных решений в области проектирования интерфейсов программных модулей, может критически анализировать существующие решения	Уметь критически анализировать существующие решения Знать достоинства и недостатки различных архитектурных решений в области проектирования интерфейсов программных модулей	Спроектировать облачный сервис домашнего робота пылесоса для уборки помещений Спроектировать сценарии (Use case) (подключение, выбор помещения, программы уборки, настройка расписания, просмотр статистики..), в виде Актор – Прецедент
	3.Описывает интерфейс программной системы в формализованном виде по определенным стандартам, демонстрирует знания общепринятых стандартов описания архитектуры программной системы	Уметь описывать интерфейс программной системы в формализованном виде по определенным стандартам Знать общепринятые стандарты описания архитектуры программной системы	Построить диаграмму последовательностей для предметной области «Продажа ЖД билетов» Построить диаграмму последовательностей для предметной области «Интернет-магазин»
	4.Проектирует интерфейс программного модуля с учетом требований к программной системе в целом и с учетом интеграции с другими программными модулями.	Уметь проектировать интерфейс программного модуля Знать требования к программной системе в целом	Построить диаграмму состояний для предметной области «Продажа ЖД билетов» Построить диаграмму состояний для предметной области «Интернет-магазин»
Способность управлять изменениями в разрабатываемой программной системе (ПКП-2)	1.Понимает необходимость развития и изменения разрабатываемого программного продукта, имеет представление о процессах по внесению изменений в программный проект, владеет инструментальными средствами контроля версий.	Уметь использовать на практике инструментальные средства контроля версий. Знать необходимость развития и изменения разрабатываемого программного продукта	Произвести рефакторинг кода на С, реализующего алгоритм адаптивного арифметического кодирования. Произвести рефакторинг кода на С, реализующего игру виселица.

	<p>2. Управляет качеством программного продукта в процессе внесения изменений, владеет методиками мониторинга качества, организывает процесс сопровождения программной системы.</p>	<p>Уметь управлять качеством программного продукта в процессе внесения изменений</p> <p>Знать методики мониторинга качества, процесс сопровождения программной системы</p>	<p>Произвести рефакторинг уже существующей реализации алгоритма подбора близкой для числа дроби</p> <p>Произвести рефакторинг уже существующей реализации алгоритма игры, которая предлагает игроку различные примеры на арифметические действия.</p>
	<p>3. Организует и автоматизирует процессы поддержки и сопровождения программных проектов, включая процессы миграции данных, периодические релизы, тестирование и непрерывное развертывание программных систем, демонстрирует знания соответствующих инструментальных средств.</p>	<p>Уметь автоматизировать процессы поддержки и сопровождения программных проектов, включая процессы миграции данных, периодические релизы, тестирование и непрерывное развертывание программных систем</p> <p>Знать соответствующие инструментальные средства</p>	<p>Произвести рефакторинг уже существующей реализации алгоритма игры «Крестики-нолики»</p> <p>Произвести рефакторинг кода на C, реализующего карточную игру «Пасьянс».</p>
	<p>4. Прогнозирует наиболее вероятное направление внесения изменений, выбирает те или иные архитектурные решения, облегчающие процесс внесения изменений в программную систему на этапе проектирования программной системы.</p>	<p>Уметь прогнозировать наиболее вероятное направление внесения изменений, выбирать те или иные архитектурные решения, облегчающие процесс внесения изменений в программную систему на этапе проектирования программной системы</p> <p>Знать процесс внесения изменений в программную систему на этапе</p>	<p>Проектирование архитектуры для системы управления контентом, учитывая требования к гибкости и масштабируемости.</p> <p>Создание диаграмм состояний и активностей для системы</p>

		проектирования программной системы	автоматизации процессов бизнеса
Способность проектировать архитектуру и дизайн программной системы на основе требований к ней (ПКП-4)	1.Декомпозирует задачу, решаемую программной системой на более простые, элементарные подзадачи, реализует простые задачи с помощью алгоритмизации и в виде программного модуль на языке программирования.	Уметь использовать на практике простые структуры данных, оценивать сложность алгоритмов Знать основные алгоритмы и структуры данных	Для заданной предметной области «Обнаружение спама» построить диаграмму пакетов Для заданной предметной области «Обнаружение спама» построить диаграмму компонентов
	2.Демонстрирует знания основ теории архитектуры и дизайна программного обеспечения, разделения задачи на уровни абстракции, подбирает те или иные архитектурные решения на основе анализа требований к программной системе в целом.	Уметь подбирать те или иные архитектурные решения на основе анализа требований к программной системе в целом Знать основы теории архитектуры и дизайна программного обеспечения, разделение задачи на уровни абстракции	Постройте архитектуру системы оценки кредитоспособности (кредитных рисков) лица Постройте модель системы оценки кредитоспособности (кредитных рисков) лица
	3.Проводит интеграцию программных модулей, описывает, анализирует и модифицирует интерфейс программных модулей, понимает принципы разделения ответственности, модульности и инверсии зависимостей применительно к созданию программного обеспечения.	Уметь проводить интеграцию программных модулей, описывать, анализировать и модифицировать интерфейс программных модулей Знать принципы разделения ответственности, модульности и инверсии зависимостей применительно к созданию программного обеспечения	Построить диаграмму классов для предметной области «Продажа ЖД билетов» 1. Диаграмма прецедентов системы онлайн покупки авиабилетов. 2. Диаграмма классов системы онлайн покупки авиабилетов.

	<p>4.Проектирует пользовательское взаимодействие с программной системой, включая пользовательский интерфейс, сценарии использования, механизмы аутентификации и разделения ролей, в том числе, с учетом требований информационной безопасности.</p>	<p>Уметь проектировать пользовательское взаимодействие с программной системой, включая пользовательский интерфейс</p> <p>Знать сценарии использования, механизмы аутентификации и разделения ролей, в том числе, с учетом требований информационной безопасности</p>	<p>Построить диаграмму классов для предметной области «Регистратура поликлиники»</p> <p>1.Диаграмма прецедентов системы онлайн записи к врачу.</p> <p>2.Диаграмма классов системы онлайн записи к врачу</p>
	<p>5.Адаптирует архитектуру и интерфейс программной системы под известную внешнюю среду, включая интерфейсы внешних программных систем.</p>	<p>Уметь адаптировать архитектуру и интерфейс программной системы под известную внешнюю среду, включая интерфейсы внешних программных систем</p> <p>Знать интерфейсы внешних программных систем</p>	<p>Построить диаграмму классов для предметной области «Учет лекарств в аптеке»</p> <p>1.Диаграмма прецедентов системы учета лекарств в аптеке.</p> <p>2. Диаграмма классов системы учета лекарств в аптеке</p>
	<p>6.Учитывает нефункциональные требования к программной системе в ее архитектуре, в том числе требования к надежности, согласованности, скорости работы программной системы.</p>	<p>Уметь собирать, формулировать, систематизировать и анализировать нефункциональные требования к программной системе</p> <p>Знать архитектурные решения на основе нефункциональных требований к программной системе</p>	<p>Построить диаграмму классов для предметной области «Прокат велосипедов»</p> <p>1.Диаграмма прецедентов системы проката велосипедов.</p> <p>2. Диаграмма классов системы проката велосипедов</p>

8. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины

Основная литература:

1. Гагарина, Л. Г. Технология разработки программного обеспечения: учебное пособие / Л.Г. Гагарина, Е.В. Кокорева, Б.Д. Сидорова-Виснадул; под ред. Л.Г. Гагариной. — Москва: ИНФРА-М, 2026. — 400 с. — <https://znanium.ru/catalog/product/2212387> – Режим доступа: Электронно-библиотечная система Znanium.com – Текст: электронный.

2. Назаров, С. В. Архитектура и проектирование программных систем: монография / С.В. Назаров. — 2-е изд., перераб. и доп. — Москва: ИНФРА-М, 2025. — 374 с. — (Научная мысль). — ISBN 978-5-16-011753-9. - URL: <https://znanium.ru/catalog/product/2166195> – Режим доступа: Электронно-библиотечная система Znanium.com – Текст: электронный.

Дополнительная литература:

3. Лаврищева, Е. М. Программная инженерия и технологии программирования сложных систем: учебник для вузов / Е. М. Лаврищева. — 2-е изд., испр. и доп. — Москва: Издательство Юрайт, 2025. — 432 с. — (Высшее образование). — ISBN 978-5-534-07604-2. — URL: <https://urait.ru/bcode/561885> – Режим доступа: Электронно-библиотечная система Юрайт. – Текст: электронный.

4. Новожилов, О. П. Архитектура ЭВМ и вычислительных систем: учебник для вузов / О. П. Новожилов. — 2-е изд., испр. и доп. — Москва: Издательство Юрайт, 2026. — 505 с. — (Высшее образование). — ISBN 978-5-534-20365-3. — URL: <https://urait.ru/bcode/589607> – Режим доступа: Электронно-библиотечная система Юрайт. – Текст: электронный.

5. Галиаскаров, Э. Г. Анализ и проектирование систем с использованием UML: учебник для вузов / Э. Г. Галиаскаров, А. С. Воробьев. — Москва: Издательство Юрайт, 2026. — 125 с. — (Высшее образование). — ISBN 978-5-534-14903-6. — URL: <https://urait.ru/bcode/588976> – Режим доступа: Электронно-библиотечная система Юрайт. – Текст: электронный.

9. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

1. Информационно-образовательный портал Финансового университета при Правительстве Российской Федерации <http://portal.ufrf.ru/>
2. Сайт департамента анализа данных и машинного обучения.
3. Электронная библиотека Финансового университета (ЭБ) <http://elib.fa.ru/>
4. Электронно-библиотечная система BOOK.RU <http://www.book.ru>
5. Электронно-библиотечная система «Университетская библиотека ОНЛАЙН» <http://biblioclub.ru/>
6. Электронно-библиотечная система Znanium <http://www.znanium.com>
7. Электронно-библиотечная система издательства «ЮРАЙТ» <https://urait.ru/>
8. Электронно-библиотечная система издательства Проспект <http://ebs.prospekt.org/books>
9. Электронно-библиотечная система издательства «Лань» <https://e.lanbook.com/>
10. Электронная библиотека Издательского дома «Гребенников» <https://grebennikon.ru/>

10. Методические указания для обучающихся по освоению дисциплины

Методика освоения дисциплины предусматривает подготовку обучающихся к лекциям, семинарам и практическим занятиям, выполнение студентами самостоятельной внеаудиторной работы, в том числе – контрольной работы.

Рекомендации по подготовке к лекционным занятиям.

Для наиболее полного освоения дисциплины студентам необходимо:

- перед каждой лекцией просматривать рабочую программу дисциплины, ее основные вопросы и рекомендуемую литературу. Это позволит сэкономить время на

записывание основных вопросов темы;

- перед очередной лекцией просматривать материалы предыдущих, чтобы освоение материала не оставляло пробелов.

Рекомендации по подготовке к семинарам, практическим занятиям.

Студентам следует:

- проработать теоретический материал к занятию по рекомендованным литературным источникам и лекциям;
- использовать при подготовке к занятию нормативно-правовые документы, научные публикации, информационный материал, рекомендуемый преподавателем;
- перед занятиями задать вопросы по невыясненным в ходе самостоятельной подготовки темам или отдельным положениям темы;
- в ходе занятия давать четкие и исчерпывающие ответы на вопросы;
- на занятии демонстрировать понимание обсуждаемых тем и вопросов.

Студентам, пропустившим занятия по различным причинам, необходимо перед очередным занятием отработать пропущенный материал, подготовив его самостоятельно.

Методические рекомендации по выполнению различных форм самостоятельной работы

Студентам при организации самостоятельной работы следует руководствоваться Приказом Финансового университета № 1040/о от 11.05.2021г. «Об утверждении методических рекомендаций по планированию и организации внеаудиторной самостоятельной работы студентов по образовательным программам бакалавриата и магистратуры в Финансовом университете».

Самостоятельная работа содержит в себе различные виды и формы работ. Самостоятельная работа студентов включает в себя выполнение различного рода заданий, которые ориентированы на более глубокое усвоение материала изучаемой дисциплины. По теме учебной дисциплины студентам предлагается перечень заданий для самостоятельной работы.

В ходе изучения дисциплины предусмотрены следующие формы самостоятельной работы:

- подготовка к опросу;
 - выполнение заданий самостоятельной работы,
 - решение практико-ориентированных задач;
 - выполнение контрольной работы (эссе, домашнего творческого задания, проектной работы);
 - выполнение курсовой работы;
- подготовка к зачету и экзамену.

К выполнению заданий для самостоятельной работы предъявляются следующие требования: задания должны выполняться самостоятельно и представляться в установленный срок, а также должны соответствовать установленным требованиям по оформлению.

Студентам следует:

- руководствоваться графиком самостоятельной работы, определенным РПД;
- выполнять все плановые задания, выдаваемые преподавателем для самостоятельного выполнения, разбирать на занятиях и консультациях неясные вопросы;
- прорабатывать соответствующие теоретические и практические разделы дисциплины, фиксируя неясные фрагменты для их обсуждения на консультации.

Методические рекомендации для обучающихся по выполнению проектной работы

Проектная работа является обязательной формой внеаудиторной самостоятельной работы студентов по дисциплине.

Целью проектной работы является развитие у студентов способности прогнозировать, проектировать, моделировать, формирование учебно-исследовательских навыков, закрепление умений самостоятельно работать с различными источниками информации; проверка сформированности компетенций.

Проектная работа может выполняться как индивидуально, так и в составе группы. Количество групп и их численный состав определяет преподаватель, ведущий

семинарские занятия.

Заказчиками выполнения проекта могут являться представители работодателей. В этом случае проектная работа выполняется исходя из потребностей заказчика.

Выполнение проекта предполагает:

- диагностику ситуации (проблематизация, целеполагание, конкретизация цели, форматирование проекта);
- проектирование (уточнение цели, функций, задач и плана работы; теоретическое моделирование методов и средств решения задач; детальная проработка этапов решения конкретных задач; пошаговое выполнение запланированных проектных действий; систематизация и обобщение полученных результатов, конструирование предполагаемого результата, пошаговое выполнение проектных действий);
- рефлексия (выяснение соответствия полученного результата замыслу; определение качества полученного продукта; перспективы его развития и использования);
- фиксация результатов в виде исполненного проекта.

Проектная работа состоит из нескольких частей. Состав проектной работы и очередность размещения отдельных частей:

- титульный лист;
- основная часть;
- список использованных источников;
- приложения (при наличии).

Титульный лист является первой страницей проектной работы и заполняется по определенным правилам.

Основная часть выполняется согласно заданию преподавателя или исходя из потребностей заказчика.

В список использованных источников включаются названия законодательных актов, нормативных документов, книг, статей, учебных пособий и т. п., которые, так или иначе, использовались студентом при выполнении работы.

Иллюстративный материал (схемы, диаграммы, рисунки, таблицы и др.) встраивается в текст работы или выносится в Приложения. В Приложения выносятся

вспомогательные материалы, которые не содержат основную информацию, либо материалы, которые сложно разместить по тексту работы (большие схемы, таблицы, графические материалы, расчетные справочные данные, образцы первичных документов и т.п.). Непременным условием включения данных материалов в приложение является ссылка на них в тексте работы.

Требования к оформлению проектной работы.

Проектная работа выполняется на компьютере на одной стороне белой бумаги формата А4 (210х297 мм). Размер шрифта -13 или 14, междустрочный интервал – одинарный или полуторный.

Размеры полей: левое - 30 мм, правое - 15 мм, верхнее - 20 мм, нижнее-20 мм. Отступ первой строки абзаца - 1,25. Нумерация страниц – внизу в центре.

При написании допускаются только общепринятые сокращения (например, тыс. руб.).

Общий объем проектной работы составляет не более 10 страниц, не включая таблицы, графики и т.п. (при наличии), а также приложения (при наличии).

В тексте обязательны ссылки на литературные источники, лучше всего постраничные.

Законченная проектная работа, содержащая все требуемые элементы оформления, вставленная в папку (или файл) и скрепленная по левому краю, сдается на кафедру или непосредственно **руководителю проектной работы** – преподавателю; ведущему семинарские (практические) занятия по дисциплине. Он осуществляет проверку проектной работы, а также оказывает помощь при подготовке к ее защите.

Проектная работа защищается в назначенные сроки. Защита проектной работы проводится до начала сессии (в крайнем случае, до начала экзамена по соответствующему предмету). При защите студент кратко излагает основные положения работы, последовательность ее выполнения, свои предложения.

При защите проектной работы студент должен свободно ориентироваться в изложенном материале работы; ответить на все замечания преподавателя; уметь отвечать на вопросы преподавателя по проектной работе.

Оценка проектных работ студентов проводится в процессе текущего контроля успеваемости.

Критерии оценки проектной работы

Оценка «отлично» (5-6 баллов) выставляется студенту, если проектная работа отличается творческим (креативным) подходом, собственным оригинальным отношением автора к идее проекта; содержит полную диагностику ситуации, а также теоретическое моделирование методов и детальную проработку этапов решения конкретных задач; в работе сделаны необходимые выводы, намечены перспективы использования проекта, спланированы действия по его продвижению; работа отличается грамотным оформлением в точном соответствии с установленными правилами, с соблюдением логической последовательности изложения материала; студент в работе выдвигает новые идеи и трактовки, демонстрирует способность анализировать материал; на дополнительные вопросы при защите проектной работы даны полные ответы.

Оценка «хорошо» (3-4 балла) выставляется студенту, если проектная работа содержит достаточно полную диагностику ситуации, а также теоретическое моделирование методов и этапов решения конкретных задач; в работе сделаны выводы, намечены перспективы использования проекта; работа оформлена правильно с учетом 1-2 мелких погрешностей или 2-3 недочетов, исправленных самостоятельно или по требованию преподавателя; в работе соблюдена логическая последовательность изложения материала; студент в работе демонстрирует творческие способности и хорошую способность анализировать материал. На дополнительные вопросы при защите проектной работы даны не совсем полные ответы.

Оценка «удовлетворительно» (1-2 балла) выставляется студенту, если проектная работа содержит отдельные элементы моделирования методов и этапов решения конкретных задач; в работе сделаны выводы, намечены перспективы использования проекта; работа выполнена и оформлена правильно, но в ней допущены 1-2 погрешности

или одна грубая ошибка; в работе соблюдена логическая последовательность изложения материала; студент в работе демонстрирует удовлетворительную способность анализировать материал; допущены ошибки при ответе на дополнительные вопросы при защите проектной работы.

Оценки «неудовлетворительно» (0 баллов) заслуживает студент, если в работе отсутствуют элементы моделирования; студент в работе не проявил способность анализировать, прогнозировать и проектировать; в работе отсутствует логическая последовательность изложения материала, допущены грубые ошибки, которые обучающийся не может исправить даже по требованию преподавателя или работа не выполнена полностью.

При оценивании проектной работы на «неудовлетворительно» работа должна быть переделана (исправлена) в соответствии с полученными замечаниями, сдана на проверку заново и защищена не позднее срока окончания ее приёма и защиты.

Оценка результатов текущего контроля успеваемости и промежуточной аттестации обучающихся осуществляется в соответствии с Балльно-рейтинговой системой Финансового университета (Приказ Финансового университета № 2187/о от 01.10.2024 г. «Об утверждении Положения о проведении текущего контроля успеваемости и промежуточной аттестации студентов, обучающихся по образовательным программам высшего образования в Финансовом университете»).

11. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень необходимого программного обеспечения и информационных справочных систем

11.1. Комплект лицензионного программного обеспечения:

- 1) Антивирусная защита Kaspersky Security для виртуальных и облачных сред;
- 2) Windows, Microsoft Office или Astra Linux, Libre Office.

11.2. Современные профессиональные базы данных и информационные

справочные системы:

- Информационно-правовая система «Гарант»: <https://www.garant.ru>
- Информационно-правовая система «Консультант Плюс»
- Электронная энциклопедия: <http://ru.wikipedia.org/wiki/Wiki>
- Система комплексного раскрытия информации «СКРИН» - <http://www.skrin.ru/>

11.3. Сертифицированные программные и аппаратные средства защиты информации

Не используются

11.4. Python 3.

11.5. StarUML.

12. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине

Учебная аудитория для проведения учебных занятий, предусмотренных программой бакалавриата, оснащенная оборудованием и техническими средствами обучения

Аудитория № 36

Специализированная мебель:

Стол (учительский) – 1 шт.

Стол компьютерный – 1 шт.

Стол (студенческий) двухместный – 13 шт.

Стулья – 27 шт.

Доска меловая – 1 шт.

Технические средства обучения:

Компьютер в сборе – 1 шт.

Экран настенный – 1 шт.

Подключение к сети «Интернет» и обеспечение доступа в электронную

информационно-образовательную среду Финансового университета

Учебная аудитория для проведения учебных занятий, предусмотренных программой бакалавриата, оснащенная оборудованием и техническими средствами обучения

Аудитория № 32

Специализированная мебель:

Стол компьютерный – 20 шт.

Стол (двухместный) – 7 шт.

Стул – 34 шт.

Шкаф – 1 шт.

Технические средства обучения:

Компьютер в сборе – 20 шт.

Мультимедиа-проектор – 1 шт.

Экран настенный – 1 шт.

Подключение к сети «Интернет» и обеспечение доступа в электронную информационно-образовательную среду Финансового университета

Помещение для самостоятельной работы обучающихся:

Кабинет № 55. Читальный зал:

Специализированная мебель:

Стол – 20 шт.

Стул – 40 шт.

Шкаф для книг – 4 шт.

Стеллаж книжный – 13 шт.

Стеллаж выставочный – 4 шт.

Технические средства обучения:

Компьютер в сборе – 6 шт.

Телевизор – 1 шт.

Помещение для самостоятельной работы обучающихся оснащено компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду Финансового университета