

Межгосударственное образовательное учреждение высшего образования
«Белорусско-Российский университет»

УТВЕРЖДАЮ

Первый проректор Белорусско-Российского
университета


Ю.В. Машин

20.10 2023

Регистрационный № УД-010304/Б.Р.О.23 /р

ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ

(наименование дисциплины)

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Направление подготовки 01.03.04 Прикладная математика

Направленность (профиль) Разработка программного обеспечения

Квалификация Бакалавр

	Форма обучения
	Очная
Курс	2, 3
Семестр	4, 5
Лекции, часы	50
Лабораторные занятия, часы	84
Контактная работа по учебным занятиям, часы	134
Экзамен, семестр	4, 5
Самостоятельная работа, часы	154
Всего часов / зачетных единиц	288/8

Кафедра-разработчик программы: Программное обеспечение информационных технологий

Составитель: Н.Н. Горбатенко, канд. техн. наук, доцент
(И.О. Фамилия, ученая степень, ученое звание)

Могилев, 2023

Рабочая программа составлена в соответствии с федеральным государственным образовательным стандартом высшего образования - бакалавриат по направлению подготовки 01.03.04 – «Прикладная математика» (уровень бакалавриата), утвержденным приказом № 11 от 10.01.2018. и учебным планом, утвержденным Рег. № 010304-2.1 от 28.04.2023

Рассмотрена и рекомендована к утверждению кафедрой «Программное обеспечение информационных технологий»
«20» 09 2023 г., протокол № 2.

Зав. кафедрой ПОИТ

 В. В. Кутузов

Одобрена и рекомендована к утверждению Научно-методическим советом Белорусско-Российского университета

18.10.2023, протокол № 2.

Зам. председателя
Научно-методического совета

 С.А. Сухоцкий

Рецензент:

Акиншева И. В., заведующий кафедрой программного обеспечения информационных технологий МГУ имени А. А. Кулешова, канд. техн. наук

(И.О. Фамилия, должность, ученая степень, ученое звание рецензента)

Рабочая программа согласована:

Заведующий кафедрой Высшей математики

 В. Г. Замураев

Ведущий библиотекарь

 Е. Н. Киселева

Начальник учебно-методического
отдела

 О.Е. Печковская

1 ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

1.1 Цель учебной дисциплины

Целью учебной дисциплины является обучение студентов объектно-ориентированному программированию, формирование навыков программирования с использованием объектно-ориентированных технологий.

1.2 Планируемые результаты изучения дисциплины

В результате освоения учебной дисциплины студент должен

знать:

- основы объектно-ориентированного подхода к программированию;
- способы реализации отношений между классами;
- использование свойств полиморфизма, наследования и инкапсуляции;
- использование абстрактных классов, интерфейсов и шаблонов.

уметь:

- определять абстракции, модули, строить иерархию классов для реализации программ;
- использовать методы: типизации, инкапсуляции, наследования, полиморфизма для раз-работки программных продуктов;
- использовать возможности стандартных библиотек;
- использовать механизм исключений для создания устойчивых приложений;
- создавать свои и использовать предоставляемые стандартные библиотеки шаблонов сложных структур данных;
- использовать технологию ООП для разработки сложных программ и систем;

владеть:

- методами и инструментальными средствами и системами разработки объектно-ориентированных программ;
- техникой создания объектно-ориентированных программных компонент и организацией их взаимодействия в программных проектах.

1.3 Место учебной дисциплины в системе подготовки студента

Дисциплина относится к блоку 1 «Дисциплины (модули) Обязательная часть блока 1.

Перечень учебных дисциплин, изучаемых ранее, усвоение которых необходимо для изучения данной дисциплины:

- программирование;
- линейная алгебра;
- дискретная математика.

Перечень учебных дисциплин, которые будут опираться на данную дисциплину:

- математическое программирование;
- интеграция программных модулей и компонент;

Кроме того, знания, полученные при изучении дисциплины на лекционных и лабораторных занятиях будут применены при прохождении преддипломной практики, а также при подготовке выпускной квалификационной работы и дальнейшей профессиональной деятельности.

1.4 Требования к освоению учебной дисциплины

Освоение данной учебной дисциплины должно обеспечивать формирование следующих компетенций:

Коды формируемых компетенций	Наименования формируемых компетенций
ОПК-4	Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения
ПК-3	Способен осуществлять непосредственное руководство процессами разработки программного обеспечения

2 СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

Вклад дисциплины в формирование результатов обучения выпускника (компетенций) и достижение обобщенных результатов обучения происходит путём освоения содержания обучения и достижения частных результатов обучения, описанных в данном разделе.

2.1 Содержание учебной дисциплины

Номера тем	Наименование тем	Содержание	Коды формируемых компетенций
1	Введение в язык C++	Структура программы. Функция main(). Директивы препроцессора. Инструкции. Виды инструкций: объявления, выражения, управляющие конструкции. Встроенные функции. Функции пользователя. Комментарии. Ввод и вывод в консоль. Директива using. Подключение пространств имен и определение псевдонимов. Объявления переменных. Фундаментальные типы: int, long, long long, char, float, double, long double, bool. Модификаторы типов. Идентификаторы. Литералы. Неявные и явные преобразования типов. Область видимости переменных. Выражения. Базовые операторы. Арифметические операторы. Операторы сравнения. Побитовые операторы. Логические операторы. Операторы присваивания. Операторы инкремента и декремента. Тернарный оператор. Оператор sizeof(). Оператор запятая. Приоритет операторов. Переопределение порядка выполнения операторов в выражении.	ОПК-4 ПК-3
2	Управляющие инструкции	Инструкция if. Условные выражения. Вложенные инструкции if. Инструкция if-else-if. Инструкция switch. Цикл for. Варианты цикла for. Цикл while. Цикл do-while. Объявление переменных в условных инструкциях и циклах. Инструкции перехода: continue; break; goto. Вложенные циклы.	ОПК-4 ПК-3
3	Компиляция и запуск программы.	Назначение компиляции. Этапы компиляции. Препроцессинг. Ассемблирование. Компилирование. Линковка. Основы препроцессорной обработки. Директивы препроцессора. #include, #define, #if, #ifdef, #ifndef, #else, #elif, #endif, #error, #pragma once. Виды ошибок при запуске программы. Ошибки времени компиляции (Compile-time errors). Ошибки времени выполнения (runtime errors). Неопределенное поведение (undefined behavior).	ОПК-4 ПК-3
4	Массивы. Строки. Перечисления.	Понятие массива. Виды массивов. Одномерные массивы. Объявление. Инициализация. Доступ к	ОПК-4 ПК-3

	Структуры.	<p>элементам массива. Получение длины массива. Перебор элементов массива с помощью цикла <code>for</code>. Ввод/вывод значений массива в консоль.</p> <p>Двумерные массивы. Объявление. Инициализация. Доступ к элементам массива. Перебор элементов массива с помощью цикла <code>for</code>. Ввод/вывод значений массива в консоль.</p> <p>Выход индекса массива за пределы допустимого диапазона.</p> <p>Одномерные массивы символов. Двумерные массивы символов. Перебор двумерных символьных массивов в цикле <code>for</code>. Ввод символьных массивов с консоли.</p> <p>Введение в строки. Недостатки строк в C-стиле с завершающим нулевым символом.</p> <p>Использование класса <code>std::string</code> из стандартной библиотеки <code><string></code> для работы со строками. Создание и инициализация переменных типа <code>string</code>. Получение и изменение символов строки. Доступ к символам строки по индексу. Чтение строки с консоли.</p> <p>Пользовательские типы данных: перечисления, структуры. Объявление и использование перечислений. Объявление и определение структур. Инициализация структур. Присваивание значений членам структур. Массивы структур.</p>	
5	Ссылки и указатели	<p>Понятие ссылки. Формат объявления ссылки. Ссылки на переменные и константы. Область применения ссылок.</p> <p>Понятие указателя. Определение указателя. Получение адреса переменной с помощью оператора <code>&</code>. Инициализация указателя. Нулевой указатель <code>nullptr</code>. Получение значения переменной по ее адресу. Операция разыменованье указателя <code>*</code>.</p> <p>Операции с указателями. Присваивание адреса. Разыменованье указателя. Присвоение указателю другого указателя. Ссылки на указатели. Адрес указателя. Операции сравнения. Приведение типов.</p> <p>Арифметические операции над указателями: сложение, вычитание, инкремент, декремент.</p> <p>Константы и указатели. Указатели на константы. Константный указатель. Константный указатель на константу.</p> <p>Указатели и массивы. Указатель на строки и массивы символов. Массивы указателей.</p>	ОПК-4 ПК-3
6	Память программы	<p>Структура памяти программы на языке C++ в ОЗУ компьютера: область кода программы, область данных (статическая память), область стека (автоматическая память).</p> <p>Область кода программы как место хранения программы в машинных кодах.</p> <p>Статическая память как область памяти, в которой хранится информация на протяжении всего времени работы программы (глобальные переменные, локальные статические переменные).</p> <p>Стековая память как область памяти для хранения локальных переменных, объявленных в функции. Работа стека. Указатель стека. Адрес возврата функции. Размер стека. Переполнение стека.</p> <p>Динамическая память. Выделение памяти. Освобождение памяти. Оператор <code>new</code>. Оператор <code>delete</code>. Выделение и освобождение динамической памяти под простую переменную и массив. Динамические одномерные и многомерные массивы.</p> <p>Ошибки, возникающие при работе с динамической памятью.</p> <p>Использование класса <code>std::vector</code> из стандартной библиотеки <code><vector></code> для реализации динамических массивов. Создание объекта <code>vector</code>. Инициализация вектора. Длина вектора. Емкость вектора. Доступ к элементам вектора. Проход по элементам вектора. Ввод/вывод элементов вектора в консоль.</p>	ОПК-4 ПК-3
7	Функции	<p>Определение функции. Объявление функции. Выполнение функции.</p> <p>Область видимости и время жизни переменных. Глобальные переменные. Локальные переменные. Автоматические объекты. Скрытие переменных. Статические переменные.</p> <p>Параметры функции. Аргументы функции. Аргументы по умолчанию. Автоматическое выведение типа параметров.</p> <p>Передача аргументов по значению и по ссылке. Передача</p>	ОПК-4 ПК-3

		<p>аргументов по значению. Передача параметров по ссылке. Преобразования типов.</p> <p>Константные параметры. Константные ссылки.</p> <p>Оператор return и возвращение результата. return без возвращения значения. Выведение типа результата.</p> <p>Указатели в параметрах функции. Константные параметры-указатели. Параметры по ссылке или параметры-указатели.</p> <p>Массивы в параметрах функции. Передача маркера конца массива. Константные массивы. Передача массива по ссылке. Передача многомерного массива.</p> <p>Возвращение указателей и ссылок. Возвращение указателя. Возвращение ссылки.</p> <p>Перегрузка функций. Перегрузка функций и параметры-ссылки. Перегрузка и параметры-константы.</p> <p>Рекурсивные функции.</p> <p>Указатели на функции. Определение и инициализация указателя. Указатель на функцию с параметрами. Массивы указателей на функции.</p> <p>Указатели на функции как параметры. Тип функции.</p> <p>Указатель на функцию как возвращаемое значение.</p>	
8	Многомодульные программы	<p>Понятие модуля как функционально законченного фрагмента программы, оформленного в виде отдельного файла с исходным кодом, предназначенным для использования в других модулях.</p> <p>Преимущества использования многомодульных программ.</p> <p>Структура файлов многомодульных программ: главный файл, файл заголовков, файл реализации.</p> <p>Главный файл. Файл с расширением .cpp, содержащий метод main()</p> <p>Файл заголовков. Файл с расширением .h, который содержит объявления пользовательских типов.</p> <p>Файл реализации. Файл с расширением .cpp, который содержит определения пользовательских типов объявленных в файле заголовков.</p> <p>Способы защиты от повторного включения заголовочных файлов в программу.</p>	ОПК-4 ПК-3
9	Принципы объектно-ориентированного программирования	<p>Парадигмы программирования: процедурная; объектно-ориентированная; обобщенная; функциональная.</p> <p>Объектно-ориентированное программирование: сущность; отличие от процедурной парадигмы программирования; достоинства; недостатки; область применения.</p> <p>Основные принципы объектно-ориентированного программирования. Инкапсуляция. Наследование. Полиморфизм.</p> <p>Объектная декомпозиция задачи. Абстрагирование.</p>	ОПК-4 ПК-3
10	Классы. Объекты. Инкапсуляция	<p>Понятие класса и объекта класса. Формат объявления класса. Обзор членов класса. Члены-данные (поля класса). Члены-функции (методы класса). Члены-типы (вложенные пользовательские типы). Область видимости класса. Спецификаторы public, private, protected.</p> <p>Создание объекта класса. Указатели на объекты.</p> <p>Действия над объектами классов. Присваивание объекта одного и того же класса (производится почленное копирование членов-данных). Получение адреса объекта с помощью операции &.</p> <p>Передавать объект в качестве параметра в функцию. Возвращать объект в качестве результата работы функции. Осуществлять доступ к элементам объекта с помощью операции '.', а если используется указатель на объект, то с помощью операции '->'. Вызывать методы класса, определяющие поведение объекта.</p> <p>Сущность инкапсуляции. Механизм реализации инкапсуляции. Закрытые поля класса. Открытые интерфейсные методы класса. Геттеры. Сеттеры. Преимущества инкапсуляции.</p> <p>Объявление и определение функций класса. Встроенные функции. Ключевое слово inline. Раздельное объявление и определение функций класса.</p> <p>Скрытый указатель this.</p>	ОПК-4 ПК-3

11	Специальные методы класса	<p>Конструкторы. Виды конструкторов: конструктор умолчания, конструктор копирования, конструктор преобразования.</p> <p>Примеры конструкторов.</p> <p>Деструктор. Идиома программирования RAII.</p> <p>Правила автоматической генерации специальных методов класса: конструктора умолчания, конструктора копирования, деструктора, операторного метода присваивания.</p> <p>Поверхностное копирование. Пример класса с полями встроенного типа.</p> <p>Глубокое копирование. Пример класса с полями указателями. Переопределение операции присваивания.</p> <p>Вызов конструкторов в классах содержащих другие классы в качестве членов-данных. Список инициализации членов класса. Инициализация нестатических членов класса. Делегирующие конструкторы.</p> <p>Порядок вызова конструкторов и деструкторов.</p> <p>Случаи вызова конструктора копирования: явно, в случае инициализации объектом того же типа, в случае инициализации временным объектом.</p> <p>Вызов деструкторов.</p>	ОПК-4 ПК-3
12	Статические члены класса. Дружественные функции и классы	<p>Особенности статических членов-данных и членов функций класса. Описываются в классе с квалификатором <code>static</code>. Существуют в одном экземпляре и доступны для всех объектов данного класса. Существуют независимо от конкретных экземпляров класса, поэтому обращаться к ним можно ещё до размещения в памяти первого объекта этого класса, а также изменять, используя, например, имя константного объекта класса. Необходимо предусмотреть выделение памяти под каждый статический член-данное класса (т.е. описать его вне класса с возможной инициализацией), т.к. при описании самого класса под его экземпляры память под статические члены-данные не выделяется. Доступ к статическим членам класса (наряду с обычным способом) можно осуществлять через имя класса (без указания имени соответствующего экземпляра) и оператор разрешения области видимости «<code>::</code>».</p> <p>Особенности использования статических методов класса. Используются в основном для работы с глобальными объектами или статическими полями данных соответствующего класса. Не могут пользоваться не статическими членами-данными класса. Не могут пользоваться указателем <code>this</code>, Не могут быть виртуальными и константными (<code>inline</code> – могут).</p> <p>Друзья класса. Виды друзей класса: обычная функция, функция-член другого класса, весь класс. Свойства друзей класса: дружба не обладает ни наследуемостью, ни транзитивностью. Примеры использования друзей класса. Преимущества использования друзей класса: эффективность реализации (можно обходить ограничения доступа, предназначенные для обычных пользователей класса), функция-друг нескольких классов позволяет упростить интерфейс этих классов, используется при перегрузке операций.</p>	ОПК-4 ПК-3
13	Перегрузка операций	<p>Понятие перегрузки операций над объектами класса. Операторная функция. Формат объявления операторной функции. Способы перегрузки операций: с помощью метода класса, с помощью внешней функции, функции-друга.</p> <p>Какие операции можно перегружать, а какие нельзя? Перегрузка унарных, бинарных операций, операций отношения и логических операций.</p> <p>Пример перегрузки операции '+' двух комплексных чисел с помощью функции-члена класса и функции друга.</p> <p>Пример перегрузка операции '*' комплексного числа на вещественное число.</p> <p>Перегрузка операции '->'. Перегрузка операции '()'</p>	ОПК-4 ПК-3
14	Связи между объектами	<p>Типы связей между объектами.</p> <p>Композиция. Реализация композиции.</p>	ОПК-4 ПК-3

		<p>Агрегация. Реализация агрегации. Ассоциация. Реализация ассоциации. Зависимость. Реализация зависимости. Различия между композицией, агрегацией и ассоциацией. Контейнерные классы.</p>	
15	Наследование классов	<p>Понятие наследования классов. Терминология. Виды наследования: одиночное наследование, множественное наследование. Одиночное наследование. Формат наследования. Порядок построения дочерних классов. Конструкторы и инициализация. Инициализация членов родительского класса. Цепочки наследований. Деструкторы. Спецификатор доступа protected. Типы наследований. Доступ к членам. Наследование типа private. Наследование типа protected. Добавление нового функционала в дочерний класс. Вызов методов родительского класса. Переопределение методов родительского класса. Скрытие методов родительского класса. Множественное наследование. Проблемы с множественным наследованием. Указатели, ссылки и родительские классы. Выбор между композицией и наследованием. Сравнение композиции и наследования.</p>	ОПК-4 ПК-3
16	Виртуальные функции и полиморфизм	<p>Понятие полиморфизма. Виды полиморфизма: статический, динамический, параметрический. Указатели и ссылки на родительский класс. Использование указателей и ссылок на родительский класс для доступа к объекту дочернего класса. Использование ключевого слова virtual. Понятие виртуальной функции. Наследование виртуальных функций. Недостаток виртуальных функций. Модификатор override. Модификатор final. Ковариантный тип возврата. Виртуальные деструкторы и виртуальное присваивание. Раннее и позднее связывание идентификатора функции с машинным адресом. Виртуальные таблицы. Процесс заполнения виртуальных таблиц. Чистые виртуальные функции. Интерфейсы. Абстрактные функции. Виртуальные базовые классы. Повышающее и понижающее приведение типов в иерархии классов. Повышающее преобразование и использование оператора dynamic_cast. Невозможность конвертации через dynamic_cast. Понижающее приведение и использование оператора static_cast. Оператор dynamic_cast и ссылки. Сравнение оператора dynamic_cast с оператором static_cast. Сравнение понижающего приведения с виртуальными функциями.</p>	ОПК-4 ПК-3
17	Обобщенное программирование. Шаблоны	<p>Понятие обобщенного программирования. Шаблоны функций и классов как средство реализации обобщенного программирования. Шаблоны функций. Создание и использование шаблонов функций. Экземпляры шаблонов функций. Операторы, вызовы функций и шаблоны функций. Шаблоны классов. Создание и использование шаблонов классов. Шаблоны и контейнерные классы. Шаблоны классов и Заголовочные файлы. Параметр non-type в шаблоне. Явная специализация шаблонов функций и шаблонов классов. Полная и частичная специализация шаблонов функций и шаблонов классов. Частичная специализация шаблонов и указатели.</p>	ОПК-4 ПК-3
18	Исключения	<p>Назначение исключений.</p>	ОПК-4

		<p>Обработка исключений. Операторы <code>throw</code>, <code>try</code> и <code>catch</code>. Генерация исключений. Поиск исключений. Обработка исключений. Использование <code>throw</code>, <code>try</code> и <code>catch</code> вместе.</p> <p>Исключения, функции и раскручивание стека. Генерация исключений за пределами блока <code>try</code>.</p> <p>Непойманные исключения и обработчики <code>catch-all</code>. Обработчики всех типов исключений. Использование обработчика <code>catch-all</code> в функции <code>main()</code>. Спецификации исключений.</p> <p>Классы-исключения и наследование. Исключения в перегрузке операторов. Исключения и конструкторы. Классы-исключения. Исключения и наследование. Интерфейсный класс <code>std::exception</code>. Создание собственных классов-исключений, дочерних классу <code>std::exception</code>.</p> <p>Повторная генерация исключений. Генерация нового исключения.</p> <p>Функциональный <code>try</code>-блок.</p> <p>Исключения и спецификатор <code>noexcept</code>.</p> <p>Недостатки и опасности использования исключений.</p>	ПК-3
19	Умные указатели и семантика перемещения	<p>Проблемы использования обычных указателей. Умные указатели. <code>l-values</code> и <code>r-values</code> выражения. Ссылки <code>l-value</code> и <code>r-value</code>. Ссылки <code>r-value</code> в качестве параметров функции. Возврат ссылки <code>r-value</code> из функции.</p> <p>Конструктор копирования и оператор присваивания копированием. Конструктор перемещения и оператор присваивания перемещением. Семантика перемещения. Использование семантики перемещения с <code>l-values</code>. Отключение копирования. Функция <code>std::move()</code>.</p> <p>Умный указатель <code>std::unique_ptr</code>. Доступ к объекту, который хранит умный указатель. Умный указатель <code>std::unique_ptr</code> и динамические массивы. Функция <code>std::make_unique()</code>. Возврат умного указателя <code>std::unique_ptr</code> из функции. Передача умного указателя <code>std::unique_ptr</code> в функцию. Неправильное использование умного указателя <code>std::unique_ptr</code>.</p> <p>Умный указатель <code>std::shared_ptr</code>. Функция <code>std::make_shared()</code>. Детали реализации умного указателя <code>std::shared_ptr</code>. Умный указатель <code>std::shared_ptr</code> и массивы. Опасности использования умного указателя <code>std::shared_ptr</code>.</p>	ОПК-4 ПК-3
20	Стандартная библиотека шаблонов (STL)	<p>Назначение, область применения стандартной библиотеки шаблонов (STL). Состав STL.</p> <p>Контейнеры STL. Последовательные контейнеры. Ассоциативные контейнеры. Адаптеры.</p> <p>Итераторы STL. Функционал итераторов. Итерация по вектору. Итерация по списку. Итерация по <code>set-y</code>. Итерация по ассоциативному массиву.</p> <p>Алгоритмы STL. Алгоритмы <code>min_element()</code> и <code>max_element()</code>. Алгоритмы <code>find()</code> и <code>list::insert()</code>. Алгоритмы <code>sort()</code> и <code>reverse()</code>.</p>	ОПК-4 ПК-3
21	Строковые классы <code>std::string</code> и <code>std::wstring</code>	<p>Создание, уничтожение и конвертация <code>std::string</code>. Длина и ёмкость <code>std::string</code>. Доступ к символам <code>std::string</code>. Конвертация <code>std::string</code> в строки C-style. Присваивание и перестановка значений с <code>std::string</code>. Добавление к <code>std::string</code>. Вставка символов и строк в <code>std::string</code>.</p>	ОПК-4 ПК-3
22	Потоки ввода и вывода	<p>Библиотека <code>iostream</code>. Потоки в C++. Ввод/вывод в C++.</p> <p>Стандартные потоки в C++.</p> <p>Класса <code>istream</code>. Оператор извлечения. Извлечение и пробелы. Специальная версия функции <code>getline()</code> для <code>std::string</code>.</p> <p>Классов <code>ostream</code> и <code>ios</code>. Форматирование вывода. Форматирование вывода. Флаги, манипуляторы и методы форматирования. Точность, запись чисел и десятичная точка. Ширина поля, символы-заполнители и выравнивание.</p> <p>Потоковые классы и строки. Конвертация строк в числа и наоборот. Очистка <code>stringstream</code> для повторного использования.</p> <p>Валидация пользовательского ввода. Строковая валидация. Числовая валидация. Числовая валидация с помощью строки.</p> <p>Базовый файловый ввод и вывод. Файловый вывод. Файловый</p>	ОПК-4 ПК-3

		ввод. Буферизованный вывод. Режимы открытия файлов. Явное открытие файлов с помощью функции open(). Чтение и запись текстовых файлов. Чтение и запись двоичных файлов. Рандомный файловый ввод и вывод. Рандомный доступ к файлам с помощью функций seekg() и seekp(). Одновременное чтение и запись в файл с помощью fstream.	
23	Лямбда-выражения	Назначение лямбда-выражений. Синтаксис лямбда-выражения. Тип лямбда-выражений. Обобщенные лямбды. Обобщенные лямбды и статические переменные. Вывод возвращаемого типа и возвращаемые типы trailing. Функциональные объекты Стандартной библиотеки C++. Понятие лямбда-захвата. Сущность работы лямбда-захватов. Захваты переменных и const. Захват по значению. Захват по ссылке. Захват нескольких переменных. Захваты по умолчанию. Определение новых переменных в лямбда-захвате. Висячие захваченные переменные. Непреднамеренные копии лямбд.	ОПК-4 ПК-3
24	Основные принципы объектно-ориентированного проектирования SOLID	Принцип единственной обязанности (SRP). Принцип открытости/закрытости (OCP). Принцип подстановки Лисков (LSP). Принцип разделения интерфейсов (ISP). Принцип инверсии зависимости (DIP).	ОПК-4 ПК-3
25	Паттерны проектирования GoF	Порождающие паттерны. Поведенческие паттерны. Структурные паттерны	ОПК-4 ПК-3

2.2 Учебно-методическая карта учебной дисциплины

№ недели	Лекции (наименование тем)	Часы	Практические (семинарские) занятия	Часы	Лабораторные занятия	Часы	Самостоятельная работа, часы	Форма контроля знаний	Баллы (max)
Семестр 4									
Модуль 1									
1	Тема 1. Введение в язык C++	2			Л.р.№ 1. Среда программирования Visual Studio. Программирование линейных алгоритмов	2	2		
2	Тема 2. Управляющие инструкции	2			Л.р.№ 2. Программирование разветвляющихся алгоритмов	2	2	ЗЛР	6
3	Тема 3. Компиляция и запуск программы.	2			Л.р.№ 3. Программирование циклических алгоритмов	2	2	ЗЛР	6
4	Тема 4. Массивы. Строки. Перечисления. Структуры	2			Л.р.№ 4. Программирование с использованием одномерных массивов	2	2		
5	Тема 5. Ссылки и указатели	2			Л.р.№ 5. Указатели. Программирование с использованием динамических двумерных массивов	2	2	ЗЛР	6
6	Тема 6. Память программы	2			Л.р.№ 6. Программирование с использованием строк	2	2		
7	Тема 7. Функции	2			Л.р.№ 7. Программирование с использованием структур.	2	2	ЗЛР	6
8	Тема 8. Многомодульные программы	2			Л.р.№ 8. Программирование с использованием функций	2	2	ЗЛР ПКУ	6 30
Модуль 2									

9	Тема 9. Принципы объектно-ориентированного программирования	2		Л.р.№ 9. Создание классов и объектов	2	2	ЗЛР	4
10	Тема 10. Классы. Объекты. Инкапсуляция	2		Л.р.№ 10. Конструкторы и деструкторы	2	2	ЗЛР	4
11	Тема 11. Специальные методы класса	2		Л.р.№ 11. Одиночное наследование	2	2	ЗЛР	4
12	Тема 12. Статические члены класса. Дружественные функции и классы	2		Л.р.№ 12. Статические члены класса	2	2	ЗЛР	4
13	Тема 13. Перегрузка операций	2		Л.р.№ 13. Использование дружественных функций	2	2		
14	Тема 14. Связи между объектами	2		Л.р.№ 14. Перегрузка операций	2	2	ЗЛР	4
15	Тема 15. Наследование классов	2		Л.р.№ 15. Агрегация и композиция классов	2	4	ЗЛР	5
16	Тема 16. Виртуальные функции и полиморфизм	2		Л.р.№ 16. Наследование классов	2	4		
17	Тема 17. Обобщенное программирование. Шаблоны	2		Л.р.№ 17. Полиморфизм и виртуальные методы	2	4	ЗЛР ПКУ	5 30
18-20						36	ПА (экзамен)	40
Итого		34			34	76		100
Семестр 5								
Модуль 1								
1	Тема 18. Исключения	2		Л.р.№ 18. Шаблоны классов	2	2	ЗЛР	6
2				Л.р.№ 18. Шаблоны классов	4	2	ЗЛР	6
3	Тема 19. Умные указатели и семантика перемещения	2		Л.р.№ 19. Обработка исключений	2	2		
4				Л.р.№ 19. Обработка исключений	4	2	ЗЛР	6
5	Тема 20. Стандартная библиотека шаблонов (STL)	2		Л.р.№ 20. Контейнеры	2	2		
6				Л.р.№ 20. Контейнеры	4	2	ЗЛР	6
7	Тема 21. Строковые классы std::string и std::wstring	2		Л.р.№ 21. Алгоритмы	2	2		
8				Л.р.№ 21. Алгоритмы	4	2	ЗЛР ПКУ	6 30
Модуль 2								
9	Тема 22. Поток ввода и вывода	2		Л.р.№ 22. Организация работы с текстовыми файлами	2	2		
10				Л.р.№ 22. Организация работы с бинарными файлами	4	2	ЗЛР	5
11	Тема 23. Лямбда-выражения	2		Л.р.№ 23. Лямбда выражения	2	2		
12				Л.р.№ 24. Программирование с использованием библиотеки STL	4	2	ЗЛР	5

13	Тема 24. Основные принципы объектно-ориентированного проектирования SOLID	2		Л.р.№ 24. Программирование с использованием библиотеки STL	2	2	ЗЛР	5
14				Л.р.№ 25. Принципы SOLID	4	4	ЗЛР	5
15	Тема 25. Паттерны проектирования GoF	2		Л.р.№ 26. Порождающие паттерны	2	4	ЗЛР	5
16				Л.р.№ 27. Структурные паттерны	4	4		
17				Л.р.№ 28. Поведенческие паттерны	2	4	ЗЛР ПКУ	5 30
18-21						36	ПА (экзамен)	40
Итого		16			50	78		100

Принятые обозначения:

Текущий контроль –

ЗЛР – защита лабораторной работы;

ПКУ – промежуточный контроль успеваемости.

ПА - Промежуточная аттестация.

Итоговая оценка определяется как сумма текущего контроля и промежуточной аттестации и соответствует баллам:

Экзамен

Оценка	Отлично	Хорошо	Удовлетворительно	Неудовлетворительно
Баллы	87-100	65-86	51-64	0-50

3 ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

При изучении дисциплины используется модульно-рейтинговая система оценки знаний студентов. Применение форм и методов проведения занятий при изучении различных тем курса представлено в таблице.

№ п/п	Форма проведения занятия	Вид аудиторных занятий			Всего часов
		Лекции	Практические занятия	Лабораторные занятия	
2	Мультимедиа	Темы 1-25			50
7	С использованием ЭВМ			Л.р.№№ 1-28	84
	ИТОГО				134

4 ОЦЕНОЧНЫЕ СРЕДСТВА

№ п/п	Вид оценочных средств	Количество комплектов
1	Вопросы к экзамену	1
2	Экзаменационные билеты	1
3	Задания к защите лабораторных работ	28
4	Вопросы к самостоятельной подготовке и к лабораторным работам	28

5 МЕТОДИКА И КРИТЕРИИ ОЦЕНКИ КОМПЕТЕНЦИЙ СТУДЕНТОВ

5.1 Уровни сформированности компетенций

№ п/п	Уровни сформированности компетенции	Содержательное описание уровня*	Результаты обучения**
<p><i>Компетенция ОПК-4 Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения программ</i></p> <p><i>ИОПК-4.4 Способен формализовать и алгоритмизировать поставленные задачи, применять знание принципов, концепций и языков объектно-ориентированного программирования при разработке компьютерных программ</i></p>			
1	Пороговый уровень	Разработка простых алгоритмов и программ. Знание основ информатики и программирования, проектирования, конструирования и тестирования программных продуктов	Разрабатывает простые алгоритмы и программы. Знает основы информатики и программирования, проектирования, конструирования и тестирования программных продуктов
2	Продвинутый уровень	Разработка сложных алгоритмов и программ. Применение основ информатики и программирования к проектированию, конструированию и тестированию программных продуктов	Разрабатывает сложные алгоритмы и программы. Применяет основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов
3	Высокий уровень	Разработка алгоритмов и программ для решения проблемно-ориентированных задач. Синтез методов проектирования, конструирования и тестирования программных продуктов	Разрабатывает алгоритмы и программы для решения проблемно-ориентированных задач. Синтезирует методы проектирования, конструирования и тестирования программных продуктов
<p><i>Компетенция ПК-3 Способен осуществлять непосредственное руководство процессами разработки программного обеспечения</i></p> <p><i>ИПК-3.3 Способен оценивать качество и эффективность программного кода, принимать решения по изменению программного кода, редактировать программный код при использовании объектно-ориентированного программирования</i></p>			
1	Пороговый уровень	Знание принципов использования операционных систем, сетевых технологий, средств разработки программного интерфейса, применения языков и методов формальных спецификаций, систем управления базами данных для решения поставленных задач.	Знает принципы использования операционных систем, сетевых технологий, средств разработки программного интерфейса, применения языков и методов формальных спецификаций, систем управления базами данных.
2	Продвинутый уровень	Применение принципов использования операционных систем, сетевых технологий,	Применяет принципы использования операционных систем,

		средств разработки программного интерфейса, применения языков и методов формальных спецификаций, систем управления базами данных для решения поставленных задач.	сетевых технологий, средств разработки программного интерфейса, применения языков и методов формальных спецификаций, систем управления базами данных для решения поставленных задач.
3	Высокий уровень	Синтезирует решения на базе принципов использования операционных систем, сетевых технологий, средств разработки программного интерфейса, применения языков и методов формальных спецификаций, систем управления базами данных для решения	Синтез решений на базе принципов использования операционных систем, сетевых технологий, средств разработки программного интерфейса, применения языков и методов формальных спецификаций, систем управления базами данных для решения

5.2 Методика оценки знаний, умений и навыков студентов

Результаты обучения	Оценочные средства*
<i>Компетенция ОПК-4. Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения программ</i>	
<p>Разрабатывает простые алгоритмы и программы. Знает основы информатики и программирования, проектирования, конструирования и тестирования программных продуктов</p> <p>Разрабатывает сложные алгоритмы и программы. Применяет основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов</p> <p>Разрабатывает алгоритмы и программы для решения проблемно-ориентированных задач. Синтезирует методы проектирования, конструирования и тестирования программных продуктов</p>	<p>Вопросы к самостоятельной подготовке к лабораторным работам 1-23.</p> <p>Задания к защите лабораторных работ 1-25.</p>
<i>Компетенция ПК-3 Способен осуществлять непосредственное руководство процессами разработки программного обеспечения</i>	
<p>Знает принципы использования операционных систем, сетевых технологий, средств разработки программного интерфейса, применения языков и методов формальных спецификаций, систем управления базами данных.</p> <p>Применяет принципы использования операционных систем, сетевых технологий, средств</p>	<p>Вопросы к самостоятельной подготовке к лабораторным работам 8, 14, 18, 20-25.</p> <p>Задания к защите лабораторных работ по лабораторным работам 8, 14, 18, 20-25.</p>

<p>разработки программного интерфейса, применения языков и методов формальных спецификаций, систем управления базами данных для решения поставленных задач.</p> <p>Синтез решений на базе принципов использования операционных систем, сетевых технологий, средств разработки программного интерфейса, применения языков и методов формальных спецификаций, систем управления базами данных для решения</p>	
---	--

5.3 Критерии оценки лабораторных работ

Студент обязан самостоятельно в полном объеме выполнить лабораторные работы согласно рабочей программе.

Задание на работы выдает ведущий занятия преподаватель.

По результатам выполнения работ студент обязан оформить отчет по лабораторной работе в соответствии с действующими в Университете требованиями по оформлению отчета.

Отсутствие отчета является причиной недопуска к сдаче лабораторной работы.

За наличие правильно оформленного отчета начисляется 3 балла.

Защита отчета проводится устно, путем ответов на контрольные вопросы к работе, решения задачи по теме лабораторной работы и демонстрации навыков, полученных при выполнении работы.

При защите лабораторной работы студент имеет право пользоваться собственноручно оформленным отчетом.

При отсутствии ответов на заданные преподавателем вопросы отчет не засчитывается и баллы не выставляются.

Правильные ответы оцениваются согласно оценочным уровням сформированности компетенций по изучаемой теме.

Наличие общего представления по изучаемой тематике оценивается в 1-3 балла.

Суммарная оценка за сдаваемую лабораторную работу начисляется в соответствии с представленными критериями.

5.4 Критерии оценки экзамена

5.4.1. К сдаче экзамена допускаются студенты, получив за семестр в сумме не менее 36 баллов. На экзамене студент может набрать от 0 до 40 баллов. Студенты сдают экзамен в комбинированной форме. Количество баллов, набранных студентом, рассчитывается как сумма баллов, полученных за четыре компонента экзамена: письменный ответ на первый теоретический вопрос (от 0 до 10 баллов); письменный ответ на второй теоретический вопрос (от 0 до 10 баллов), решение задачи в письменном виде (от 0 до 10 баллов) и устные ответы на дополнительные вопросы (от 0 до 10 баллов).

5.4.2. Оценка ответа на теоретический вопрос

10 баллов – десять:

- систематизированные, глубокие и полные знания по всем разделам учебной программы, а также по основным вопросам, выходящим за ее пределы;
- точное использование научной терминологии (в том числе – на иностранном языке), стилистически грамотное, логически правильное изложение ответа на вопросы;
- безупречное владение инструментарием учебной дисциплины, умение его эффективно использовать в постановке и решении научных и профессиональных задач;
- выраженная способность самостоятельно и творчески решать сложные проблемы в нестандартной ситуации;

- полное и глубокое усвоение основной и дополнительной литературы, рекомендованной учебной программой дисциплины;
- умение ориентироваться в теориях, концепциях и направлениях по изучаемой дисциплине и давать им критическую оценку, использовать научные достижения других дисциплин.

9 баллов – девять:

- систематизированные, глубокие и полные знания по всем разделам учебной программы;
- точное использование научной терминологии (в том числе на иностранном языке), стилистически грамотное, логически правильное изложение ответа на вопросы;
- владение инструментарием учебной дисциплины, умение его эффективно использовать в постановке и решении научных и профессиональных задач;
- способность самостоятельно и творчески решать сложные проблемы в нестандартной ситуации в рамках учебной программы;
- полное усвоение основной и дополнительной литературы, рекомендованной учебной программой дисциплины;
- умение ориентироваться в основных теориях, концепциях и направлениях по изучаемой дисциплине и давать им критическую оценку.

8 баллов – восемь:

- систематизированные, глубокие и полные знания по всем поставленным вопросам в объеме учебной программы;
- использование научной терминологии, стилистически грамотное, логически правильное изложение ответа на вопросы, умение делать обоснованные выводы;
- владение инструментарием учебной дисциплины (методами комплексного анализа, техникой информационных технологий), умение его использовать в постановке и решении научных и профессиональных задач;
- способность самостоятельно решать сложные проблемы в рамках учебной программы;
- усвоение основной и дополнительной литературы, рекомендованной учебной программой;
- умение ориентироваться в основных теориях, концепциях и направлениях по изучаемой дисциплине и давать им критическую оценку.

7 баллов – семь:

- систематизированные, глубокие и полные знания по всем разделам учебной программы;
- использование научной терминологии (в том числе на иностранном языке), лингвистически и логически правильное изложение ответа на вопросы, умение делать обоснованные выводы;
- владение инструментарием учебной дисциплины, умение его использовать в постановке и решении научных и профессиональных задач;
- усвоение основной и дополнительной литературы, рекомендованной учебной программой дисциплины;
- умение ориентироваться в основных теориях, концепциях и направлениях по изучаемой дисциплине и давать им критическую оценку.

6 баллов – шесть:

- достаточно полные и систематизированные знания в объеме учебной программы;
- использование необходимой научной терминологии, стилистически грамотное, логически правильное изложение ответа на вопросы, умение делать обоснованные выводы;
- владение инструментарием учебной дисциплины, умение его использовать в решении учебных и профессиональных задач;
- способность самостоятельно применять типовые решения в рамках учебной программы;
- усвоение основной литературы, рекомендованной учебной программой дисциплины;
- умение ориентироваться в базовых теориях, концепциях и направлениях по изучаемой дисциплине и давать им сравнительную оценку.

5 баллов – пять:

- достаточные знания в объеме учебной программы;
- использование научной терминологии, стилистически грамотное, логически правильное изложение ответа на вопросы, умение делать выводы;
- владение инструментарием учебной дисциплины, умение его использовать в решении учебных и профессиональных задач;
- способность самостоятельно применять типовые решения в рамках учебной программы;
- усвоение основной литературы, рекомендованной учебной программой дисциплины;
- умение ориентироваться в базовых теориях, концепциях и направлениях по изучаемой дисциплине и давать им сравнительную оценку.

4 балла – четыре, зачтено:

- достаточный объем знаний в рамках образовательного стандарта;
- усвоение основной литературы, рекомендованной учебной программой дисциплины;
- использование научной терминологии, стилистическое и логическое изложение ответа на вопросы, умение делать выводы без существенных ошибок;
- владение инструментарием учебной дисциплины, умение его использовать в решении стандартных (типовых) задач;
- умение под руководством преподавателя решать стандартные (типовые) задачи;
- умение ориентироваться в основных теориях, концепциях и направлениях по изучаемой дисциплине и давать им оценку.

3 балла – три, незачтено:

- недостаточно полный объем знаний в рамках образовательного стандарта;
- знание части основной литературы, рекомендованной учебной программой дисциплины;
- использование научной терминологии, изложение ответа на вопросы с существенными лингвистическими и логическими ошибками;
- слабое владение инструментарием учебной дисциплины, некомпетентность в решении стандартных (типовых) задач;
- неумение ориентироваться в основных теориях, концепциях и направлениях изучаемой дисциплины.

2 балла – два, незачтено:

- фрагментарные знания в рамках образовательного стандарта;
- знания отдельных литературных источников, рекомендованных учебной программой дисциплины;
- неумение использовать научную терминологию дисциплины, наличие в ответе грубых стилистических и логических ошибок.

1 балл – один, незачтено:

– Отсутствие знаний и компетенций в рамках образовательного стандарта или отказ от ответа.

5.4.3. Оценка решения задачи.

На экзамене студент решает одну задачу.

Задача представляет собой задание на разработке нескольких классов и применение к ним действий, указанных в условии задачи.

10 баллов – десять:

Задача решена полностью, с использованием творческого подхода и нестандартных решений. Присутствует валидация данных, обработка исключений.

9 баллов – девять:

Задача решена полностью, с использованием стандартного подхода. Присутствует валидация данных, обработка исключений.

8 баллов – восемь:

Задача решена полностью, с использованием стандартного подхода. Присутствует валидация данных.

7 баллов – семь:

Задача решена полностью, с использованием стандартного подхода.

6 баллов – шесть:

Задача решена не полностью. Отсутствует часть действий над классами.

5 баллов – пять:

Задача решена не полностью. Отсутствует часть действий над классами или классы реализованы не полностью.

4 балла – четыре, зачтено:

Задача решена не полностью. Реализована часть классов и часть действий над ними.

3 балла – три, незачтено:

Задача не компилируется. Реализована часть классов и часть действий над ними.

2 балла – два, незачтено:

Задача не компилируется. Реализована часть классов.

1 балл – один, незачтено:

Задача не компилируется. Реализована небольшая часть классов.

5.4.4. Оценка ответов на дополнительные вопросы.

Количество дополнительных вопросов, задаваемых студенту на экзамене, определяется количеством пропущенных лекций. По каждой пропущенной лекции студенту задается один дополнительный вопрос.

Если количество дополнительных вопросов не превышает десять, разность между десятью баллами и количеством дополнительных вопросов добавляется к баллам, полученным на экзамене. К баллам, полученным на экзамене, добавляются также по одному баллу за каждый правильный ответ на дополнительный вопрос.

Если количество дополнительных вопросов превышает десять, за каждый правильный ответ на дополнительный вопрос студенту добавляется доля, равная отношению десяти к количеству дополнительных вопросов.

По окончании ответов на дополнительные вопросы по просьбе студента ему могут быть задано еще несколько вопросов по курсу (не больше десяти), за правильный ответ на каждый из которых студенту добавляется один балл.

6 МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ОРГАНИЗАЦИИ И ВЫПОЛНЕНИЮ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ ПО УЧЕБНОЙ ДИСЦИПЛИНЕ

Самостоятельная работа студентов (СРС) направлена на закрепление и углубление освоения учебного материала, развитие практических умений. СРС включает следующие виды самостоятельной работы студентов:

Перечень контрольных вопросов и заданий для самостоятельной работы студентов приведен в приложении и хранится на кафедре.

Для СРС рекомендуется использовать источники, приведенные в п. 7.

7 УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

7.1 Основная литература

№ п/п	Библиографическое описание	Гриф	Количество экземпляров
1	Гагарина, Л. Г. Технология разработки программного обеспечения : учеб. пособие / Л.Г. Гагарина, Е.В. Кокорева, Б.Д. Сидорова-Виснадул ; под ред. Л.Г. Гагариной. — Москва : ИД «ФОРУМ» : ИНФРА-М, 2019. — 400 с. — (Высшее образование: Бакалавриат). - ISBN 978-5-8199-0707-8. - Текст : электронный.	Допущено Учебно-методическим объединением вузов по университетскому политехническому образованию в качестве учебного пособия для студентов высших учебных заведений, обучающихся по направлениям подготовки 09.04.01 и 09.03.03 «Информатика и вычислительная техника»	https://znanium.com/catalog/product/1011120
2	Немцова Т.И. Программирование на языке высокого уровня. Программирование на языке С++ : учебное пособие / Т.И. Немцова, С.Ю. Голова, А.И. Терентьев : под ред. Л.Г. Гагариной — Москва: ИД -"ФОРУМ". ИМФРА-М. 2021. - 512с. + Доп. материалы [Электронный ресурс]. - (Среднее профессиональное образование) – Режим доступа:	Рекомендовано Научно-методическим советом Московского государственного института электронной техники (технического университета) в качестве учебного пособия для студентов обучающихся по направлениям подготовки 09.03.01 Информатика и вычислительная техника. 09.03.03 «Прикладная информатика». 09.03.04 «Программная инженерия»	https://znanium.com/catalog/document?id=363426

7.2 Дополнительная литература

№ п/п	Библиографическое описание	Гриф	Количество экземпляров
1.	Мякишев, Д.В. Разработка программного обеспечения АСУ ТП на основе объектно-ориентированного подхода : метод. пособие / Д.В. Мякишев. — Москва ; Вологда : Инфра-Инженерия, 2019. - 128 с. - ISBN 978-5-9729-0305-4. - Текст : электронный. https://znanium.com/catalog/product/1048733	—	https://znanium.com/catalog/product/1048733
2.	Абрамян, М. Э. Введение в стандартную библиотеку шаблонов C++. Описание, примеры использования, учебные задачи : учебник / М. Э. Абрамян ; Южный федеральный университет. — Ростов-на-Дону ; Таганрог : Издательство Южного федерального университета. 2017. — 178 с. - ISBN 978-5-9275-2374-0. - Текст : электронный.	—	https://znanium.com/catalog/product/1020515
3.	Корнеев, В. И. Программирование графики на C++. Теория и примеры : учебное пособие / В. И. Корнеев, Л. Г. Гагарина, М. В. Корнеева. — Москва : ФОРУМ : ИНФРА-М, 2019. — 517 с. — (Высшее образование: Бакалавриат). - ISBN 978-5-8199-0837-2. - Текст : электронный.	Рекомендовано федеральным государственным автономным образовательным учреждением высшего образования «Национальный исследовательский университет "Московский институт электронной техники"» в качестве учебного пособия для студентов, обучающихся по направлению подготовки 09.04.04 «Программная инженерия» и группам направлений 11.03.04 «Электроника и наноэлектроника» и 11.03.02 «Инфокоммуникационные технологии и системы связи»	https://znanium.com/catalog/product/1018909
4.	Кузин, А. В. Программирование на языке Си : учебное пособие / А. В. Кузин, Е. В. Чумакова. — Москва : ФОРУМ : ИНФРА-М, 2019. — 144 с. — (Высшее образование). - ISBN 978-5-00091-066-5. - Текст : электронный.	Рекомендовано в качестве учебного пособия для студентов высших учебных заведений, обучающихся по укрупненной группе специальностей 09.00.00 «Информатика и вычислительная техника» (квалификация (степень) «бакалавр»)	https://znanium.com/catalog/product/1007488

7.3 Перечень ресурсов сети Интернет по изучаемой дисциплине

1. <https://ru.cppreference.com/w/>
2. <https://ravesli.com/uroki-cpp/>
3. <https://metanit.com/cpp/tutorial/>

7.4 Перечень наглядных и других пособий, методических рекомендаций по проведению учебных занятий, а также методических материалов к используемым в образовательном процессе техническим средствам

7.4.1 Методические рекомендации

1 Объектно-ориентированное программирование. Методические рекомендации к лабораторным работам для студентов направления подготовки 01.03.04 «Прикладная математика» очной формы обучения / Сост. Сергиенко О. В., Вайнилович Ю.В., Горбатенко Н.Н. БРУ, Могилев, 2022 г., 38 с.

7.4.2 Информационные технологии

Мультимедийные презентации по лекционному курсу:

Тема 1. Введение в язык C++

Тема 2. Управляющие инструкции

Тема 3. Компиляция и запуск программы.

Тема 4. Массивы. Строки. Перечисления. Структуры

Тема 5. Ссылки и указатели

Тема 6. Память программы

Тема 7. Функции

Тема 8. Многомодульные программы

Тема 9. Принципы объектно-ориентированного программирования

Тема 10. Классы. Объекты. Инкапсуляция

Тема 11. Специальные методы класса

Тема 12. Статические члены класса. Дружественные функции и классы

Тема 13. Перегрузка операций

Тема 14. Связи между объектами

Тема 15. Наследование классов

Тема 16. Виртуальные функции и полиморфизм

Тема 17. Шаблоны

Тема 18. Исключения

Тема 19. Умные указатели

Тема 20. Стандартная библиотека шаблонов (STL)

Тема 21. Строковые классы `std::string` и `std::wstring`

Тема 22. Потоки ввода и вывода

Тема 23. Лямбда-выражения

Тема 24. Основные принципы объектно-ориентированного проектирования SOLID

Тема 25. Паттерны проектирования GoF

7.4.3 Перечень программного обеспечения, используемого в образовательном процессе

Используется свободно распространяемое программное обеспечение

1. Microsoft Visual Studio Community 2019 Бесплатная полнофункциональная интегрированная среда разработки для учащихся, разработчиков открытого ПО и отдельных разработчиков.

2. Visual Studio Code (Free. Built on open source.)

3. Adobe Acrobat Reader

8 МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

Материально-техническое обеспечение дисциплины содержится в паспорте лаборатории « а. 517/2», рег. номер ПУЛ-4/517.2-23 , « а. 518/2». рег. номер ПУЛ-4/518.2-23 , «а. 519/2 », рег. номер ПУЛ-4/519.2-23 .

ДОПОЛНЕНИЯ И ИЗМЕНЕНИЯ К РАБОЧЕЙ ПРОГРАММЕ ДИСЦИПЛИНЫ

по учебной дисциплине Объектно-ориентированное программирование

направление подготовки 01.03.04 Прикладная математика

направленность (профиль) Разработка программного обеспечения

на 2024/2025 учебный год

№№ пп	Дополнения и изменения	Основание
1	Дополнений и изменений нет	

Рабочая программа пересмотрена и одобрена на заседании кафедры

«Программное обеспечение информационных технологий»

(название кафедры-разработчика программы)

(протокол № 8 от 29 02 2024)

Заведующий кафедрой

канд. техн. наук., доцент
(ученая степень, ученое звание)



В.В. Кутузов

УТВЕРЖДАЮ

Декан экономического факультета

(название факультета, выпускающего по данной специальности)

канд. физ.-мат. наук., доцент
(ученая степень, ученое звание)



И.И. Маковецкий

16. 04 2024

СОГЛАСОВАНО:

Зав. кафедрой

«Высшая математика»

(название выпускающей кафедры
данной специальности)



В.Г. Замураев

Ведущий библиотекарь



Е.Н. Киселева

Начальник учебно-
методического
отдела



О.Е. Печковская

16. 04 2024