


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

Факультет информационных технологий, экономики и управления

Кафедра «Системы управления»

УТВЕРЖДАЮ

Декан ФИТЭУ

 /И.А. Рычка/

«28» января 2026 г.

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ**

«Современные технологии разработки программных комплексов»


Направление подготовки 09.04.01 «Информатика и вычислительная техника»  
(уровень магистратуры)

профиль:

«Программное обеспечение автоматизированных систем (в рыбохозяйственном комплексе)»

Рабочая программа разработана в соответствии с ФГОС ВО по направлению подготовки 09.04.01 «Информатика и вычислительная техника», профиль «Программное обеспечение автоматизированных систем(в рыбохозяйственном комплексе)», и учебного плана ФГБОУ ВО «КамчатГТУ».

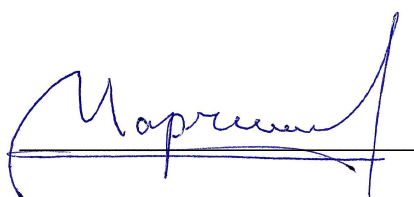
Составитель рабочей программы:

Профессор кафедры СУ, д.ф.-м..н.  Марापудец Ю.В.

Рабочая программа рассмотрена на заседании кафедры «Системы управления»

Протокол № 5 от «20» декабря 2025 года.

«20» декабря 2025 г.



Заведующий кафедрой  
«Системы управления»  
А.А. Марченко

## 1. ЦЕЛЬ И ЗАДАЧИ УЧЕБНОЙ ДИСЦИПЛИНЫ

**Цель** изучения дисциплины «Современные технологии разработки программных комплексов» является теоретическая и практическая подготовка студентов в области разработки программного обеспечения, включая изучение методов и подходов к оценке качества поставленных задач, их алгоритмизации, проектированию и разработке программного кода.

**Задачи** дисциплины:

- формирование систематизированного представления о методах и приемах формализации и алгоритмизации поставленных задач;
- Изучение методов анализа моделей теории массового обслуживания с использованием математических инструментов, таких как теория вероятностей, статистика и численные методы.
- формирование практических навыков разработки программного обеспечения с использованием современных сред разработки программного кода и систем контроля версий;
- приобретение знаний о методах и принципах принятия управленческих решений при разработке программных продуктов

## 2. ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Дисциплина «Современные технологии разработки программных комплексов» направлена на освоение следующих компетенций основной профессиональной образовательной программы по направлению 09.04.01 «Информатика и вычислительная техника» федерального государственного образовательного стандарта высшего образования:

- Способен разрабатывать компоненты системы управления базами данных (ПК-1).

Планируемые результаты обучения при изучении дисциплины, соотнесенные с планируемыми результатами освоения образовательной программы, представлены в таблице.

Код компетенции	Планируемые результаты освоения образовательной программы	Код и наименование индикатора	Планируемый результат обучения по дисциплине	Код показателя освоения
ПК-1	Способен разрабатывать компоненты системы управления базами данных	<b>ИД-1<sub>ПК-1</sub></b> Знает теорию баз данных	<b>Знать:</b> классификацию, жизненный цикл, стадии и этапы разработки программных комплексов	<b>З(ПК-1)1</b>
		<b>ИД-2<sub>ПК-1</sub></b> Умеет находить ошибки в работе системы управления базами данных	<b>Уметь:</b> ◦ спроектировать оптимальные алгоритм и структуру программных комплексов	<b>У(ПК-1)1</b>
		<b>ИД-3<sub>ПК-1</sub></b> Владеет навыками	<b>Владеть:</b> ◦ навыками	<b>В(ПК-1)1</b>

Код компетенции	Планируемые результаты освоения образовательной программы	Код и наименование индикатора	Планируемый результат обучения по дисциплине	Код показателя освоения
		устранения ошибок в компонентах системы управления базами данных по данным эксплуатации	разработки, тестирования и устранения ошибок современных программных комплексов	

### 3. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Учебная дисциплина Б1.В.01 «Современные технологии разработки программных комплексов» относится к части, формируемой участниками образовательных отношений в структуре образовательной программы.

### 4. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

#### 4.1. Тематический план дисциплины заочная форма

Наименование разделов и тем	Всего часов	Контактная работа	Контактная работа по видам учебных занятий			Самостоятельная работа	Формы текущего контроля	Итоговый контроль знаний по дисциплине
			Лекции	Семинары (практические занятия)	Лабораторные работы			
<b>Тема 1.</b> Классификация, жизненный цикл, стадии и этапы разработки программных комплексов и программной документации.	50	2	2	-	-	48	Опрос, ПЗ	
<b>Тема 2.</b> Принципы разработки программных комплексов на объектно-ориентированном языке программирования	85	<b>16</b>	2	-	14	69	Опрос, ПЗ, РЗ	
Экзамен		-	-	-	-	-	-	9
Всего	252	34	10		24	209		9

\*ПЗ – практическое задание, РЗ – решение задач, КС – конкретная ситуация

#### 4.2. Описание содержания дисциплины

**Тема 1. Классификация, жизненный цикл, стадии и этапы разработки программных комплексов и программной документации.**

Лекция 1.1. **Введение:** краткая характеристика дисциплины, цели и задачи изучения дисциплины, обзор литературы и рекомендуемая литература для освоения дисциплины. Классификация программных комплексов. Жизненный цикл программного комплекса. Стадии и этапы разработки программных комплексов и программной документации.

**СРС по теме 1.** Проработка теоретического материала по следующим темам:

- Основные концепции объектно-ориентированного программирования [1,2,6,7].
- Основные модели объектно-ориентированного программирования [1,2,6,7].
- Современные технологии разработки и тестирования программного обеспечения [1, 2].
- Жизненный цикл программных продуктов [3].

## **Тема 2. Принципы разработки программных комплексов на объектно-ориентированном языке программирования высокого уровня**

Лекция 2.1. **Системы программирования.** Основные компоненты систем программирования. Технология разработки и алгоритмы проектирования основных компонентов программных комплексов. Принципы разработки программного кода на объектно-ориентированном языке программирования C++.

Лабораторная работа 2.1 **Разработка программы, использующей общие принципы построения классов.** Разработка программы, использующей элементарные классы.

Цель работы: изучение принципов разработки программы, использующей элементарные классы с конструкторами и деструкторами.

Работа выполняется в соответствии с методическими указаниями по выполнению лабораторных работ. Лабораторная работа №1.

Содержание отчета:

1. Цель работы;
2. Исходные тексты разработанных программ;
3. Результаты работы программ (копии экранов);
4. Выводы по проделанной работе с указанием достоинств и недостатков предложенного исходного кода.

Лабораторная работа 2.2 **Разработка программы, использующей механизм наследования классов.** Разработка программы, использующей механизм наследования классов.

Цель работы: изучение принципов разработки программы, использующей механизм наследования классов.

Работа выполняется в соответствии с методическими указаниями по выполнению лабораторных работ. Лабораторная работа №2.

Содержание отчета:

1. Цель работы;
2. Исходные тексты разработанных программ;
3. Результаты работы программ (копии экранов);
4. Выводы по проделанной работе с указанием достоинств и недостатков предложенного исходного кода.

Лабораторная работа 2.3 **Разработка программы, использующей шаблоны классов.** Разработка программы, использующей шаблоны классов.

Цель работы: изучение принципов разработки программы, использующей шаблоны классов.

Работа выполняется в соответствии с методическими указаниями по выполнению лабораторных работ. Лабораторная работа №3.

Содержание отчета:

1. Цель работы;
2. Исходные тексты разработанных программ;

3. Результаты работы программ (копии экранов);
4. Выводы по проделанной работе с указанием достоинств и недостатков предложенного исходного кода.

Лабораторная работа 2.4 **Разработка программы, осуществляющей использование библиотеки STL на примере использования контейнера vector**. Разработка программ, использующих контейнер vector.

Цель работы: изучение принципов разработки программы, осуществляющей использование библиотеки STL на примере использования контейнера vector.

Работа выполняется в соответствии с методическими указаниями по выполнению лабораторных работ. Лабораторная работа №4

Содержание отчета:

1. Цель работы;
2. Исходные тексты разработанных программ;
3. Результаты работы программ (копии экранов);
4. Выводы по проделанной работе с указанием достоинств и недостатков предложенного исходного кода.

Лабораторная работа 2.5 **Разработка программы, осуществляющей использование контейнеров stack и queue**. Разработка программ, использующих контейнеры stack и queue.

Цель работы: изучение принципов разработки программы, осуществляющей использование контейнеров stack и queue.

Работа выполняется в соответствии с методическими указаниями по выполнению лабораторных работ. Лабораторная работа №5

Содержание отчета:

1. Цель работы;
2. Исходные тексты разработанных программ;
3. Результаты работы программ (копии экранов);
4. Выводы по проделанной работе с указанием достоинств и недостатков предложенного исходного кода.

Лабораторная работа 2.6 **Разработка программы, осуществляющей использование контейнеров deque и list**. Разработка программ, использующих контейнеры deque и list. (4 часа).

Цель работы: изучение принципов разработки программы, осуществляющей использование контейнеров deque и list.

Работа выполняется в соответствии с методическими указаниями по выполнению лабораторных работ. Лабораторная работа №6

Содержание отчета:

1. Цель работы;
2. Исходные тексты разработанных программ;
3. Результаты работы программ (копии экранов);
4. Выводы по проделанной работе с указанием достоинств и недостатков предложенного исходного кода.

Лабораторная работа 2.7 **Разработка программы, осуществляющей использование контейнеров, описанных в заголовках <set> и <map>**. Разработка программ, использующих контейнеры, описанные в заголовках <set> и <map>.

Цель работы: изучение принципов разработки программы, осуществляющей использование контейнеров, описанных в заголовках <set> и <map>.

Работа выполняется в соответствии с методическими указаниями по выполнению лабораторных работ. Лабораторная работа №7.

Содержание отчета:

1. Цель работы;
2. Исходные тексты разработанных программ;

3. Результаты работы программ (копии экранов);
4. Выводы по проделанной работе с указанием достоинств и недостатков предложенного исходного кода.

**СРС по теме 2.** Проработка теоретического материала по следующим темам:

1. Объектно-ориентированное программирование на языке C++ [1,2, 4-9].
2. Современные библиотеки C++ [1].

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

В целом внеаудиторная самостоятельная работа обучающегося 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 и лабораторным работам;  
поиск и проработка материалов из Интернет-ресурсов, периодической печати;  
выполнение домашних заданий в форме творческих (проблемно-поисковых, групповых) заданий, докладов;
- подготовка к текущему и итоговому (промежуточная аттестация) контролю знаний по дисциплине.

Основная доля самостоятельной работы обучающихся приходится на подготовку к практическим занятиям и лабораторным работам, тематика которых полностью охватывает содержание курса. Самостоятельная работа по подготовке к практическим занятиям и лабораторным работам 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 «Программирование» представлен в приложении к рабочей программе дисциплины и включает в себя:

1. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы;
2. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания;
3. Типовые контрольные задания или материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций;
4. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций.

Фонд оценочных средств содержит:

1. Тестовые задания для контроля усвоения материала.
2. Перечень вопросов к промежуточной аттестации.
3. Комплект билетов к промежуточной аттестации.

### **Примерный перечень вопросов к промежуточной аттестации**

1. Факторы, обусловившие появление и содержание концепции ООП. Основные свойства ООП: абстракция, инкапсуляция, наследование, полиморфизм.
2. История возникновения языка C++. Общий синтаксис объектно-ориентированного C++, отличия от языка Си. Типы данных.

3. Операции, операторы, препроцессор в C++.
4. Указатели, адреса и массивы.
5. Особенности работы с памятью в языке C++. Статическое и динамическое выделение памяти.
6. Определение внешней функции. Вызов функции. Использование и определение параметров.
7. Подключение библиотек. Стандартные (CRT) библиотеки языка C++.
8. Структуры. Динамические структуры данных.
9. Классы: определение, описание, члены класса. Управление доступом к элементам классов, область определения элементов.
10. Конструкторы и деструкторы. Дружественные функции класса. Статические элементы класса.
11. Основы механизма наследования классов.
12. Виртуальные функции и классы. Абстрактные классы.
13. Шаблоны классов.
14. Современные библиотеки языка C++.
15. Библиотеки потоковых классов ввода-вывода.
16. Особенности ввода-вывода в файл при использовании потоковых классов.
17. Библиотека шаблонов STL. Общее описание. Контейнеры, итераторы, алгоритмы.
18. Библиотека `<string>` для работы со строками.
19. Контейнер `vector`. Библиотека `<vector>`: особенности, стандартные имена и методы.
20. Контейнер `stack`. Библиотека `<stack>`: особенности, стандартные имена и методы.
21. Контейнер `queue`. Библиотека `<queue>`: особенности, стандартные имена и методы.
22. Контейнер `deque`. Библиотека `<deque>`: особенности, стандартные имена и методы.
23. Контейнер `list`. Библиотека `<list>`: особенности, стандартные имена и методы.
24. Контейнеры `set`, `multiset`. Библиотеки `<set>`, `<multiset>`: особенности, стандартные имена и методы.
25. Контейнер `bitset`. Библиотека `<bitset>`. Использование пар и кортежей.
26. Контейнеры `map`, `multimap`. Библиотеки `<map>`, `<multimap>`: особенности, стандартные имена и методы.
27. Принципы использования аллокаторов для работы с контейнерами.
28. Принципы использования итераторов для работы с контейнерами.
29. Библиотека `<algorithm>`, основные функции.
30. Средства обработки исключений в языке C++.

## 7. РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

### 7.1. Основная литература

1. *Марапулец Ю.В.* Язык C++. Основы программирования. (Издание второе, исправленное и дополненное, рекомендовано ДВ РУМЦ в качестве учебного пособия). Петропавловск-Камчатский: КамГУ им. Витуса Беринга, 2019. – 158 с.
2. *Страуструп Б.* Язык программирования C++. Четвертое издание. М.: БИНОМ. Лаборатория знаний, 2023. – 1216 с.
3. *Кнут Д.* Искусство программирования для ЭВМ. Ч.1. М.: Мир, 1976. – 726 с.

### 7.2. Дополнительная литература

4. *Карпов Б., Баранова Т.* C++. Специальный справочник. С.-Пб.: Питер, 2001. – 479 с.
5. *Климова Л.М.* C++. Практическое программирование. М.: КУДИЦ-ОБРАЗ, 2001. – 587 с.

6. *Марапулец Ю.В.* Программирование на языке высокого уровня. Рекомендовано ДВ РУМЦ в качестве учебного пособия. Петропавловск-Камчатский: Издательство КамчатГТУ, 2008 - 189 с.
7. *Лафоре Р.* Объектно-ориентированное программирование в C++. Классика Computer Science. Четвертое издание. СПб.: Питер, 2004. – 924 с.
8. *Романов Е.Л.* Практикум по программированию на C++. С.-Пб.:БХВ-Петербург, 2004. – 427 с.
9. *Д.Круглински, С.Уингоу, Д.Шефферд.* Программирование на Microsoft Visual C++ для профессионалов. С.-Пб.: Питер, 2004 г.

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

1. Электронно-библиотечная система «eLibrary»: [Электронный ресурс]. - Режим доступа: <http://www.elibrary.ru>
2. Официальный сайт компании Microsoft по изучению языка C++: [Электронный ресурс]. - Режим доступа: <https://learn.microsoft.com/ru-ru/cpp/cpp/?view=msvc-160>
3. Руководства и описание языка C++ от его создателя Бьярна Страуструпа: [Электронный ресурс]. - Режим доступа: <https://www.stroustrup.com/C++.html>
4. Руководства и описание языка C++ на интернет ресурсе для программистов Хабр: [Электронный ресурс]. - Режим доступа: <https://habr.com/ru/hub/cpp/>
5. Онлайн справочник программиста на языке C++: [Электронный ресурс]. - Режим доступа: <http://www.c-cpp.ru/>

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

### **Подготовка к лекционным занятиям**

Лекции составляют основу теоретического обучения и дают систематизированные основы научных знаний по дисциплине, концентрируют внимание обучающихся на наиболее сложных и узловых вопросах, стимулируют их активную познавательную деятельность и способствуют формированию творческого мышления.

При подготовке к лекции следует предварительно ознакомиться с учебным материалом по теме занятия и при конспектировании лекции акцентировать внимание на новых теоретических положениях и иных данных, не нашедших отражения в учебной литературе.

Для успешного изучения дисциплины студенту рекомендуется систематически готовиться к каждому занятию по следующей схеме:

- повторить материал предыдущей лекции, используя конспекты, учебную и специальную литературу
- ответить на контрольные вопросы по изучаемой теме.

### **Подготовка к лабораторным занятиям**

Лабораторная работа– это выполнение студентами под руководством преподавателя или по инструкции заданий(решение задач, написание программ) с применением персонального компьютера.

В ходе лабораторных работ студенты воспринимают и осмысливают новый учебный материал. Лабораторные занятия носят систематический характер, регулярно следуя за лекционными занятиями. Лабораторные работы выполняются согласно графику, при этом соблюдается принцип индивидуального выполнения работ.

Обучающийся должен подготовить отчет к каждой лабораторной работе, предусмотренной планом.

При подготовке к лабораторным занятиям необходимо заранее изучить методические рекомендации по его проведению. Обратит внимание на цель занятия, на основные вопросы для подготовки к занятию, на содержание темы занятия.

### **Подготовка к практическим занятиям**

Практическое занятие – это занятие, проводимое под руководством преподавателя в учебной аудитории, направленное на углубление научно-теоретических знаний и овладение определенными методами самостоятельной работы. В процессе таких занятий вырабатываются практические умения. Практические занятия призваны углубить и расширить знания, полученные в ходе лекций. Практические занятия носят систематический характер, регулярно следуя за лекционными занятиями.

Для успешной подготовки к практическим занятиям требуется предварительная самостоятельная работа студентов по теме планируемого занятия. Студенты должны предварительно поработать над конспектом, учебником, учебным пособием, чтобы овладеть теорией вопроса.

Обучающийся должен подготовить отчет по каждому практическому занятию, предусмотренному планом.

### **Перечень методических указаний к проведению учебных, лабораторных занятий и самостоятельной работы студентов**

1. Марапулец Ю.В. Современные технологии разработки программных комплексов. Методическое пособие к выполнению лабораторных работ по направлению подготовки бакалавров 09.04.01 «Информатика и вычислительная техника» по дисциплине «Современные технологии разработки программных комплексов» для студентов очной формы обучения, Петропавловск-Камчатский: КамчатГТУ, 2024. – 78 с. (электронная форма).

## **10. КУРСОВОЙ ПРОЕКТ (РАБОТА)**

В соответствии с учебным планом курсовое проектирование по дисциплине «Современные технологии разработки программных комплексов» не предусмотрено.

### **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 в электронной информационной образовательной среде.

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

При освоении дисциплины используется лицензионное программное обеспечение:

- операционные системы Astra Linux (или иная операционная система, включенная в реестр отечественного программного обеспечения);
- комплект офисных программ Р-7 Офис (в составе текстового процессора, программы работы с электронными таблицами, программные средства редактирования и демонстрации презентаций);
- компилятор языка С++;
- программа проверки текстов на предмет заимствования «Антиплагиат».

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

Лекционные занятия по данной дисциплине проводятся в учебной аудитории 7-518 с комплектом учебной мебели. Лабораторные и практические занятия проводятся в лаборатории разработки программного обеспечения (учебная аудитория 7-513), оборудованной 10 рабочими станциями с программным обеспечением, представленным в п. 11.2, доступом к сети «Интернет», в электронную информационно-образовательную среду и комплектом учебной мебели.