

Федеральное государственное образовательное учреждение
высшего образования
«КАЛИНИНГРАДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ»

В. А. Петрикин

ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ C++

Учебно-методическое пособие по изучению дисциплины
для студентов бакалавриата по направлению подготовки
09.03.01 Информатика и вычислительная техника,
профиль программы
«Промышленная информатика и системы управления»

Калининград
Издательство ФГБОУ ВО «КГТУ»
2025

УДК 681.5

Рецензенты:

кандидат технических наук, заведующий кафедрой цифровых систем и автоматике института цифровых технологий ФГБОУ ВО «Калининградский государственный технический университет» В. И. Устич;

кандидат технических наук, доцент кафедры прикладной информатики института цифровых технологий ФГБОУ ВО «Калининградский государственный технический университет» О. М. Топоркова

Петрикин, В. А.

Программирование на языке C++: учеб.-метод. пособие по изучению дисциплины для студентов бакалавриата направления подготовки 09.03.01 Информатика и вычислительная техника, профиль программы «Промышленная информатика и системы управления» / В. А. Петрикин. – Калининград: Изд-во ФГБОУ ВО «КГТУ», 2025. – 23 с.

В учебно-методическом пособии приведен тематический план по дисциплине и даны методические указания по её самостоятельному изучению, подготовке к практическим занятиям, подготовке и сдаче зачета и экзамена, выполнению самостоятельной работы.

Пособие подготовлено в соответствии с требованиями утвержденной рабочей программы дисциплины направления подготовки 09.03.01 Информатика и вычислительная техника.

Табл. 4, список лит. – 6 наименований

Учебно-методическое пособие рассмотрено и одобрено в качестве локального электронного методического материала кафедрой цифровых систем и автоматике 27 ноября 2024 г., протокол № 3

Учебно-методическое пособие по изучению дисциплины рекомендовано к использованию в качестве локального электронного методического материала в учебном процессе методической комиссией института цифровых технологий ФГБОУ ВО «Калининградский государственный технический университет» 21 января 2025 г., протокол № 1

УДК 681.5

© Федеральное государственное бюджетное образовательное учреждение высшего образования «Калининградский государственный технический университет», 2025 г.
© Петрикин В. А., 2025 г.

ОГЛАВЛЕНИЕ

1 Введение.....	4
2 Тематический план.....	5
3 Содержание дисциплины.....	6
4 Методические указания по проведению лабораторных занятий.....	10
5 Методические указания по выполнению самостоятельной работы.....	14
6 Методические указания по проведению занятий и освоению дисциплины.....	15
7 Требования к аттестации по дисциплине.....	16
7.1 Текущая аттестация.....	16
7.2 Промежуточная аттестация по дисциплине.....	17
8 Заключение.....	21
9 Библиографический список.....	21

1 ВВЕДЕНИЕ

Данное учебно-методическое пособие предназначено для студентов направления подготовки 09.03.01 Информатика и вычислительная техника, изучающих дисциплину «Программирование на языке C++».

Целью освоения дисциплины является формирование у студентов знаний, умений и навыков, необходимых для профессиональной деятельности в области алгоритмизации и программирования на языке высокого уровня C++.

Задачи изучения дисциплины: приобретение студентом способности создавать инструментальные средства программирования и разрабатывать программное обеспечение систем управления.

В результате изучения дисциплины студент должен:

знать:

- основные принципы программирования на языке C++;
- типы данных и операции над ними;
- структуры данных (списки, деревья, графы);
- принципы объектно-ориентированного программирования;
- принципы работы с файлами;
- стандартную библиотеку шаблонов (STL);

уметь:

- разрабатывать программы на языке C++;
- работать с типами данных и операциями над ними;
- использовать структуры данных (списки, деревья, графы);
- применять принципы объектно-ориентированного программирования;
- работать с файлами;
- использовать стандартную библиотеку шаблонов (STL);

владеть:

- навыками написания программ на языке C++;
- навыками работы с типами данных и операциями над ними;
- навыками использования структур данных (списки, деревья, графы);
- навыками применения принципов объектно-ориентированного программирования;
- навыками работы с файлами;
- навыками использования стандартной библиотеки шаблонов (STL).

Дисциплина «Программирование на языке C++» входит в состав профессионального модуля части, формируемой участниками образовательных отношений образовательной

программы бакалавриата по направлению подготовки 09.03.01 Информатика и вычислительная техника.

Для успешного освоения дисциплины, в соответствии с учебным планом, ей предшествуют такие дисциплины, как: «Информатика и основы программирования», «Математические основы вычислительной техники и программирования», «Программирование на языках высокого уровня».

Результаты освоения дисциплины могут быть использованы при изучении последующих дисциплин учебного плана, выполнении выпускной квалификационной работы, а также в дальнейшей профессиональной деятельности.

Далее в пособии представлен тематический план, содержащий перечень изучаемых тем и лабораторных занятий, мероприятий текущей аттестации и отводимое на них аудиторное время (занятия в соответствии с расписанием) и самостоятельную работу. При формировании личного образовательного плана на семестр следует оценивать рекомендуемое время на изучение дисциплины, возможно потребуется больше времени на выполнение отдельных заданий или проработку отдельных тем.

В разделе «Содержание дисциплины» приведены подробные сведения об изучаемых вопросах, по которым можно ориентироваться в случае пропуска каких-то занятий, а также методические рекомендации для самостоятельной подготовки, каждая тема имеет ссылки на литературу (или иные информационные ресурсы), а также контрольные вопросы для самопроверки.

Раздел «Требования к аттестации по дисциплине» содержит описание обязательных мероприятий контроля самостоятельной работы и усвоения разделов или отдельных тем дисциплины. Далее изложены требования к завершающей аттестации – курсовой работе, зачету и экзамену.

Помимо данного пособия, студентам следует использовать материалы, размещенные в соответствующем данной дисциплине разделе ЭИОС, в которые более оперативно вносятся изменения для адаптации дисциплины под конкретную группу.

2 ТЕМАТИЧЕСКИЙ ПЛАН

Общая трудоемкость дисциплины составляет 10 зачетных единиц (ЗЕТ), т. е. 360 академических часов контактной (лекционных и лабораторных занятий, а также контактной работы посредством электронной-информационно-образовательной среды) и самостоятельной работы студента, в том числе связанной с текущей и промежуточной (заключительной) аттестацией по дисциплине.

Формы аттестации по дисциплине:

– очная форма, пятый семестр – зачет; шестой семестр – курсовая работа, экзамен.

Тематический план лекционных занятий для очной формы обучения приведен в таблице 1.

Таблица 1 – Тематический план лекционных занятий

Номер темы	Тема лекционного занятия	Объем учебной работы
		очная форма, ч
Пятый семестр		
Тема 1	Введение в С++	4
Тема 2	Базовые конструкции языка	9
Тема 3	Указатели и массивы	7
Тема 4	Структуры и объединения	7
Тема 5	Файлы и потоки ввода/вывода	5
Итого:		32
Шестой семестр		
Тема 6	Объектно-ориентированное программирование (ООП)	14
Тема 7	Шаблоны и исключения	8
Тема 8	Стандартная библиотека С++ (STL)	12
Тема 9	Продвинутые концепции С++	10
Тема 10	Проектирование и разработка программ	4
Итого:		48
Всего:		80

3 СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

Содержательно структура дисциплины представлена десятью тематическими блоками (темами).

Тема 1. Введение в С++

Перечень изучаемых вопросов:

Цели и задачи дисциплины. Место дисциплины в структуре образовательной программы. Планируемые результаты освоения дисциплины.

История и развитие языка C++.

Синтаксис и структура программы на C++.

Компиляция и выполнение программ.

Интегрированные среды разработки (IDE).

Рекомендуемая литература: [1, введение, гл. 1]; [2, гл. 1]; [3, гл. 10]; [4].

Контрольные вопросы:

1. В чем основное отличие C++ от C?
2. Какие преимущества дает использование C++ перед другими языками программирования?
3. Какие этапы проходит программа от исходного кода до исполняемого файла?
4. Что такое IDE и какие популярные IDE для C++ вы знаете?

Тема 2. Базовые конструкции языка

Перечень изучаемых вопросов:

Переменные и типы данных.

Операторы и выражения.

Условия и циклы.

Функции.

Рекомендуемая литература: [1, гл. 4]; [2, гл. 3]; [3, гл. 3–6]; [4].

Контрольные вопросы:

1. Какие основные типы данных поддерживаются в C++?
2. Как в C++ создаются и инициализируются переменные?
3. Что такое оператор присваивания в C++ и как он работает?
4. Какие виды циклов имеются в C++ и как они используются?
5. Какова общая структура функции в C++ и как она определяется?

Тема 3. Указатели и массивы

Перечень изучаемых вопросов

Указатели.

Массивы.

Строки и символьные массивы.

Рекомендуемая литература: [1, гл. 2]; [2, гл. 1]; [3, гл. 6–8]; [4].

Контрольные вопросы:

1. Что такое указатель в C++ и как он создается?
2. Как работают арифметические операции с указателями?

3. Как создаются и инициализируются массивы в C++?
4. Как происходит передача массива в качестве параметра в функцию?
5. Что такое разыменованное указание и как оно работает?

Тема 4. Структуры и объединения

Перечень изучаемых вопросов:

Структуры.

Объединения.

Битовые поля.

Рекомендуемая литература: [1, гл. 2]; [3, гл. 13]; [4].

Контрольные вопросы:

1. Что такое структура в C++ и как она определяется?
2. В чем отличие между структурой и объединением в C++?
3. Как создается экземпляр структуры в C++ и как к нему обращаются?
4. Какие ограничения накладываются на объединения в C++?
5. Как задать битовые поля в структуре в C++?

Тема 5. Файлы и потоки ввода/вывода

Перечень изучаемых вопросов:

Потоки ввода/вывода.

Работа с файлами.

Рекомендуемая литература: [1, гл. 10]; [2, гл. 10], [3, гл. 10].

Контрольные вопросы:

1. Как открывается файл для чтения в C++?
2. Как записываются данные в файл в C++?
3. Какие классы C++ используются для работы с вводом/выводом?
4. Как закрывается файл после завершения работы с ним?
5. Как обработать ошибку при открытии файла в C++?

Тема 6. Объектно-ориентированное программирование (ООП)

Перечень изучаемых вопросов:

Принципы ООП.

Классы и объекты.

Инкапсуляция.

Наследование.

Полиморфизм.

Рекомендуемая литература: [1, гл. 5, 6, 12]; [2, гл. 4–9]; [3, гл. 14]; [4].

Контрольные вопросы:

1. Что такое класс в C++ и как он определяется?
2. Как создается объект класса в C++ и как он инициализируется?
3. Какие уровни доступа к членам класса существуют в C++?
4. Как работает механизм наследования в C++?
5. Что такое полиморфизм в C++ и как он реализуется?

Тема 7. Шаблоны и исключения

Перечень изучаемых вопросов:

Шаблоны функций и классов.

Исключения.

Рекомендуемая литература: [1, гл. 8, 12]; [2, гл. 6]; [3, гл. 7].

Контрольные вопросы:

1. Что такое шаблон функции в C++ и как он определяется?
2. Как использовать шаблоны функций для обобщенного программирования?
3. Что такое шаблон класса в C++ и как он определяется?
4. Как генерируется исключение в C++ с помощью оператора throw?
5. Как перехватываются исключения в C++ с помощью блока try-catch?

Тема 8. Стандартная библиотека C++ (STL)

Перечень изучаемых вопросов:

Контейнеры STL.

Алгоритмы STL.

Итераторы.

Рекомендуемая литература: [1, гл. 13]; [2, гл. 12]; [3, гл. 11]; [4].

Контрольные вопросы:

1. Что такое контейнеры STL и какие основные контейнеры вы знаете?
2. Как использовать контейнеры STL для хранения и обработки данных?
3. Какие операции доступны для каждого типа контейнера?
4. Как управлять памятью в контейнерах STL?
5. Какие алгоритмы STL вы знаете и как они используются?

Тема 9. Продвинутое концепции C++

Перечень изучаемых вопросов:

Умные указатели.

Лямбда-функции.

Многопоточное программирование.

Управление памятью.

Рекомендуемая литература: [1, гл. 13]; [3, гл. 13]; [4].

Контрольные вопросы:

1. Что такое умные указатели в C++ и зачем они нужны?
2. Как используются лямбда-функции в C++?
3. Какие средства для работы с потоками предоставляет C++?
4. Как бороться с утечками памяти и находить их причины?
5. Как использовать инструменты для анализа и профилирования использования памяти?

Тема 10. Проектирование и разработка программ

Перечень изучаемых вопросов:

Проектирование программных систем.

Тестирование и отладка.

Рекомендуемая литература: [1, гл. 11]; [3, гл. 2, 12]; [4].

Контрольные вопросы:

1. Что такое умные указатели в C++ и зачем они нужны?
2. Как используются лямбда-функции в C++?
3. Какие средства для работы с потоками предоставляет C++?
4. Как бороться с утечками памяти и находить их причины?
5. Как применять инструменты для анализа и профилирования использования памяти?

4 МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ПРОВЕДЕНИЮ ЛАБОРАТОРНЫХ ЗАНЯТИЙ

Лабораторные работы охватывают широкий спектр тем и позволяют студентам углубленно освоить язык C++ и его возможности.

Тематический план и трудоемкость лабораторных занятий приведены в таблице 2.

Таблица 2 – Объем (трудоемкость освоения) и структура ЛЗ

Номер темы	Содержание лабораторного занятия	Очная форма, ч
Пятый семестр		
1, 2	<p>Лабораторная работа № 1: Основы работы с базовыми конструкциями языка Цель: Изучение базовых конструкций языка C++: переменных, типов данных, операторов, условий и циклов. Задания:</p> <ol style="list-style-type: none"> 1. Напишите программу для вычисления суммы двух чисел. 2. Создайте программу, которая запрашивает у пользователя ввод числа и выводит его квадрат. 3. Реализуйте программу, которая проверяет, является ли введенное число четным или нечетным. 4. Напишите программу, которая считает сумму всех целых чисел от 1 до N (N вводится пользователем) 	4
2, 3	<p>Лабораторная работа № 2: Указатели и динамическая память Цель: Освоение работы с указателями и динамической памятью. Задания:</p> <ol style="list-style-type: none"> 1. Напишите программу, которая создает массив через указатель и заполняет его случайными числами. 2. Реализуйте функцию, которая принимает два указателя на целые числа и меняет их значения местами. 3. Создайте программу, которая считывает массив из файла и выводит его на экран 	4
3	<p>Лабораторная работа № 3: Работа с массивами Цель: Закрепление навыков работы с массивами. Задания:</p> <ol style="list-style-type: none"> 1. Напишите программу, которая находит максимальный элемент массива. 2. Реализуйте сортировку массива методом пузырька. 3. Создайте программу, которая ищет элемент в массиве с помощью бинарного поиска 	4
4	<p>Лабораторная работа № 4: Структуры и объединения Цель: Изучение структур и объединений в C++. Задания:</p> <ol style="list-style-type: none"> 1. Определите структуру для хранения информации о студенте (ФИО, номер группы, средний балл). Заполните массив таких структур данными и выведите их на экран. 2. Реализуйте объединение для хранения либо целого числа, либо строки. Создайте функцию, которая определяет, какой тип данных хранится в объединении, и выводит соответствующее значение 	4
5	<p>Лабораторная работа № 5: Файловый ввод-вывод Цель: Практическое освоение работы с файлами. Задания:</p> <ol style="list-style-type: none"> 1. Напишите программу, которая записывает данные в файл и затем читает их обратно. 2. Реализуйте программу, которая копирует содержимое одного файла в другой. 	4

Номер темы	Содержание лабораторного занятия	Очная форма, ч
	3. Создайте программу, которая удаляет все пробелы из текста в файле	
5	<p>Лабораторная работа № 6: Поток ввода-вывода Цель: Изучение потоков ввода-вывода. Задания:</p> <ol style="list-style-type: none"> 1. Напишите программу, которая использует поток cin для чтения данных от пользователя и выводит их на экран. 2. Реализуйте программу, которая использует поток cout для форматированного вывода данных. 3. Создайте программу, которая читает данные из стандартного ввода и записывает их в стандартный вывод 	4
7	<p>Лабораторная работа № 7: Обработка исключений Цель: Изучение обработки исключительных ситуаций. Задания:</p> <ol style="list-style-type: none"> 1. Напишите программу, которая обрабатывает исключение при делении на ноль. 2. Реализуйте программу, которая пытается открыть несуществующий файл и корректно обрабатывает ошибку. 3. Создайте программу, которая генерирует различные исключения и обрабатывает их соответствующим образом 	4
1-7	<p>Лабораторная работа № 8: Комплексная программа Цель: Интеграция всех изученных тем в одной программе. Задание:</p> <ol style="list-style-type: none"> 1. Создайте программу, которая: <ul style="list-style-type: none"> • Читает данные из файла. • Сортирует их с использованием указателей. • Сохраняет результат в новый файл. • В случае возникновения ошибок корректно обрабатывает исключения 	4
Итого:		32
Шестой семестр		
6	<p>Лабораторная работа № 9: Основы объектно-ориентированного программирования (ООП) Цель: Изучить базовые принципы ООП: классы, объекты, инкапсуляция, наследование, полиморфизм. Задания:</p> <ol style="list-style-type: none"> 1. Создать класс «Студент» с полями: ФИО, номер группы, средний балл. Реализовать методы для установки и получения значений полей. 2. Создать производный класс «Магистр», который наследует класс «Студент». Добавить дополнительные поля и методы. 3. Реализовать виртуальные функции для вывода информации об объекте 	4
7	<p>Лабораторная работа № 10: Шаблоны и исключения Цель: Изучение шаблонов функций и классов, а также обработка исключений.</p>	4

Номер темы	Содержание лабораторного занятия	Очная форма, ч
	Задания: <ol style="list-style-type: none"> 1. Создать шаблонную функцию для нахождения минимального элемента в массиве различных типов данных. 2. Реализовать шаблонный класс стека, поддерживающий работу с различными типами данных. 3. Написать программу, которая генерирует и обрабатывает исключения при работе с файлами 	
8	Лабораторная работа № 11: Стандартная библиотека C++ (STL) Цель: Ознакомиться со стандартной библиотекой C++, изучить контейнеры и алгоритмы STL. Задания: <ol style="list-style-type: none"> 1. Использовать контейнер <code>vector</code> для создания списка студентов и выполнения операций над ним (добавление, удаление, поиск). 2. Применить алгоритм <code>sort</code> для сортировки элементов контейнера. 3. Реализовать программу, использующую ассоциативные контейнеры (<code>map</code>, <code>set</code>) для хранения и обработки данных 	4
9	Лабораторная работа № 12: Продвинутое концепции C++ Цель: Изучение продвинутой концепции языка C++, таких как лямбда-функции, умные указатели, <code>move</code> -семантика. Задания: <ol style="list-style-type: none"> 1. Написать программу, использующую лямбда-функцию для фильтрации элементов вектора. 2. Реализовать класс с поддержкой <code>move</code>-семантики. 3. Использовать умные указатели (<code>unique_ptr</code>, <code>shared_ptr</code>) для управления памятью 	4
10	Лабораторная работа № 13: Проектирование и разработка программ Цель: Научиться проектированию и разработке программ с использованием принципов ООП и паттернов проектирования. Задания: <ol style="list-style-type: none"> 1. Разработать программу для ведения учета товаров в магазине, используя паттерн MVC (Model-View-Controller). 2. Реализовать систему авторизации пользователей с использованием паттерна Singleton. 3. Создать программу для управления базой данных сотрудников компании, применяя паттерн Repository 	4
10	Лабораторная работа № 14: Разработка программы управления микроконтроллером Цель: Закрепление знаний по созданию программ для управления микроконтроллерами. Задания: <ol style="list-style-type: none"> 1. Написать программу для управления светодиодами на плате Arduino. 2. Реализовать программу для считывания данных с датчика температуры и отображения их на экране. 3. Создать программу для управления сервоприводом с использованием микроконтроллера 	4

Номер темы	Содержание лабораторного занятия	Очная форма, ч
10	<p>Лабораторная работа № 15: Многопоточность и параллельное программирование</p> <p>Цель: Изучение многопоточности и параллельного программирования в C++.</p> <p>Задания:</p> <ol style="list-style-type: none"> 1. Реализовать многопоточную программу для вычисления факториала большого числа. 2. Использовать библиотеки thread и mutex для синхронизации доступа к общим ресурсам. 3. Создать программу, которая выполняет несколько задач параллельно и объединяет результаты 	4
10	<p>Лабораторная работа № 16: Комплексная программа</p> <p>Цель: Применение всех изученных концепций для разработки комплексной программы.</p> <p>Задания:</p> <ol style="list-style-type: none"> 1. Разработать программу для управления системой безопасности здания, включающую следующие компоненты: <ul style="list-style-type: none"> • Модуль аутентификации пользователей. • Управление доступом к различным зонам здания. • Слежение за состоянием датчиков движения и температуры. • Логирование событий и оповещение о критических ситуациях. <p>Программа должна использовать ООП, шаблоны, исключения, STL, многопоточность и другие изученные концепции</p>	4
Итого:		32
Всего:		64

Подробная информация о темах лабораторных занятий, включая методические указания по подготовке к ним, теоретические сведения, а также ссылки на литературу, приведены в учебно-методическом пособии.

5 МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

Самостоятельная работа студентов по дисциплине, а также работа в ЭИОС университета может проводиться в компьютерном классе кафедры цифровых систем и автоматики, оснащенном персональными компьютерами с выходом в сеть Интернет.

Самостоятельная работа студента включает в себя освоение теоретического учебного материала (в том числе подготовка к лабораторным занятиям).

6 МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ПРОВЕДЕНИЮ ЗАНЯТИЙ И ОСВОЕНИЮ ДИСЦИПЛИНЫ

Для успешного освоения дисциплины необходимо ознакомиться с основными принципами алгоритмизации и программирования на языке высокого уровня C++, областью его применения в современных технологиях и научных исследованиях.

При разработке образовательной технологии организации учебного процесса основной упор сделан на сочетание активной и интерактивной форм обучения. Интерактивная форма позволяет студентам проявить самостоятельность в освоении теоретического материала и овладении практическими навыками, формирует интерес и позитивную мотивацию к учебе.

При изучении дисциплины внимание студентов постоянно акцентируется не только на теоретическом аспекте изучения языка, но и его практическом применении для создания и развития программного обеспечения.

Для планирования работы студента в начале семестра производится выдача тем для самостоятельного изучения, определяются источники информации и график проведения текущего контроля. В качестве источников информации рекомендуется наряду с учебными пособиями использовать периодические издания (журналы) из области профессиональной деятельности.

В ходе лекционных занятий студенту следует вести конспектирование учебного материала. При самостоятельном изучении заданных преподавателем тем рекомендуется вносить основные материалы по ним в тот же конспект лекций в соответствии с рекомендованным порядком следования учебного материала.

При проведении занятий в интерактивной форме важно участвовать в процессе обсуждения и решения поставленных задач, задавать преподавателю вопросы с целью уяснения теоретических положений, области их применения, разрешения спорных ситуаций.

На лекциях изложению нового материала предшествуют обсуждение предыдущей темы с целью восстановления и закрепления студентами изученного теоретического и практического материала и ответы на вопросы студентов. В конце лекции выделяется время для ответов на вопросы по текущему материалу и его обсуждению. Активность студентов и проявленные знания при обсуждении материала учитываются при текущей и промежуточной (заключительной) аттестации по дисциплине.

Самостоятельная работа студентов проводится с целью:

- систематизации и закрепления полученных теоретических знаний и практических умений студентов;
- углубления и расширения теоретических знаний;
- формирования умений использовать специальную литературу.

7 ТРЕБОВАНИЯ К АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ

7.1 Текущая аттестация

Текущая аттестация (текущий контроль) проводится с целью оценки освоения теоретического учебного материала, в том числе в рамках самостоятельной работы студента (п. 5).

Контроль на лекциях и лабораторных занятиях производится в виде тестирования или устного опроса.

Типовые контрольные вопросы для устного опроса по темам приведены в п. 3 настоящего пособия. Тестовые задания представлены в фонде оценочных средств (приложение к рабочей программе дисциплины).

Положительная оценка («зачтено») по результатам каждого контроля (опроса) выставляется в соответствии с универсальной системой оценивания, приведенной в таблице 3. В случае получения оценки «не зачтено» студент должен пройти повторный контроль по данной теме в ходе последующих консультаций.

Таблица 3 – Система оценок и критерии выставления оценки при прохождении контроля (опроса)

Критерий	Система оценок			
	«не зачтено»	«зачтено»		
1 Системность и полнота знаний в отношении изучаемых объектов	Обладает частичными и разрозненными знаниями, которые не может научно-корректно связывать между собой (только некоторые из которых может связывать между собой)	Обладает минимальным набором знаний, необходимым для системного взгляда на изучаемый объект	Обладает набором знаний, достаточным для системного взгляда на изучаемый объект	Обладает полнотой знаний и системным взглядом на изучаемый объект

Текущий контроль в виде защиты лабораторных работ проводится на лабораторном практикуме, целью которого является формирование знаний и навыков в области программирования на языке C++. Оценка результатов выполнения задания по каждой лабораторной работе производится при представлении студентом отчета по лабораторной работе и на основании ответов студента на вопросы по тематике лабораторной работы. Студент, самостоятельно выполнивший задание, продемонстрировавший знание использованных им технических средств, алгоритмов и языков программирования задачи,

получает по лабораторной работе оценку «зачтено». Если студентом получены неправильные результаты или действия выполнены некорректно, то практическое задание получает оценку «не зачтено» и требует доработки.

7.2 Промежуточная аттестация по дисциплине

Промежуточная аттестация по дисциплине в 5 семестре предусматривает проведение зачета (зачетного тестирования).

Промежуточная аттестация в форме зачета проходит по результатам прохождения всех видов текущего контроля успеваемости. В отдельных случаях (при не прохождении всех видов текущего контроля) зачет может быть проведен в виде тестирования. Тестовые задания для проведения зачетного тестирования приведены в фонде оценочных средств по дисциплине.

Промежуточная (заключительная) аттестация по дисциплине (6 семестр) предусматривает проведение:

- защиты курсовой работы;
- экзамена (экзаменационного тестирования).

Подробная информация о темах курсовых работ, включая методические указания по содержанию и выполнению их, теоретические сведения, а также ссылки на литературу, приведены в учебно-методическом пособии.

По согласованию с преподавателем студент может предложить свою тему курсовой работы.

В ходе выполнения курсовых работ, темы которых предусматривают большой объем теоретического и практического материала, могут использоваться формы обучения: «Работа в команде» и «Поисковый метод». При этом для совместной работы желательно использование систем контроля версий.

Критерии оценки курсовой работы:

– оценка «отлично» ставится, если выполнены все требования к написанию и защите КР: обозначена проблема и обоснована её актуальность, сделан краткий анализ различных точек зрения на рассматриваемую проблему и логично изложена собственная позиция, сформулированы выводы, тема раскрыта полностью, выдержан объём, соблюдены требования к внешнему оформлению, даны правильные ответы на дополнительные вопросы;

– оценка «хорошо» – основные требования к КР и ее защите выполнены, но при этом допущены недочёты. В частности, имеются неточности в изложении материала; отсутствует логическая последовательность в суждениях; не выдержан объём КР; имеются упущения в оформлении; дополнительные вопросы при защите даны неполные ответы;

– оценка «удовлетворительно» – имеются существенные отступления от требований к КР. В частности, тема освещена лишь частично; допущены ошибки в содержании КР или при ответе на дополнительные вопросы; во время защиты отсутствует вывод;

– оценка «неудовлетворительно» – тема КР не раскрыта, обнаруживается существенное непонимание проблемы.

К экзамену допускаются студенты:

– выполнившие и защитившие все предусмотренные лабораторные работы (получившие положительную оценку по результатам лабораторного практикума);

– выполнившие и защитившие курсовую работу.

Экзамен может проводиться как в традиционной форме, так и в виде экзаменационного тестирования.

Экзаменационный билет содержит два экзаменационных вопроса.

Примерный перечень экзаменационных вопросов по дисциплине:

1. Что такое компилятор и его роль в процессе разработки программ?
2. Какие этапы включает процесс компиляции программы на языке C++?
3. Чем отличается препроцессор от компилятора?
4. Основные преимущества использования языка C++ перед другими языками программирования.
5. Синтаксис объявления переменной в C++ и известные типы данных.
6. Принцип работы условных операторов if и switch.
7. Циклы for, while и do-while: определение и примеры использования.
8. Функции в C++: назначение заголовков и тел функций, рекурсия.
9. Указатели: объявление, инициализация, отличие от ссылок.
10. Массивы: объявление, использование, передача в функции, динамическое выделение памяти.
11. Структура в C++: определение, использование, отличия от классов.
12. Объединения: определение и применение.
13. Битовые поля в структуре: определение и использование.
14. Анонимные структуры и объединения: определение и применение.
15. Потoki ввода/вывода в C++: основные классы, чтение и запись файлов.
16. Буферизация ввода/вывода: управление и важность закрытия открытых файлов.
17. Класс в C++: определение и использование.
18. Инкапсуляция, наследование и полиморфизм: определения и примеры использования.
19. Конструкторы и деструкторы: назначение и работа.
20. Перегрузка операторов: определение и пример перегрузки оператора сложения.

21. Абстрактные классы и чистые виртуальные функции: назначение и использование.
22. Дружественные функции и классы: определение и использование.
23. Множественное наследование: особенности и возможные проблемы.
24. Виртуальные функции: механизм позднего связывания.
25. Статические члены класса: определение и использование.
26. Пространства имен: применение в C++.
27. Исключительные ситуации: обработка и стандартные исключения.
28. Кастомные исключения: создание собственного класса исключений.
29. Контейнеры STL: примеры использования основных контейнеров.
30. Умные указатели: предотвращение утечек памяти.
31. Что такое метапрограммирование? Приведите пример использования шаблонов для метапрограммирования.
32. Что такое вариадические шаблоны? Как они расширяют возможности шаблонов?
33. Что такое специализация шаблонов? Как она применяется?
34. Что такое обобщённое программирование? Как оно связано с шаблонами и STL?
35. Что такое паттерны проектирования? Приведите примеры нескольких популярных паттернов.
36. Что такое SOLID-принципы? Как они способствуют улучшению качества кода?
37. Что такое рефакторинг? Как он помогает улучшить существующий код?
38. Что такое модульное тестирование? Как оно применяется в C++?
39. Что такое непрерывная интеграция (CI)? Как она используется в разработке программного обеспечения?
40. Что такое документирование кода? Как оно выполняется в C++?

Тестовые задания для проведения экзаменационного тестирования приведены в фонде оценочных средств по дисциплине.

Система оценивания и критерии выставления оценок по экзамену (экзаменационному тестированию) приведена в таблице 4.

Таблица 4 – Система оценок и критерии выставления оценки по экзамену (экзаменационному тестированию)

Критерий	Система оценок			
	2	3	4	5
	0–40 %	41–60 %	61–80 %	81–100 %
	«неудовлетворительно»	«удовлетворительно»	«хорошо»	«отлично»
1. Системность и полнота знаний в	Обладает частичными и разрозненными	Обладает минимальным набором знаний,	Обладает набором знаний,	Обладает полнотой знаний и

Критерий	Система оценок			
	2	3	4	5
	0–40 %	41–60 %	61–80 %	81–100 %
	«неудовлетворительно»	«удовлетворительно»	«хорошо»	«отлично»
отношении изучаемых объектов	знаниями, которые не может научно-корректно связывать между собой (только некоторые из которых может связывать между собой)	необходимым для системного взгляда на изучаемый объект	достаточным для системного взгляда на изучаемый объект	системным взглядом на изучаемый объект
2. Работа с информацией	Не в состоянии находить необходимую информацию, либо в состоянии находить отдельные фрагменты информации в рамках поставленной задачи	Может найти необходимую информацию в рамках поставленной задачи	Может найти, интерпретировать и систематизировать необходимую информацию в рамках поставленной задачи	Может найти, систематизировать необходимую информацию, а также выявить новые, дополнительные источники информации в рамках поставленной задачи
3. Научное осмысление изучаемого явления, процесса, объекта	Не может делать научно-корректных выводов из имеющихся у него сведений, в состоянии проанализировать только некоторые из имеющихся у него сведений	В состоянии осуществлять научно-корректный анализ предоставленной информации	В состоянии осуществлять систематический и научно-корректный анализ предоставленной информации, вовлекает в исследование новые релевантные задаче данные	В состоянии осуществлять систематический и научно-корректный анализ предоставленной информации, вовлекает в исследование новые релевантные поставленной задаче данные, предлагает новые ракурсы поставленной задачи
4. Освоение стандартных алгоритмов решения профессиональных задач	В состоянии решать только фрагменты поставленной задачи в соответствии с заданным алгоритмом, не освоил предложенный алгоритм, допускает ошибки	В состоянии решать поставленные задачи в соответствии с заданным алгоритмом	В состоянии решать поставленные задачи в соответствии с заданным алгоритмом, понимает основы предложенного алгоритма	Не только владеет алгоритмом и понимает его основы, но и предлагает новые решения в рамках поставленной задачи

8 ЗАКЛЮЧЕНИЕ

Освоение дисциплины «Программирование на языке C++» имеет важное значение для студентов направления Информатика и вычислительная техника, так как этот язык широко используется в самых разных областях, начиная от разработки операционных систем и заканчивая созданием приложений реального времени. Он обеспечивает высокую степень переносимости кода между различными платформами и архитектурами, что делает его незаменимым инструментом для инженеров и программистов. Многие промышленные системы требуют прямого взаимодействия с аппаратурой, такой как датчики, приводы, контроллеры и т. д. C++ позволяет писать код, который обеспечивает точное управление и контроль над физическими процессами, где критически важна способность системы реагировать на события в реальном времени. C++ поддерживает ООП, что особенно полезно в крупных проектах, где важны структурированность и масштабируемость кода. Знание C++ важно для карьерных возможностей в сфере промышленной информатики и систем управления. Этот навык востребован в таких отраслях, как автомобилестроение, энергетика, аэрокосмическая промышленность, робототехника и многие другие. Приобретенные в ходе изучения дисциплины знания, умения и навыки будут углубляться и совершенствоваться в процессе дальнейшего обучения и могут быть применены в профессиональной деятельности.

9 БИБЛИОГРАФИЧЕСКИЙ СПИСОК

Основная литература

1. Страуструп, Б. Язык программирования C++ для профессионалов / Б. Страуструп. – Москва: Национальный Открытый Университет ИНТУИТ, 2024. – 670 с. – ISBN intuit571. – URL: <https://www.ibooks.ru/bookshelf/394598/reading> (дата обращения: 08.06.2024). – Текст: электронный.

Дополнительная литература

2. C/C++. Программирование на языке высокого уровня / Т. А. Павловская. Санкт-Петербург: Питер, 2020. – 464 с: ил. ISBN 978-5-4461-1350-7

3. Кетков, Ю. Л. Введение в языки программирования C и C++: учеб. пособие / Ю. Л. Кетков. – 2-е изд. – Москва: ИНТУИТ, 2016. – 291 с. – Текст: электронный // Лань: электронно-библиотечная система. – URL: <https://e.lanbook.com/book/100719> (дата обращения: 07.06.2024). — Режим доступа: для авториз. пользователей.

4. Кетков, Ю. Л. Введение в языки программирования C и C++ / Ю. Л. Кетков. – Москва: Национальный Открытый Университет ИНТУИТ, 2024. – 291 с. – ISBN intuit144. –

URL: <https://m.ibooks.ru/bookshelf/394176/reading> (дата обращения: 24.12.2024). – Текст: электронный.

Учебно-методические пособия:

5. Лебеде́нко, Л. Ф. Основы программирования на C++: учеб. пособие / Л. Ф. Лебеде́нко, О. И. Моренкова. – 2-е изд., перераб. и доп. – Новосибирск: Сибирский государственный университет телекоммуникаций и информатики, 2021. – 200 с.: ил., табл., схем. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=694769> (дата обращения: 06.06.2024). – Библиогр. в кн. – Текст: электронный.

6. Белоцерковская, И. Е. Алгоритмизация. Введение в язык программирования C++: учеб. пособие / И. Е. Белоцерковская, Н. В. Галина, Л. Ю. Катаева. – 2-е изд., испр. – Москва: Национальный Открытый Университет «ИНТУИТ», 2016. – 197 с.: ил. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=428935> (дата обращения: 06.06.2024). – Текст: электронный.

Интернет-ресурсы

Электронные образовательные ресурсы:

- Научная электронная библиотека eLIBRARY.RU <https://www.elibrary.ru/defaultx.asp>
- ЭБС «Университетская библиотека online» <http://biblioclub.ru/>
- ЭБС «ЛАНЬ» <https://e.lanbook.com>

Локальный электронный методический материал

Виктор Анатольевич Петрикин

ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ C++

Редактор С. Кондрашова
Корректор Т. Звада

Уч.-изд. л. 1,6. Печ. л. 1,4

Издательство федерального государственного бюджетного образовательного
учреждения высшего образования
«Калининградский государственный технический университет».
236022, Калининград, Советский проспект, 1