

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«КАМЧАТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»  
(ФГБОУ ВО «КамчатГТУ»)

Факультет информационных технологий

Кафедра «Системы управления»

УТВЕРЖДАЮ

Декан факультета ИТ

 /И.А. Рычка/

«28» января 2026 г.

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ**

**«Проектирование и архитектура программных систем»**

направление подготовки  
09.03.04 Программная инженерия  
(уровень бакалавриата)

направленность (профиль)  
«Разработка программно-информационных систем»

Петропавловск-Камчатский,  
2026

Рабочая программа дисциплины составлена на основании ФГОС ВО по направлению подготовки 09.03.04 Программная инженерия, профиль «Разработка программно-информационных систем», учебного плана ФГБОУ ВО «КамчатГТУ».

Составитель рабочей программы


Доцент кафедры  
«Системы управления»

  
\_\_\_\_\_ (подпись)

С.В. Чебанюк  
(Ф.И.О.)

Рабочая программа рассмотрена на заседании кафедры «Системы управления».  
«20» декабря 2025 г., протокол № 5.

«20» декабря 2025 г.

  
\_\_\_\_\_

Заведующий кафедрой  
«Системы управления»  
А.А. Марченко

## 1. ЦЕЛИ И ЗАДАЧИ УЧЕБНОЙ ДИСЦИПЛИНЫ

Целью дисциплины является изучение современных инженерных принципов (методов) проектирования и анализа архитектуры программных систем; формирование у студентов понимания необходимости архитектурного проектирования как этапа жизненного цикла программного обеспечения.

Задачами дисциплины является рассмотрение архитектурных шаблонов в различных предметных областях, приобретение навыков описания архитектуры, понимание преимуществ и ограничений, накладываемых на систему при выборе той или иной архитектуры.

В результате изучения дисциплины студент должен:

Знать:

- Архитектурные паттерны в различных предметных областях. Понятие архитектуры предприятия.

- Этапы развития архитектуры программных систем. История развития языков описания архитектуры, тенденции развития концепции архитектуры программных систем.

- Стандарты описания архитектуры программных систем. Место архитектуры в контексте жизненного цикла программного обеспечения.

Уметь:

- Использовать архитектуру программной системы в процессе разработки.

- Использовать преимущества и недостатки конкретных архитектурных решений.

- Использовать шаблоны проектирования, связанные со специфическими предметными областями.

- Описывать архитектуру, использовать базовые архитектурные стили и модели, документировать архитектуру программных систем.

Владеть:

- Паттернами проектирования, соответствующими основными компонентами корпоративной архитектуры.

- Языками описания архитектуры, методами проектирования архитектуры программного обеспечения.

## 2. ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Процесс изучения дисциплины направлен на формирование общепрофессиональной компетенции:

– Способен участвовать в разработке стандартов, норм и правил, а также технической документации, связанной с профессиональной деятельностью (ОПК-4).

Планируемые результаты освоения практики, соотнесенные с планируемыми результатами освоения образовательной программы, представлены в таблице.

Таблица – Планируемые результаты обучения по дисциплине, соотнесенные с установленными в программе бакалавриата индикаторами достижения компетенций

Код компетенции	Наименование компетенции	Код и наименование индикатора достижения	Планируемый результат обучения по дисциплине	Код показателя освоения
ОПК-4	Способен участвовать в разработке стандартов, норм и правил, а	ИД-2 <sub>ОПК-4</sub> применять стандарты оформления	<b>Знать:</b> основные стандарты оформления технической документации на различных стадиях жизненного цикла.	3(ОПК-4)1

Код компетенции	Наименование компетенции	Код и наименование индикатора достижения	Планируемый результат обучения по дисциплине	Код показателя освоения
	также технической документации, связанной с профессиональной деятельностью	технической документации на различных стадиях жизненного цикла информационной системы	<b>Уметь:</b> конструировать программное обеспечение, разрабатывать основные программные документы	У(ОПК-4)1
			<b>Владеть:</b> методами и средствами разработки и оформления технической документации.	В(ОПК-4)1

### 3. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Дисциплина «Проектирование и архитектура программных систем» относится к обязательной части структуры образовательной программы.

Дисциплина «Проектирование и архитектура программных систем» базируется на знаниях и умениях, полученных студентами в процессе изучения дисциплины «Информатика и программирование», «Теоретическая информатика», «Введение в программную инженерию», «Разработка и анализ требований», «Конструирование программного обеспечения», «Тестирование программного обеспечения». Знания, приобретенные при освоении дисциплины «Проектирование и архитектура программных систем» будут использованы при изучении дисциплины «Проектирование человеко-машинного интерфейса», при выполнении курсового проекта и в дипломном проектировании при подготовке выпускной квалификационной работы.

### 4. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

#### 4.1 Тематический план дисциплины

Наименование разделов и тем	Всего часов/ЗЕ	Аудиторные занятия	Контактная работа по видам учебных занятий			Самостоятельная работа	Формы текущего контроля	Итоговый контроль знаний по дисциплине
			Лекции	Практические занятия	Лабораторные работы			
<b>Очная форма обучения</b>								
<u>Лекция 1.</u> Введение. Проблемы создания сложных программных систем.	10	2	2	-	-	8	Опрос	
<u>Лекция 2.</u> Архитектуры программных систем.	30	15	3	-	12	15	Опрос, защита лаб-й работы	

Лекция 3. Жизненный цикл программных систем (ПС).	31	16	4	-	12	15	Опрос, защита лаб-й работы	
Лекция 4. Проектирование программных систем. Постановка требований к ПС.	31	16	4	-	12	15	Опрос, защита лаб-й работы	
Лекция 5. Проектирование программных систем. Анализ требований и разработка внешних спецификаций.	16	7	2	-	5	9	Опрос, защита лаб-й работы	
Лекция 6. Проектирование архитектуры программных систем.	26	12	2	-	10	14	Опрос, защита лаб-й работы	
Экзамен				-			36	
<b>Итого</b>	<b>180/5</b>	<b>68</b>	<b>17</b>	<b>-</b>	<b>51</b>	<b>76</b>	<b>36</b>	
Заочная форма обучения								
Лекция 1. Введение. Проблемы создания сложных программных систем.	24	2	1		1	26	Опрос, защита лаб-й работы	
Лекция 2. Архитектуры программных систем.	18	2	1		1	20	Опрос, защита лаб-й работы	
Лекция 3. Жизненный цикл программных систем (ПС).	18	2	1		1	20	Опрос, защита лаб-й работы	
Лекция 4. Проектирование программных систем. Постановка требований к ПС.	26	2	1		1	28	Опрос, защита лаб-й работы	
Лекция 5. Проектирование программных систем. Анализ требований и разработка внешних спецификаций.	27	2			2	29	Опрос, защита лаб-й работы	
Лекция 6. Проектирование архитектуры программных систем.	27	2			2	32	Опрос, защита лаб-й работы	
Экзамен	4							9
<b>Итого</b>	<b>180/5</b>	<b>12</b>	<b>4</b>	<b>-</b>	<b>8</b>	<b>155</b>		<b>9</b>

#### 4.2 Содержание дисциплины

## Лекция 1. Введение. Проблемы создания сложных программных систем.

Программные системы (ПС) как отрасль экономики. Проблемы создания ПС. Кризис программирования. Становление и развитие программной инженерии. Развитие технологий программирования. CASE-технологии.

## Лекция 2. Архитектуры программных систем.

Понятие архитектуры программной системы. Что определяет и на что влияет архитектура. Архитектурные структуры и представления. Модульные структуры. Структуры “компонент и соединитель”. Структуры распределения. Отношения между структурами. Варианты архитектур программных систем. Архитектура, основанная на уровнях абстракций. Архитектуры, основанные на портах. Архитектуры независимых компонентов. архитектуры, основанные на потоках данных.

Лабораторная работа №1. Техническое задание

## Лекция 3. Жизненный цикл программных систем (ПС).

Понятие жизненного цикла ПС. Основные процессы ЖЦ ПС. Вспомогательные процессы ЖЦ ПС. Организационные процессы ЖЦ ПС. Взаимосвязь между процессами ЖЦ ПС. Состав и стадии жизненного цикла ПС. Модели ЖЦ ПС. Каскадная модель (классический жизненный цикл). Итерационная модель ЖЦ ПС. Макетирование. Стратегии конструирования ПС. Инкрементная модель. Спиральная модель. Спиральная модель ЖЦ ПС “Рациональный унифицированный процесс” (RUP).

Лабораторная работа №2. Эскизный проект

## Лекция 4. Проектирование программных систем. Постановка требований к ПС.

Модель проектирования ПС как последовательная трансляция требований, предъявляемых к системе. Методология решения задач проектирования по Г. Майерсу. Уровни требований к программным системам. Определение требований к программным системам. Определение требований к программным системам. Функциональные и нефункциональные (эксплуатационные) требования. Процесс определения целей продукта и проекта. Разработка технического задания.

Лабораторная работа №3. Технический проект

## Лекция 5. Проектирование программных систем. Анализ требований и разработка внешних спецификаций.

Разработка предварительного внешнего проекта. Процесс внешнего проектирования. Проектирование взаимодействия с пользователем. Подготовка внешних спецификаций. Проверка правильности внешних спецификаций. Планирование изменений спецификаций. Анализ требований и определение спецификаций при структурном подходе к проектированию. Метод функционального моделирования. Функциональные диаграммы. Диаграммы потоков данных. Диаграммы переходов состояний. Анализ требований и

определение спецификаций при объектном подходе. Анализ требований и определение спецификаций при объектном подходе к проектированию.

Лабораторная работа №4. Стадия «реализация»

#### Лекция 6. Проектирование архитектуры программных систем.

Методология проектирования. Методы проектирования компонентных и модульных архитектур программных систем. Структурное проектирование. Модульность и ее характеристики. Оценка сложности модульных иерархических структур. Слои программного продукта. Метод восходящей разработки (“снизу-вверх”). Метод нисходящей разработки (“сверху-вниз”). Замечания по структурному проектированию. Формальное описание методики разработки модульной архитектуры программных систем. Пример проектирования структуры программной системы. Проектирование и программирование модулей. Проектирование программных систем при объектном подходе. Рефакторинг архитектуры программных систем. Паттерны системного проектирования.

Лабораторная работа №5. Тестирование программы

### **5. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ**

#### ***5.1. Внеаудиторная самостоятельная работа студентов***

В целом внеаудиторная самостоятельная работа студента при изучении курса включает в себя следующие виды работ:

- проработка (изучение) материалов лекций;
- чтение и проработка рекомендованной основной и дополнительной литературы;
- подготовка к практическим занятиям;
- поиск и проработка материалов из Интернет-ресурсов, научных публикаций;
- выполнение домашних заданий в форме подготовки докладов и рефератов;
- подготовка к текущему и итоговому (промежуточная аттестация) контролю знаний по дисциплине.

Основная доля самостоятельной работы студентов приходится на подготовку к практическим занятиям, тематика которых полностью охватывает содержание курса. Самостоятельная работа по подготовке к семинарским занятиям предполагает умение работать с первичной информацией.

### **6. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ**

Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине «Проектирование и архитектура программных систем» представлен в приложении к рабочей программе дисциплины и включает в себя:

- перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы;
- описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания;

- типовые контрольные задания или материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций;

- методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций.

### **Вопросы для проведения промежуточной аттестации по дисциплине (экзамен)**

1. Дайте характеристику особенностей создания программного продукта.
2. Перечислите особенности сложных программных систем.
3. Дайте характеристику современных методологий создания ПС.
4. Что понимается под технологией создания ПС?
5. Назовите основные этапы развития технологии проектирования ПС.
6. Какие языки программирования использовались в различных технологиях создания ПС?
7. Назовите причины неудачного завершения программных проектов.
8. Что такое архитектура ПС, как ее определить?
9. Какие виды архитектур ПС характерны для современных программных систем.
10. Какие методы и средства моделирования архитектуры ПС вы знаете? Дайте их характеристику.
11. Что такое жизненный цикл (ЖЦ) программного продукта?
12. Чем регламентируется ЖЦ программных систем (ПС)?
13. Какие группы процессов входят в состав ЖЦ ПС и какие процессы входят в состав каждой группы?
14. Какие из процессов, по вашему мнению, наиболее часто используются в реальных проектах, какие в меньшей степени и почему?
15. Назовите модели ЖЦ ПС.
16. В чем достоинства и недостатки каскадной модели ЖЦ ПС?
17. Каковы принципиальные особенности спиральной модели?
18. Как определить метод и технологию проектирования ПС?
19. Каким требованиям должна удовлетворять технология проектирования ПС?
20. Сравните стоимость исправления ошибок на различных стадиях разработки ПС.
21. Что такое управление требованиями?
22. Назовите методы выявления требований к ПС.

23. В чем заключаются основные принципы структурного подхода к определению требований?
24. Что общего и в чем различия между методом SADT и моделированием потоков данных?
25. Перечислите функциональные требования к ПС.
26. Перечислите эксплуатационные требования к ПС.
27. В чем отличия системных и функциональных требований к ПС?
28. Приведите пример диаграммы переходов состояний.
29. Приведите пример структурной схемы ПС.
30. В чем достоинства и недостатки структурного подхода к проектированию ПС?
31. В чем заключаются основные принципы объектно-ориентированного подхода к проектированию ПС?
32. Что такое CASE-технологии и CASE-средства?
33. В чем различие. Достоинства и недостатки методов проектирования сверху-вниз и снизу-вверх?
34. Охарактеризуйте метод пошаговой детализации.
35. Локальные сети: особенности, типы и характеристики
36. Структура и функции программного обеспечения ЛКС
37. Характеристика сетевого оборудования ЛКС
38. Принципы построения глобальных компьютерных сетей
39. Характеристика сети Internet
40. Семейство протоколов TCP/IP: состав и назначение
41. Способы адресации в IP-сетях
42. Характеристика и типовая структура корпоративных компьютерных сетей (ККС)
43. Программное обеспечение ККС: состав и назначение

## **7. РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА**

### ***7.1 Основная литература***

1. Вендров А. М. Проектирование программного обеспечения экономических информационных систем: Учебник. – М.: Финансы и статистика, 2002. – 352 с.
2. Вендров А. М. Практикум по проектированию программного обеспечения экономических информационных систем: Учеб. пособие. – 2-е изд., перераб. и доп. – М.: Финансы и статистика, 2006. – 192 с.

## **7.2 Дополнительная литература:**

3. Липаев В.В. Экономика производства сложных программных продуктов, М.: СИНТЕГ, 2008г. – 432 с.
4. Модели и методы исследования информационных систем : монография / А. Д. Хомоненко, А. Г. Басыров, В. П. Бубнов [и др.] ; под редакцией А. Д. Хомоненко. — Санкт-Петербург : Лань, 2022. — 204 с. — ISBN 978-5-8114-3675-0. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/206684> (дата обращения: 26.02.2023).
5. Кукарцев, В. В. Проектирование и архитектура информационных систем : учебник / В. В. Кукарцев, Р. Ю. Царев, О. А. Антамошкин. — Красноярск : СФУ, 2019. — 192 с. — ISBN 978-5-7638-3620-2. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/157581> (дата обращения: 06.02.2023).
6. Секунов Н.Ю. Разработка приложений на С++ и С#. - СПб: Питер., 2003г. – 608 с.

## **8. ПЕРЕЧЕНЬ РЕСУРСОВ ИНФОРМАЦИОННО-ТЕЛЕКОММУНИКАЦИОННОЙ СЕТИ «ИНТЕРНЕТ»**

Научная электронная библиотека eLIBRARY.RU – крупнейший российский информационный портал в области науки, технологии, медицины и образования, содержащий рефераты и полные тексты [www.elibrary.ru](http://www.elibrary.ru)

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

Методика преподавания данной дисциплины предполагает чтение лекций, проведение лабораторных занятий, групповых и индивидуальных консультаций по отдельным вопросам дисциплины. Предусмотрена самостоятельная работа студентов, а также прохождение аттестационных испытаний промежуточной аттестации.

Лекции посвящаются рассмотрению технологий и инструментов разработки программного продукта; изучению концепций и стратегий архитектурного проектирования и конструирования программного продукта; умению применять и разрабатывать основные стандарты информационно-коммуникационных систем и технологий.

Целью проведения лабораторных занятий является закрепление знаний студентов, полученных ими в ходе изучения дисциплины на лекциях и самостоятельно. Практические занятия проводятся в форме семинаров; на них обсуждаются вопросы по теме, разбираются конкретные ситуации по изучаемой теме, обсуждаются доклады. Для подготовки к занятиям семинарского типа студенты выполняют проработку рабочей программы, уделяя особое внимание целям и задачам, структуре и содержанию дисциплины; конспектирование источников; работу с конспектом лекций, просмотр рекомендуемой литературы.

При изучении дисциплины используются интерактивные методы обучения, такие как:

### **1. Лекция:**

– лекция-визуализация – подача материала осуществляется средствами технических средств обучения с кратким комментированием демонстрируемых визуальных материалов (презентаций).

2. Лабораторные работы – этот вид учебной работы в рамках которого осуществляется тот или иной эксперимент, направленный на получение результатов, имеющих значение с точки зрения успешного освоения студентами учебной программы.

## **10. КУРСОВОЙ ПРОЕКТ (РАБОТА)**

Выполнение курсового проекта (работы) не предусмотрено учебным планом.

## **11. ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, ИСПОЛЬЗУЕМЫХ ПРИ ОСУЩЕСТВЛЕНИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ, ВКЛЮЧАЯ ПЕРЕЧЕНЬ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ И ИНФОРМАЦИОННО-СПРАВОЧНЫХ СИСТЕМ**

### ***11.1 Перечень информационных технологий, используемых при осуществлении образовательного процесса***

- электронные образовательные ресурсы, представленные в п. 8 рабочей программы;
- использование слайд-презентаций;
- интерактивное общение с обучающимися и консультирование посредством электронной почты.

### ***11.2 Перечень программного обеспечения, используемого при осуществлении образовательного процесса***

- При освоении дисциплины используется лицензионное программное обеспечение:
- пакет Microsoft Office.

### ***11.3 Перечень информационно-справочных систем***

- справочно-правовая система Консультант-плюс <http://www.consultant.ru/online>
- справочно-правовая система Гарант <http://www.garant.ru/online>

## **12. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ**

– Для проведения занятий лекционного типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации используются учебные аудитории 7-519, 7-501 с комплектом учебной мебели.

– Для самостоятельной работы обучающихся, в том числе для курсового проектирования, используется кабинет 7-520; оборудованная комплектом учебной мебели, двумя компьютерами с доступом в информационно-телекоммуникационную сеть «Интернет» и в электронную информационно-образовательную среду организации, принтером и сканером.

- технические средства обучения для представления учебной информации: аудиторная доска, мультимедийное оборудование (ноутбук, проектор)
- наглядные пособия.