


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/

« 31 » января 2024 г.

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ**

**«Объектно-ориентированное программирование»**

направление подготовки:

09.03.01 «Информатика и вычислительная техника»  
(уровень бакалавриата)

направленность (профиль):

«Программное обеспечение средств вычислительной техники и  
автоматизированных систем»

Петропавловск-Камчатский  
2024

Рабочая программа разработана в соответствии с ФГОС ВО по направлению подготовки 09.03.01 «Информатика и вычислительная техника», профиль «Программное обеспечение средств вычислительной техники и автоматизированных систем», и учебного плана ФГБОУ ВО «КамчатГТУ».

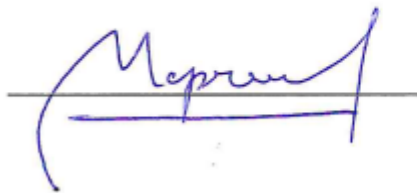
Составитель рабочей программы:

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

Рабочая программа рассмотрена на заседании кафедры «Системы управления»

Протокол № 6 от « 31 » января 2024 года.

« 31 » января 2024 г.



Заведующий кафедрой  
«Системы управления»  
А.А. Марченко

## 1. ЦЕЛЬ И ЗАДАЧИ УЧЕБНОЙ ДИСЦИПЛИНЫ

**Целью** дисциплины является ознакомление студентов с парадигмой и технологией объектно-ориентированного программирования (ООП) и сопутствующих технологий программирования, а так же - обучение студентов основам ООП на языке программирования высокого уровня С++.

**Задачей** дисциплины является дать студенту теоретические и практические знания по принципам объектно-ориентированного проектирования и разработки объектно-ориентированного программного обеспечения ЭВМ.

В результате изучения дисциплины:

- студент должен знать основные концепции объектно-ориентированного программирования, лексические и синтаксические основы объектно-ориентированного языка программирования С++, принципы построения классов и объектов, конструкторы и деструкторы, виртуальные методы и классы, абстрактные классы, принципы и виды наследования классов, шаблоны классов, обработку исключительных ситуаций, методику объектно-ориентированного анализа и проектирования..

- студент должен уметь создавать объектно-ориентированные программы с применением классов на языке С++, создавать иерархию классов на С++, использовать полиморфизм, проектировать с учетом множественного наследования, создавать шаблоны функции и классов, использовать механизм обработки исключений, использовать библиотеку потоковых классов, производить отладку программного кода в среде разработчика Visual С++

- студент должен приобрести навыки по объектно-ориентированному проектированию и разработке объектно-ориентированного программного кода в современных операционных системах.

## 2. ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Дисциплина «Объектно-ориентированное программирование» направлена для освоения следующих компетенций основной образовательной программы подготовки бакалавра по направлению подготовки 09.03.01 "Информатика и вычислительная техника" федерального государственного образовательного стандарта высшего образования:

- способностью к разработке программных интерфейсов (ПК-3). Планируемые результаты обучения при изучении дисциплины, соотнесенные с планируемыми результатами освоения образовательной программы, представлены в таблице.

Таблица - Планируемые результаты обучения при изучении дисциплины, соотнесенные с планируемыми результатами освоения образовательной программы

Код компетенции	Наименование компетенции	Код и наименование индикатора достижения	Планируемый результат обучения по дисциплине	Код показателя освоения
ПК-3	Обладать способностью разрабатывать программные интерфейсы	<b>ИД-1</b> <sub>ПК-3</sub> Знает методы и средства сборки модулей и компонент программного обеспечения.	<b>Знать:</b> – основные концепции объектно-ориентированного программирования, методику объектно-ориентированного анализа и проектирования;	<b>3(ПК-3)1</b>

			<ul style="list-style-type: none"> <li>– лексические и синтаксические основы объектно-ориентированного языка программирования C++, принципы построения классов и объектов, принципы и виды наследования классов, шаблоны классов, обработку исключительных ситуаций;</li> <li>– принципы разработки объектно-ориентированного ПО в среде разработчика Visual C++.</li> </ul>	<p><b>З(ПК-3)2</b></p> <p><b>З(ПК-3)3</b></p>
		<p><b>ИД-2</b><sub>ПК-3</sub> Знает методы и средства разработки процедур для развертывания программного обеспечения</p>	<p><b>Знать:</b> принципы разработки объектно-ориентированного ПО с консольным интерфейсом; принципы разработки объектно-ориентированного ПО с оконным интерфейсом на базе функций WinAPI; принципы разработки объектно-ориентированного ПО с оконным интерфейсом с использованием классов MFC.</p>	<p><b>З(ПК-3)41</b></p> <p><b>З(ПК-3)5</b></p> <p><b>З(ПК-3)6</b></p>
		<p><b>ИД-3</b><sub>ПК-3</sub> Умеет писать программный код процедур интеграции программных модулей.</p>	<p><b>Уметь:</b> спроектировать алгоритм и структуру объектно-ориентированного ПО; уметь разрабатывать объектно-ориентированный программный код на языке C++; уметь разрабатывать объектно-ориентированный программный код в</p>	<p><b>У(ПК-3)1</b></p> <p><b>У(ПК-3)2</b></p> <p><b>У(ПК-3)3</b></p>

			среде разработчика Visual C++.	
		<b>ИД-4</b> <sub>ПК-3</sub> Владеет навыками использования выбранной среды программирования для разработки процедур интеграции программных модулей.	<b>Владеть:</b> навыками по разработке структуры объектно-ориентированного ПО; навыками по разработке по разработке объектно-ориентированного ПО на языке C++; навыками по разработке объектно-ориентированного ПО в среде разработчика Visual C++.	<b>В(ПК-2)1</b>  <b>В(ПК-3)2</b>  <b>В(ПК-3)3</b>

### 3. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Дисциплина Б1.В.06 "Объектно-ориентированное программирование" относится к части, формируемой участниками образовательных отношений в структуре образовательной программы.

### 4. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

#### 4.1. Тематический план дисциплины

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> Теоретические основы объектно-ориентированного программирования	27	12	10	-	2	15	Опрос, ПЗ	
<b>Тема 2:</b> Основы объектно-ориентированного программирования на языке C++	64	36	18	-	18	28	Опрос, ПЗ, РЗ	
<b>Тема 3:</b> Принципы разработки объектно-ориентированных программ в среде разработчика Visual C++	53	20	6	-	14	33	Опрос, ПЗ, РЗ	
Зачет, КР		-	-	-	-	-	-	20+2
Всего	144	68	34		34	76		

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

## 4.2. Описание содержания дисциплины

### Тема 1: Теоретические основы объектно-ориентированного программирования

Лекция 1.1 **Введение.** Предмет дисциплины и ее задачи. Роль и место задач разработки программного обеспечения в интегрированных производственных комплексах, автоматизированных системах управления техническими объектами. Структура и содержание дисциплины, ее связь с другими дисциплинами специальности. Обзор рекомендуемой литературы.

Лекция 1.2 **Основные концепции объектно-ориентированного программирования.**

Факторы, обусловившие появление и содержание концепции ООП. Основные идеи ООП: использование объекта в качестве основной компоненты программы и децентрализация управления, реализуемое представлением программы как описания взаимодействия объектов. Инкапсуляция, наследование, полиморфизм. Объектно-ориентированный подход к разработке программ. Место и роль ООП в теории и практике разработки программных систем.

Лекция 1.3 **Основные модели объектно-ориентированного программирования.**

Объект как совокупность данных и набора операций. Семантика объекта. Представление данных. Классификация методов: конструкторы, деструкторы, селекторы и модификаторы. Классы объектов: назначение и семантика. Класс как абстракция совокупности объектов. Классы и абстрактные типы данных. Объекты как экземпляры классов. Основные действия с объектами: создание, инициализация, использование, уничтожение. Отношение наследования для классов. Простое и множественное наследование. Иерархия классов.

Лекция 1.4 **Современные технологии разработки и тестирования программного обеспечения.**

Объектно-ориентированный стиль программирования. Инструментальные средства автоматизации проектирования программных систем (CASE-средства). Графический подход к решению проблемы автоматизации разработки программного обеспечения. Требования качеству современных программных средств. Среда Visual C++ как пример современного автоматизированного средства разработки. Основные элементы управления в Visual C++ , создание и разработка проекта. Типы мастеров проектов. Debug и Release проекты.

Лабораторная работа 1.1 **Изучение основных элементов управления среды разработчика. Разработка первого проекта.** Основные элементы управления проектом. Состав программы. Включение препроцессора. Использование функции main. Ввод данных, решение математического выражения и вывод результата на экран.

Цель работы: Изучение принципов разработки программы в среде Visual C++.

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.

Лекция 1.5 **Жизненный цикл программных продуктов.**

Процесс разработки программ. Основные фазы проектирования программного обеспечения. Структура жизненного цикла. Модели жизненного цикла. Этапы разработки объектно-ориентированной системы.

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

1. Основные концепции объектно-ориентированного программирования [1-4].
2. Основные модели объектно-ориентированного программирования [1-3].
3. Современные технологии разработки и тестирования программного обеспечения [1].
4. Жизненный цикл программных продуктов [1].

**Тема 2:** Основы объектно-ориентированного программирования на языке C++

Лекция 2.1 **Реализация концепции объектно-ориентированного программирования в языке программирования C++.**

История возникновения языка C++. Место языка C++ во внедрении и развитии средств ООП. Значение языка C++ для профессиональной подготовки программистов. Структура программы. Ввод и вывод информации.

Лабораторная работа 2.1 **Разработка программ с операторами, указателями и массивами.** Разработка программ, использующих операторы, указатели и массивы.

Цель работы: Изучение принципов разработки программы, использующей операторы, указатели и массивы.

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.

Лекция 2.2 **Общий синтаксис языка C++.**

Типы данных и операции C++. Операторы языка. Указатели и адресная арифметика. Массивы и функции. Выделение и освобождение памяти. Макросы. Подстановка лексем. Включение файлов. Заголовочные файлы. Способы размещения программы в файлах.

Лабораторная работа 2.2 **Разработка программ с операциями ввода-вывода в файл, динамическим выделением памяти и внешними функциями.** Разработка программ, использующих файловый ввод-вывод, динамическое выделение памяти и внешние функции.

Цель работы: Изучение принципов разработки программ с операциями ввода-вывода в файл, динамическим выделением памяти и внешними функциями.

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.

Лекция 2.3 **Структуры данных.**

Структуры и объединения. Доступ к элементам структуры. Объединения. Битовые поля структур и объединений. Динамические структуры данных: стеки, очереди, списки, бинарные деревья.

Лабораторная работа 2.3 **Разработка программы, использующей структуры.** Разработка программы, использующей структуры.

Цель работы: Изучение принципов разработки программы, использующей структуры.

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.

#### Лекция 2.4 **Общие принципы создания классов.**

Способы реализации инкапсуляции. Определение класса. Личная и общая части определения класса. Функции-элементы класса и функции-друзья. Объекты класса. Статические члены объектов класса. Вложенные и локальные классы. Примеры описания и использования классов.

Лабораторная работа 2.4 **Разработка программы, использующей общие принципы построения классов.** Разработка программы, использующей элементарные классы.

Цель работы: Изучение принципов разработки программы, использующей элементарные классы.

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.

#### Лекция 2.5 **Конструкторы и деструкторы.**

Принципы создания и уничтожения объектов. Свойства конструкторов и деструкторов. Конструкторы по умолчанию, с параметрами и копирования. Выделение и уничтожение динамических объектов класса.

Лабораторная работа 2.5 **Разработка программы, использующей конструкторы и деструкторы.** Разработка программы, использующей конструкторы и деструкторы классов.

Цель работы: Изучение принципов разработки программы, использующей конструкторы и деструкторы классов.

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.

#### Лекция 2.6 **Наследование классов.**

Базовый и производный классы. Функции-элементы и функции-друзья. Правила доступа к элементам производного класса. Иерархия классов. Одиночное и множественное наследование. Виртуальные базовые классы. Особенности доступа при множественном наследовании. Виртуальные функции. Чистые виртуальные функции и абстрактные базовые классы.

Лабораторная работа 2.6 **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 №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.

**Лекция 2.7 Шаблоны классов.**

Шаблоны классов и функций. Наследование шаблонных классов. Правила отождествления параметров шаблона. Контейнерные классы. Применение шаблонных классов для создания контейнерных классов.

Лабораторная работа 2.7 **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 №8.

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.

**Лекция 2.8 Поточковые классы ввода-вывода.**

Использование функций ввода-вывода языка C++. Форматы ввода-вывода. Стандартный ввод-вывод. Ввод-вывод в файл. Строковый ввод-вывод. Ошибочные состояния потоков ввода-вывода.

Лабораторная работа 2.8 **Разработка программы, использующей потоковые классы ввода-вывода в файл.** Разработка программы, осуществляющей ввод-вывод в файл с помощью потоков.

Цель работы: Изучение принципов разработки программы, осуществляющей ввод-вывод в файл с помощью потоков.

Работа выполняется в соответствии с методическими указаниями по выполнению лабораторных работ. Лабораторная работа №9.

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.

**Лекция 2.9 Обработка исключительных ситуаций.**

Анализ различных моделей обработки исключительных ситуаций. Стандартные средства контроля подтверждений. Проверка предусловий и постусловий, вычисления инвариантов. Контроль асинхронных событий. Реализация модели обработки синхронных ситуаций с завершением в C++. Возбуждение ситуации, Описание блоков с контролем и реакций на ситуации. Система классов для описания исключительных ситуаций.

Лабораторная работа 2.9 **Разработка программы, использующей технологии обработки исключительных ситуаций.** Разработка программы, осуществляющей обработку исключений.

Цель работы: Изучение принципов разработки программы, осуществляющей обработку исключений.

Работа выполняется в соответствии с методическими указаниями по выполнению лабораторных работ. Лабораторная работа №10.

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. Реализация концепции объектно-ориентированного программирования в языке программирования C++ [1-9].
2. Общий синтаксис языка C++ [1-9].
3. Структуры данных [1-9].
4. Общие принципы создания классов [1,2, 4, 7-9].
5. Конструкторы и деструкторы [1,2, 4, 7-9].
6. Наследование классов [1,2, 4, 7-9].
7. Шаблоны классов [1, 4, 7-9].
8. Поточные классы ввода-вывода [1, 4].
9. Обработка исключительных ситуаций [1, 4].

**Тема 3:** Принципы разработки объектно-ориентированных программ в среде разработчика Visual C++

**Лекция 3.1 Разработка проектов на основе функций Win API.**

Виды интерфейсов API. Принципы разработки проекта на основе Win API. Типы мастеров проекта. Технологии разработки консольных и оконных интерфейсов. Использование в проекте DLL.

**Лекция 3.2 Разработка проектов на основе классов MFC.**

Типы мастеров проекта. Изменение установок проекта. Иерархия классов MFC. Функции MFC. Обработка сообщений в MFC. Анализ классов проекта. Создание ресурсов. Типы диалогов. Введение элементов в диалоговое окно. Виды применяемых элементов. Управление мышью. Поддержка вывода на принтер. Запись в файл.

Лабораторная работа 3.1 **Разработка приложения на основе интерфейса SDI.** Разработка программы-прототипа текстового редактора (4 часа).

Цель работы: Изучение принципов разработки программы простого текстового редактора на основе интерфейса SDI в среде Visual C++.

Работа выполняется в соответствии с методическими указаниями по выполнению лабораторных работ. Лабораторная работа №11.

Содержание отчета:

Цель работы;

Исходные тексты разработанных программ;

Результаты работы программ (копии экранов);

Выводы по проделанной работе с указанием достоинств и недостатков предложенного исходного кода.

Лабораторная работа 3.2 **Включение ресурсов в приложение.** Разработка программы-прототипа с различными ресурсами (4 часа).

Цель работы: Изучение принципов включения ресурсов в разрабатываемое приложение.

Работа выполняется в соответствии с методическими указаниями по выполнению лабораторных работ. Лабораторная работа №12.

Содержание отчета:

Цель работы;

Исходные тексты разработанных программ;

Результаты работы программ (копии экранов);

Выводы по проделанной работе с указанием достоинств и недостатков предложенного исходного кода.

Лабораторная работа 3.3 **Создание приложения на базе диалогового окна.**  
Разработка программы на базе диалогового окна (4 часа).

Цель работы: Изучение принципов создания приложения на базе диалогового окна.

Работа выполняется в соответствии с методическими указаниями по выполнению лабораторных работ. Лабораторная работа №13.

Содержание отчета:

Цель работы;

Исходные тексты разработанных программ;

Результаты работы программ (копии экранов);

Выводы по проделанной работе с указанием достоинств и недостатков предложенного исходного кода.

Лекция 3.3 **Отладка программ в Visual C++.**

Структура отладчика программ. Установка метки. Окна просмотра промежуточных результатов. Пошаговое выполнение программы.

Лабораторная работа 3.4 **Отладка программ с помощью встроенных средств отладчика.** Устранение ошибок с помощью отладчика программ.

Цель работы: Изучение принципов отладки программ в Visual C++.

Работа выполняется в соответствии с методическими указаниями по выполнению лабораторных работ. Лабораторная работа №14.

Содержание отчета:

Цель работы;

Исходные тексты разработанных программ;

Результаты работы программ (копии экранов);

Выводы по проделанной работе с указанием достоинств и недостатков предложенного исходного кода.

**СРС по теме 3.** Проработка теоретического материала по следующим темам:

1. Разработка проектов на основе функций WinAPI [10-13].
2. Разработка проектов на основе классов MFC [14].
3. Отладка программ в Visual C++ [1, 4].

## **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 практическими 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 «Объектно-ориентированное 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.
1. Перечень вопросов к промежуточной аттестации.

### **Примерный перечень вопросов к промежуточной аттестации (зачет)**

1. Факторы, обусловившие появление и содержание концепции ООП.
2. Основные свойства ООП: абстракция, инкапсуляция, наследование, полиморфизм.
3. Основные фазы проектирования программного обеспечения. Структура жизненного цикла.
4. Модели жизненного цикла. Этапы разработки объектно-ориентированной системы.
5. История возникновения языка C++. Общий синтаксис языка C++. Разделители. Использование комментариев. Структура идентификаторов.
6. Этапы подготовки программы на языке C++ к исполнению на ПЭВМ.
7. Резервированные слова.
8. Директивы препроцессора.
9. Основные типы данных. Размер данных.
10. Определение переменных. Локальные, статические и др. типы переменных.
11. Арифметические операции. Операции присваивания.
12. Операции отношения. Логические операции. Побитовые операции.
13. Оператор-выражение. Основные виды операторов.
14. Условные операторы.
15. Операторы цикла.
16. Оператор переключения.
17. Указатели и адреса.
18. Описание массива. Массивы в качестве параметров. Строки. Массивы указателей.
19. Определение внешней функции. Вызов функции. Использование и определение параметров. Функция main.
20. Подключение библиотек. Основные библиотеки языка C++.
21. Библиотеки ввода-вывода. Основные функции ввода-вывода.

22. Ввод-вывод в файл, доступ к файлам. Форматы записи данных.
23. Математическая библиотека. Основные функции.
24. Библиотеки для работы с символами и строками. Основные функции.
25. Библиотека общего назначения. Основные функции.
26. Технологии и функции для динамического выделения памяти.
27. Системные библиотеки. Основные функции.
28. Перечисления.
29. Структуры и объединения.
30. Классы: определение, описание, члены класса. Управление доступом к элементам классов, область определения элементов.
31. Конструкторы и деструкторы. Дружественные функции класса.
32. Основы механизма наследования классов.
33. Шаблоны классов.
34. 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. *Лафоре Р.* Объектно-ориентированное программирование в C++. С.-Пб.: Питер, 2018 г. – 928 с.
3. *Буч Г.* и др. Объектно-ориентированный анализ и проектирование с примерами приложений. М.: Вильямс, 2017 г. – 720 с.

### **7.2. Дополнительная литература**

4. *Марапулец Ю.В.* Программирование на языке высокого уровня. Рекомендовано ДВ РУМЦ в качестве учебного пособия. Петропавловск-Камчатский: Издательство КамчатГТУ, 2008 - 189 с.
5. *Подбельский В.В.* Язык Си++. М.: Финансы и статистика, 2005 г. – 559 с.
6. *Карнов Б., Баранова Т.* C++. Специальный справочник. С.-Пб.: Питер, 2001. – 479 с.
7. *Климова Л.М.* C++. Практическое программирование. М.: КУДИЦ-ОБРАЗ, 2001. – 587 с.
8. *Романов Е.Л.* Практикум по программированию на C++. С.-Пб.: БХВ-Петербург, 2004. – 427 с.
9. *Страуструп Б.* Язык программирования Си++. М.: Радио и связь, 1991. – 352 с.
10. *Финогенов К.Г.* Win32. Основы программирования. М.: Диалог-МИФИ, 2004. - 416 с.
11. *Рихтер Д.* Windows. Создание эффективных Win32-приложений с учетом специфики 64-разрядной версии Windows. С.-Пб.: Питер, 2005. - 624 с.
12. *Д.Круглински, С.Уингоу, Д.Шефферд.* Программирование на Microsoft Visual C++ для профессионалов. С.-Пб.: Питер, 2004 г.
13. *Щупак Ю.А.* Win32 API. Эффективная разработка приложений. С.-Пб.: Питер, 2007.-572с.
14. *Черносвитов А.* Visual C++ и MFC. С.-Пб.: Питер, 2005 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.

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

1. Марапулец Ю.В. Объектно-ориентированное программирование. Методическое пособие к выполнению лабораторных работ по направлению подготовки бакалавров 09.03.01 «Информатика и вычислительная техника» по дисциплине «Объектно-ориентированное программирование» для студентов очной формы

обучения – Петропавловск-Камчатский: КамчатГТУ, 2015. – 95 с. (электронная форма).

## 10. КУРСОВОЙ ПРОЕКТ (РАБОТА)

Рекомендуется следующая тематика курсовых проектов:

1. Реализация и графическое представление возможностей механизма наследования классов с использованием базового класса Круг и порожденного класса Кольцо.
2. Реализация и графическое представление возможностей механизма наследования классов с использованием базового класса Прямоугольник и порожденного класса Прямоугольник с кругом.
3. Реализация и графическое представление возможностей механизма наследования классов с использованием базового класса Треугольник и порожденного класса Четырехугольник.
4. Реализация и графическое представление возможностей механизма наследования классов с использованием базового класса Вектор на плоскости и порожденного класса Вектор и точка на плоскости.
5. Реализация и графическое представление возможностей механизма наследования классов с использованием базового класса Вектор на плоскости и порожденного класса Два вектора с общим началом.
6. Реализация и графическое представление возможностей механизма наследования классов с использованием базового класса Вектор на плоскости и порожденного класса Два параллельных вектора на плоскости.
7. Реализация и графическое представление возможностей механизма наследования классов с использованием базового класса Круг и порожденного класса Дуга окружности.
8. Реализация возможностей объектно-ориентированного подхода с использованием структуры Комплексное число и класса Произведение комплексных чисел.
9. Реализация возможностей объектно-ориентированного подхода с использованием класса Время и дочернего класса Расписание движения поездов.
10. Реализация и графическое представление возможностей механизма наследования классов с использованием базового класса Прямоугольник и порожденного класса Прямоугольник со скругленными углами.

## 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 AstraLinux (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 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.



### 13 ДОПОЛНЕНИЯ И ИЗМЕНЕНИЯ В РАБОЧЕЙ ПРОГРАММЕ

Дополнения и изменения в рабочей программе за \_\_\_\_ / \_\_\_\_ учебный год

В рабочую программу \_\_\_\_\_  
(наименование дисциплины)

для специальности (тей) \_\_\_\_\_  
(номер специальности)

вносятся следующие дополнения и изменения:

Дополнения и изменения внес \_\_\_\_\_  
(должность, Ф.И.О., подпись)

Рабочая программа пересмотрена и одобрена на заседании кафедры  
\_\_\_\_\_  
(наименование кафедры)

« \_\_\_\_ » \_\_\_\_\_ 20\_\_ г.

Заведующий кафедрой \_\_\_\_\_ (подпись) \_\_\_\_\_ (Ф.И.О.)