


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

«Программирование и основы алгоритмизации»

направление подготовки
27.03.04 «Управление в технических системах»
(уровень бакалавриата)

направленность (профиль)
«Управление и информатика в технических системах»

Петропавловск-Камчатский
2024

Рабочая программа разработана в соответствии с ФГОС ВО по направлению подготовки 27.03.04 «Управление в технических системах», профиль «Управление и информатика в технических системах», и учебного плана ФГБОУ ВО «КамчатГТУ».

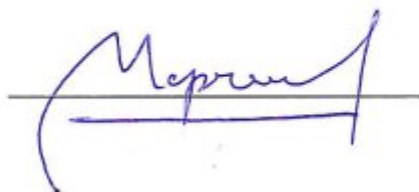
Составитель рабочей программы:

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 изучение теоретических и практических основ проектирования и разработки программного обеспечения ЭВМ.

Задачей дисциплины является дать студенту теоретические и практические знания по принципам проектирования и разработки программного обеспечения ЭВМ, алгоритмическим языкам программирования.

В результате изучения дисциплины:

- студент должен знать этапы разработки программы на ЭВМ, принципы проектирования программного обеспечения, лексические и синтаксические основы языка программирования C++, статические и динамические (списки, очереди, стеки, бинарные деревья) структуры данных, принципы объектно-ориентированного программирования, отечественные и мировые стандарты на разработку программного обеспечения.

- студент должен уметь разрабатывать алгоритмы решения прикладных задач, проектировать оптимальную для данной задачи архитектуру программы, составлять программу решения прикладной задачи на языке программирования C++, составлять программный код программы с объектно-ориентированной структурой, разрабатывать приложения для операционных систем семейства MS Windows в среде разработчика Microsoft Visual C++.

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

2. ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Дисциплина «Программирование и основы алгоритмизации» направлена для освоения следующих компетенций основной образовательной программы подготовки бакалавра по направлению подготовки 27.03.04 "Управление в технических системах" федерального государственного образовательного стандарта высшего образования:

- способностью разрабатывать (на основе действующих стандартов) техническую документацию (в том числе в электронном виде) для регламентного обслуживания систем и средств контроля, автоматизации и управления (ОПК-10).

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

Таблица 1- 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
ОПК-10	Обладать способностью разрабатывать (на основе действующих стандартов) техническую документацию (в том числе в электронном виде) для регламентного обслуживания систем и средств контроля,	ИД-1 _{ОПК-10} : Знает основные требования к проведению регламентного обслуживания систем и средств контроля, автоматизации и управления	Знать: <input checked="" type="checkbox"/> основы проектирования и разработки систем и средств контроля, автоматизации и управления (на основе действующих стандартов);	3(ОПК-10)1
			<input checked="" type="checkbox"/> язык программирования C++;	3(ОПК-10)2
			<input checked="" type="checkbox"/> принципы объектно-ориентированного программирования.	3(ОПК-10)3

Тема 1: Общие основы алгоритмизации и программирования	26	14	10	4	-	12	Опрос, ПЗ,	
Тема 2: Синтаксис языка программирования высокого уровня С++	58	38	12	12	14	20	Опрос, ПЗ, РЗ	
Тема 3: Основы объектно-ориентированного программирования	50	20	14	2	4	30	Опрос, ПЗ, РЗ	
Экзамен	36	-	-	-	-	-	-	36
Всего	180	72	36	18	18	72		36

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

4.1.2. 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			
Тема 1: Общие основы алгоритмизации и программирования	32	2	4	-	-	40	Опрос, ПЗ,	
Тема 2: Синтаксис языка программирования высокого уровня С++	66	6	4	2	2	60	Опрос, ПЗ, РЗ	
Тема 3: Основы объектно-ориентированного программирования	46	6	2	2	2	53	Опрос, ПЗ, РЗ	
Экзамен	9	-	-	-	-	-	-	9
Всего	180	18	10	4	4	153		9

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

4.2. Описание содержания дисциплины

Тема 1: Общие основы алгоритмизации и 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 **Этапы разработки и решение задач на ЭВМ.** Формулировка и постановка задачи. Разработка алгоритма. Виды алгоритмов. Принципы программирования. Ввод и отладка программ. Решение задачи на ЭВМ и анализ полученных результатов.

Практическое занятие 1.1 **Принципы разработки алгоритма программы.** Особенности разработки структуры программ, использующих основные формы алгоритмов.

Примерные задания:

1. Изучить линейную форму алгоритма, составить алгоритм конкретной задачи, использующей линейную форму;
2. Изучить ветвящуюся форму алгоритма, составить алгоритм конкретной задачи, использующей ветвящуюся форму;
3. Изучить циклическую форму алгоритма, составить алгоритм конкретной задачи, использующей циклическую форму.

Лекция 1.3 **Арифметические и логические основы программирования.** Системы счисления, перевод из одной системы счисления в другую. Выполнение арифметических операций в различных системах счисления. Формы представления чисел в ЭВМ. Элементарные функции алгебры логики и особенности их использования в программном коде.

Практическое занятие 1.2 **Системы счисления. Логические основы программирования.** Основные типы систем счисления. Взаимосвязь десятичной, двоичной, восьмеричной и шестнадцатеричной систем счисления. Правила перевода чисел из одной системы счисления в другую. Основные функции булевой алгебры. Обозначение функций булевой алгебры. Основные законы булевой алгебры.

Примерные задания:

1. Перевести число из десятичной системы счисления в двоичную и обратно;
2. Перевести число из шестнадцатеричной системы счисления в восьмеричную и двоичную;
3. Посчитать логическое ИЛИ между двумя числами в двоичном представлении;
4. Посчитать логическое И между двумя числами в двоичном представлении;
5. Посчитать логическое исключающее ИЛИ между двумя числами в двоичном представлении.

Лекция 1.4 **Проектирование программного обеспечения.** Основные фазы проектирования. Структура жизненного цикла программного обеспечения. Модели жизненного цикла. Особенности, достоинства и недостатки каскадной модели. Особенности, достоинства и недостатки спиральной модели. Критерии оценки качества программного обеспечения.

Лекция 1.5 **Стандарты на изготовление программного обеспечения.** Основные мировые и отечественные стандарты. Отечественный стандарт ГОСТ 19 ЕСПД, основные положения, особенности, достоинства, недостатки. ГОСТ 34, основные положения, особенности, достоинства, недостатки. Международный стандарт ISO/IEC 12207, основные положения, особенности, достоинства, недостатки.

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

1. Этапы разработки и решение задач на ЭВМ [1,5,6].
2. Арифметические и логические основы программирования [1,6].
3. Проектирование программного обеспечения [5].
4. Стандарты на изготовление программного обеспечения [1].

Тема 2: Синтаксис языка программирования высокого уровня C++

Лекция 2.1 **Базовые элементы языка C++.** История возникновения языка C++. Общий синтаксис. Разделители. Использование комментариев. Структура идентификаторов. Резервированные слова. Препроцессор. Основные типы данных. Размер данных. Описания. Определение локальных и глобальных переменных.

Практическое занятие 2.1 **Структура программы.** Состав основных файлов программы. Включение препроцессора. Главная функция программы. Принципы разработка элементарной программы на языке C++ в среде Visual C++.

Примерные задания:

1. Изучить включение в код программы элементов препроцессора;
2. Изучить включение в код программы целых переменных;
3. Изучить включение в код программы вещественных переменных;

4. Изучить включение в код программы символьных переменных.

Лабораторная работа 2.1 **Изучение основных элементов управления среды разработчика. Разработка первого проекта.** Основные элементы управления проектом. Состав программы. Включение препроцессора. Использование функции main. Расчет элементарного математического выражения. Вывод решения 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.

Лабораторная работа 2.2 **Разработка программы, осуществляющей ввод информации с клавиатуры и вывод на экран.** Разработка программ, позволяющих вводить с клавиатуры информацию, далее производящих различные математические вычисления и выводящих на экран результаты вычислений.

Цель работы: Изучение принципов разработки программы, осуществляющей ввод данных с клавиатуры и вывод на экран.

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 **Использование операторов языка C++.** Виды операторов языка программирования C++. Операторы присваивания. Операторы проверки условия. Операторы цикла. Другие виды операторов.

Практическое занятие 2.2 **Использование операторов.** Принципы разработки программ, включающей операторы цикла, присваивания, проверки условия.

Примерные задания:

1. Изучить включение в код программы операторов присваивания;
2. Изучить включение в код программы операторов проверки условия;
3. Изучить включение в код программы операторов цикла;
4. Изучить включение в код программы оператора-переключателя.

Лабораторная работа 2.3 **Разработка программы, использующей операторы проверки условия и циклов.** Ввод исходных данных с клавиатуры. Разработка основных элементов программы, включающих операторы. Вывод ответа на экран.

Цель работы: Изучение принципов разработки программы, использующей операторы языка C++.

Работа выполняется в соответствии с методическими указаниями по выполнению лабораторных работ. Лабораторные работы №3-5.

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 2.3 **Указатели, массивы и функции.** Указатели и адреса. Описание массива. Массивы в качестве параметров. Строки. Массивы указателей. Определение функции. Вызов функции. Использование и определение параметров. Передача значений через список параметров.

Практическое занятие 2.2 **Указатели и массивы.** Изучение принципов разработки программ, использующих адресную структуру. Поиск элемента по адресу. Обращение

через указатели. Включение массивов данных в программу. Одномерные и многомерные массивы. Обращению к массиву с помощью указателя.

Примерные задания:

1. Изучить включение в код программы указателей;
2. Изучить включение в код программы массива;
3. Изучить включение в код программы многомерного массива;
4. Изучить включение в код программы массива указателей.

Лабораторная работа 2.4 **Разработка программы, рассчитывающей элементы массива.**

Разработка программы, позволяющей задать с клавиатуры количество элементов массива, ввести с клавиатуры элементы массива, а далее вывести на экран количество ненулевых элементов массива и их среднее арифметическое.

Цель работы: Изучение принципов разработки программы, использующей массивы.

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 2.4 **Основные библиотеки и функции.** Математические функции. Функции работы с данными с плавающей точкой. Функции обработки сигналов. Функции работы с символами и строками. Функции проверки ошибок. Функции дат и времени. Функции общего назначения.

Практическое занятие 2.3 **Использование функций в программе на языке C++.** Использование функций в программе. Внешнее описание функций. Параметры, передаваемые функциям. Использование функции main.

Примерные задания:

1. Изучить принципы создания внешних функций;
2. Изучить принципы создания прототипов функций;
3. Изучить типы возвращаемых значений функции main.

Лабораторная работа 2.5 **Разработка программы, использующей внешние функции.** Разработать внешнюю функцию, осуществляющую расчет заданного математического выражения. Вставить функцию в текст программы. Запросить у пользователя входные данные, произвести расчет и вывести полученный результат на экран.

Цель работы: Изучение принципов разработки программы, позволяющей использовать внешние функции, созданные студентом.

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.

Лекция 2.5 **Потоки. Ввод-вывод данных.** Стандартные библиотеки ввода-вывода. Функции ввода-вывода. Описание параметров функций. Ввод символов с клавиатуры. Ввод-вывод в файл, доступ к файлам. Форматы записи данных. Потокосные классы языка C++. Стандартный ввод-вывод языка C++. Инициализация потоков. Ввод-вывод в файл.

Практическое занятие 2.4 **Функции ввода-вывода.** Использование функций ввода-вывода. Форматы ввода-вывода. Ввод-вывод в файл.

Примерные задания:

1. Изучить включение в код программы возможности по стандартному вводу-выводу;
2. Изучить включение в код программы возможности по вводу-выводу в файл;
3. Изучить включение в код программы возможности по вводу-выводу в строку.

Лабораторная работа 2.6 **Разработка программы, осуществляющей запись/чтение данных из файла.** Разработка программы, позволяющей рассчитать математическое выражение, записать полученный ответ в различных форматах в файлы на диск. Далее данные считываются из файла и выводятся на экран.

Цель работы: Изучение принципов разработки программы, позволяющей производить запись информации в файл.

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 2.6 **Выделение памяти.** Стандартные библиотеки выделения памяти. Функции выделения и освобождения памяти. Операторы выделения и освобождения памяти языка C++.

Практическое занятие 2.6 **Функции динамического выделения памяти.** Функции выделения памяти. Освобождение памяти. Выделение памяти под массив.

Примерные задания:

1. Изучить принципы динамического выделения памяти при помощи функций;
2. Изучить принципы динамического выделения памяти при помощи операторов C++;
3. Изучить принципы освобождения ранее выделенной области памяти.

Лабораторная работа 2.7 **Разработка программы, использующей технологии динамического выделения памяти.** Разработка программы, позволяющей задать с клавиатуры количество элементов массива, ввести с клавиатуры элементы массива, а далее вывести на экран количество ненулевых элементов массива и их среднее арифметическое. Память под массив выделять динамически средствами языка C++.

Цель работы: Изучение принципов разработки программы, позволяющей производить динамическое выделение памяти.

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.

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

1. Базовые элементы языка C++ [1,3,6,7,8,9].
2. Использование операторов языка C++ [1,3,6,7,8,9].
3. Указатели, массивы и функции [1,3,6,7,8,9].
4. Основные библиотеки и функции [1,2,9].

Тема 3: Основы объектно-ориентированного программирования

Лекция 3.1 **Структуры и объединения.** Общие положения. Доступ к элементам структуры. Объединения. Битовые поля структур и объединений.

Лекция 3.2 **Динамические структуры данных.** Принципы создание динамических структур на языке C++. Основные виды динамических структур: стеки, очереди, списки, бинарные деревья.

Лабораторная работа 3.1 **Разработка программы, использующей структуры.** Создается структура с заданными полями. Программа запрашивает пользователя значения полей для нескольких объектов и выводит информацию на экран.

Цель работы: Изучение принципов разработки программы с использованием структур.

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

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 3.3 **Принципы объектно-ориентированного программирования.** Главные свойства: абстракция, инкапсуляция, наследование, полиморфизм. Этапы разработки объектно-ориентированной системы.

Лекция 3.4 **Объекты и классы.** Общие положения. Понятие класса. Члены классов. Методы классов. Доступ к элементам класса. Друзья классов. Обработка исключительных ситуаций.

Лекция 3.5 **Конструкторы и деструкторы.** Общие положения. Понятие конструктора и деструктора. Конструкторы по умолчанию. Свойства конструкторов и деструкторов.

Лекция 3.6 **Наследование классов.** Общие положения. Одиночное и множественное наследование. Наследование закрытых и защищенных членов класса. Иерархия классов.

Лекция 3.7 **Шаблоны классов.** Общие положения. принципы создания шаблонов. Контейнерные классы.

Практическое занятие 3.1 **Разработка объектно-ориентированной программы.** Создание программы на основе классов. Использование конструкторов и деструкторов. Методы классов. Дружественные функции классов.

Примерные задания:

1. Изучить принципы создания программы с объектно-ориентированным кодом;
2. Изучить принципы создания конструкторов и деструкторов классов;
3. Изучить принципы наследования классов;
4. Изучить принципы создания шаблонов классов

Лабораторная работа 3.2 **Разработка программы, использующей классы.** Создается класс с заданными полями, методами, конструктором и деструктором. Программа запрашивает пользователя значения полей для нескольких объектов и выводит с помощью методов информацию на экран. Память под поля выделяется в конструкторе.

Цель работы: Изучение принципов разработки объектно-ориентированного кода

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

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 3. Проработка теоретического материала по следующим темам:

1. Структуры и объединения [1,6,8,9,11].
2. Динамические структуры данных [1,8].
3. Принципы объектно-ориентированного программирования [1,8,9,10,11].
4. Объекты и классы [1,8,9,10,11].
5. Конструкторы и деструкторы [1,8,9,10,11].
6. Наследование классов [1,8,9,10,11].
7. Шаблоны классов [1,8,9,10,11].

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 и основы алгоритмизации» 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. История возникновения языка C++. Общий синтаксис языка C++. Разделители. Использование комментариев. Структура идентификаторов.
2. Этапы подготовки программы на языке C++ к исполнению на ПЭВМ.
3. Резервированные слова.
4. Директивы препроцессора.
5. Основные типы данных. Размер данных.
6. Определение переменных. Локальные, статические и др. типы переменных.
7. Арифметические операции. Операции присваивания.
8. Операции отношения. Логические операции. Побитовые операции.
9. Оператор-выражение. Основные виды операторов.
10. Условные операторы.
11. Операторы цикла.
12. Оператор переключения.
13. Указатели и адреса.
14. Описание массива. Массивы в качестве параметров. Строки. Массивы указателей.
15. Определение внешней функции. Вызов функции. Использование и определение параметров. Функция main.
16. Подключение библиотек. Основные библиотеки языка C++.
17. Библиотеки ввода-вывода. Основные функции ввода-вывода.
18. Ввод-вывод в файл, доступ к файлам. Форматы записи данных.
19. Математическая библиотека. Основные функции.
20. Библиотеки для работы с символами и строками. Основные функции.

21. Библиотека общего назначения. Основные функции.
22. Технологии и функции для динамического выделения памяти.
23. Системные библиотеки. Основные функции.
24. Перечисления.
25. Структуры и объединения.
26. Классы: определение, описание, члены класса. Управление доступом к элементам классов, область определения элементов.
27. Конструкторы и деструкторы. Дружественные функции класса.
28. Основы механизма наследования классов.
29. Шаблоны классов.
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.

7.2. Дополнительная литература

2. *Карпов Б., Баранова Т.* C++. Специальный справочник. С.-Пб.: Питер, 2001. – 479 с.
3. *Керниган Б., Ритчи Д.* Язык программирования Си. М.: Финансы и статистика, 1992. – 272 с.
4. *Климова Л.М.* C++. Практическое программирование. М.: КУДИЦ-ОБРАЗ, 2001. – 587 с.
5. *Кнут Д.* Искусство программирования для ЭВМ. Ч.1. М.: Мир, 1976. – 726 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. *Намиот Д.Е.* Основные особенности языка программирования C++. М.: Память, 1991. – 96 с.
8. *Павловская Т.А.* C/C++. Программирование на языке высокого уровня. С.-Пб.: Питер, 2001. – 460 с.
9. *Подбельский В.В.* Язык Си++. М.: Финансы и статистика, 1996 г. – 559 с.
10. *Романов Е.Л.* Практикум по программированию на C++. С.-Пб.: БХВ-Петербург, 2004. – 427 с.
11. *Страуструп Б.* Язык программирования Си++. М.: Радио и связь, 1991. – 352 с.

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. Обратить 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. Программирование и основы алгоритмизации. Методическое пособие к выполнению лабораторных работ по направлению подготовки бакалавров 27.03.04 «Управление в технических системах» по дисциплине «Программирование и основы алгоритмизации» для студентов очной и заочной формы обучения – Петропавловск-Камчатский: КамчатГТУ, 2015. – 69 с. (электронная форма).

10. КУРСОВОЙ ПРОЕКТ (РАБОТА)

По данной дисциплине не предусмотрен

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 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.

