

Министерство науки и высшего образования
Российской Федерации

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Донецкий государственный университет»

Факультет математики и информационных технологий
Кафедра теории упругости и вычислительной математики имени академика
А.С. Космодамианского



УТВЕРЖДАЮ
проректор

П.А. Машаров

« 29 » марта 2024 г.

МП

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ И СТАНДАРТНАЯ БИБЛИОТЕКА C++ В ЧИСЛЕННЫХ МЕТОДАХ ИССЛЕДОВАНИЯ МОДЕЛЕЙ ДЕФОРМИРОВАНИЯ

Укрупненная группа направлений подготовки	01.00.00 Математика и механика
Программа высшего образования	Программа бакалавриата
Направление подготовки	01.03.02 Прикладная математика и информатика
Профиль подготовки	Прикладная математика и информатика
Квалификация	Бакалавр
Форма обучения	Очная

Рабочая программа адаптирована для лиц
с ограниченными возможностями здоровья и инвалидов

Донецк 2024

Рабочая программа дисциплины **«Объектно-ориентированное программирование и стандартная библиотека C++ в численных методах исследования моделей деформирования»** для обучающихся по направлению подготовки 01.03.02 Прикладная математика и информатика (Профиль: Прикладная математика и информатика), составлена на основании Федерального государственного образовательного стандарта высшего образования – бакалавриат по направлению подготовки 01.03.02 Прикладная математика и информатика, утвержденного приказом Министерства образования и науки Российской Федерации от 10 января 2018 г. № 9 (с изм. и доп.), Порядка организации и осуществления образовательной деятельности по образовательным программам высшего образования – программам бакалавриата, программам специалитета, программам магистратуры, утвержденного приказом Министерства науки и высшего образования Российской Федерации от 06 апреля 2021 г. № 245 (с изм. и доп.), в соответствии с учебным планом, утвержденным Ученым советом ФГБОУ ВО «ДонГУ» для набора 2024 года.

Разработчик:

профессор кафедры теории упругости и
вычислительной математики
им. акад. А.С. Космодамианского,
д-р. физ.-мат. наук, профессор

И.А. Моисеенко

Рабочая программа одобрена на заседании кафедры теории упругости и вычислительной математики имени академика А.С. Космодамианского.
Протокол от 26.03.2024 г. № 10

Врио заведующего кафедрой

Р.Н. Нескородев

СОГЛАСОВАНО:

Декан факультета математики и
информационных технологий
28.03.2024 г.

И.А. Моисеенко

Учебно-методическая комиссия факультета математики и информационных технологий.
Протокол от 27.03.2024 г. № 3.
Председатель

Л. И. Селякова

Руководитель основной профессиональной
образовательной программы,
д-р физ.-мат. наук, доцент
26.03.2024 г.

Р.Н. Нескородев

1. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

1.1. Требования к предварительной подготовке обучающихся, предшествующие и сопутствующие дисциплины, на которых основывается изучение данной:

дисциплины программы бакалавриата:

Основы информатики, Языки и методы программирования, Архитектура компьютеров, Дискретная математика.

1.2. Дисциплины, курсовые работы и практики, для которых освоение данной дисциплины необходимо как предшествующее:

Теория автоматов и формальных языков, Дополнительные главы теории алгоритмов и структур данных в механико-математических моделях, Производственная практика: технологическая (проектно-технологическая) практика (обязательная), Производственная практика: преддипломная практика.

2. ОПИСАНИЕ ДИСЦИПЛИНЫ

2.1. Общая характеристика

Наименование показателя	Значение показателя
Название образовательной программы	01.03.02 Прикладная математика и информатика (Профиль: Прикладная математика и информатика)
Шифр и название в соответствии с учебным планом	Б1.В.ОД.12 Объектно-ориентированное программирование и стандартная библиотека C++ в численных методах исследования моделей деформирования
Часть образовательной программы	Вариативная часть
Количество зачетных единиц / всего часов	5 / 180

2.2. Распределение часов по формам и периодам обучения

Форма обучения	курс	семестр	Общее количество часов					Форма контроля
			лекционных	лабораторных	практических	самостоятельной работы + контроль	всего	
Очная	2	3	34	34	17	95	180	экзамен

3. ЦЕЛИ ДИСЦИПЛИНЫ

Изучение основ классической теории объектно-ориентированного программирования, в том числе: пути эволюции технологий программирования от алгоритмического к объектно-ориентированному; основных принципов объектно-ориентированного построения программных систем (Абстракция, Инкапсуляция, Наследование, Полиморфизм); понятий классов, объектов, взаимоотношений между ними, а также многоуровневой модели OMG. Формирование понимания идеологии и ключевых аспектов объектно-ориентированного программирования на языке C++, достаточного для практического использования в процессе дальнейшего обучения и в профессиональной сфере, получение навыков разработки программ в среде Microsoft Visual Studio. Изучение средств объектно-ориентированного и обобщенного программирования языка C++. Формирование понимания идеологии организации, структуры и методов использования

стандартной библиотеки языка C++ в объеме, достаточном для практического применения в процессе дальнейшего обучения и в профессиональной сфере.

4. КОМПЕТЕНЦИИ ОБУЧАЮЩЕГОСЯ, ФОРМИРУЕМЫЕ В РЕЗУЛЬТАТЕ ОСВОЕНИЯ КОМПОНЕНТА ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ, ИХ ИНДИКАТОРЫ И ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ

4.1. Компетенции

ПК-3. Способен разрабатывать и руководить процессом разработки и модификации компьютерных программ для решения профессиональных и научно-исследовательских задач.

4.2. Индикаторы компетенций

ПК-3.1. Способен с использованием объектно-ориентированного подхода разрабатывать и модифицировать компьютерные программы для решения профессиональных и научно-исследовательских задач.

4.3. Результаты обучения

ПК-3.1.1. Знает основные теоретические понятия объектно-ориентированного подхода к программированию.

ПК-3.1.2. Умеет анализировать предметную область решаемых задач с целью использования объектно-ориентированного подхода для их реализации, разрабатывать объектную модель программы.

ПК-3.1.3. Аргументированно применяет синтаксические и семантические аспекты к реализации объектно-ориентированного подхода в языке программирования C++.

5. ПРОГРАММА ДИСЦИПЛИНЫ

Название темы	Краткое содержание темы (вопросы темы)
Раздел 1. Объектно-ориентированное программирование и стандартная библиотека C++	
1. Объектно-ориентированная парадигма	Краткий обзор основных парадигм программирования. Процессно-ориентированная парадигма и язык С. Краткий обзор ключевых конструкций С. Аппликативная (функциональная) парадигма. Парадигма логического программирования. Объектно-ориентированная парадигма. Методология разработки объектно ориентированного программного обеспечения. Принципы объектно-ориентированного подхода. Этапы объектно-ориентированной методологии.
2. Базовые классы	Понятие класса, объекта, поля, метода. Определение базового класса. Секции доступа. Область видимости класса. Инкапсуляция – объектно-ориентированная характеристика модульности. Внешний интерфейс и внутренняя реализация инкапсулированного программного объекта. Характерные признаки эффективной инкапсуляции: абстракция, общедоступный интерфейс и сокрытие реализации. Доступ к элементам класса. Статические элементы класса. Дружественные классы и функции.

	<p>Пространства имен.</p> <p>Специальные методы класса. Назначение конструкторов и деструктора. Конструктор по умолчанию. Параметризованные конструкторы. Конструктор копирования. Деструктор.</p> <p>Функциональное замыкание как механизм работы специальных методов класса по созданию и уничтожению объектов. Ограничения на использование автоматически сгенерированных компилятором специальных методов.</p>
3. Перегрузка операторов	<p>Статический полиморфизм. Перегруженные функции-члены. Концепция перегрузки операторов. Перегруженные операторы как методы класса и как дружественные функции.</p> <p>Перегрузка операторов преобразования типа. Спецификатор <i>explicit</i> для конструкторов. Перегрузка операторов <i>new</i> и <i>delete</i>.</p>
4. Потоки	<p>Файловые и строковые потоки. Средства ввода-вывода и работа с потоками. Перегрузка операторов потокового ввода/вывода. Библиотека потока C++. Общие функции ввода-вывода в поток.</p>
5. Наследование	<p>Наследование – базовое понятие объектно-ориентированного программирования. Наследование и повторное использование кода. Простое наследование. Синтаксис определения класса-потомка. Правила наследования. Преобразование типов (ссылок и указателей). Правила видимости при простом наследовании. Перекрывание имен и функциональное замыкание в случае простого наследования.</p>
6. Динамический полиморфизм	<p>Динамический полиморфизм (простое наследование). Реализация виртуальных функций. Перегруженные и переопределённые методы. Раннее и позднее связывание. Пустые и чистые виртуальные функции. Абстрактные классы.</p> <p>Множественное наследование. Синтаксис определения класса-потомка. Видимость при множественном наследовании. Виртуальные базовые классы. Функциональное замыкание в случае множественного наследования.</p> <p>Динамический полиморфизм (множественное наследование). Интерфейсы. Абстрактные классы. Динамическая информация о типе при простом и множественном наследовании.</p> <p>Объектно-ориентированная концепция обработки исключений. Блок <i>try</i>. Оператор <i>catch</i>. Классы исключений. Стандартные исключения. Последовательность действий при</p>

	возникновении исключительной ситуации (генерирование и переброска исключений).
7. Параметрический полиморфизм для функций	Параметрический полиморфизм для функций. Синтаксис определения шаблона функции. Механизмы генерирования и идентификации при функциональных обращениях.
8. Параметрический полиморфизм для классов	Синтаксис определения шаблона класса. Параметры шаблона. Специализация шаблонов классов. Реализация шаблона класса.
9. Архитектура библиотеки	Составные части библиотеки. Организация стандартной библиотеки шаблонов. Стандартные контейнеры.
10. Итераторы	Итераторы. Обратные итераторы. Итераторы вставки. Потокные итераторы.
11. Функциональные объекты	Функциональные объекты. Арифметические функциональные объекты. Предикаты. Отрицатели. Связыватели. Адаптеры указателей на функции. Адаптеры методов.
12. Последовательные контейнеры	Последовательные контейнеры. Векторы (vector). Списки (list). Стеки (stack). Двусторонние очереди (deque). Очереди (queue). Очереди с приоритетами (priority queue).
13. Ассоциативные контейнеры	Ассоциативные контейнеры. Множества (set). Множества с дубликатами (multiset). Битовые множества (bitset). Словари (map). Словари с дубликатами (multimap).
14. Операции с последовательностями	Алгоритмы. Немодифицирующие операции с последовательностями. Модифицирующие операции с последовательностями.
15. Специальные алгоритмы	Алгоритмы, связанные с сортировкой. Алгоритмы работы с множествами и пирамидами. Другие средства стандартной библиотеки.

6. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

6.1. Форма обучения – очная, курс – 2, семестр – 4

Наименования разделов и тем	Количество часов				
	Лекц.	Лабор.	Практ.	СРС+К	Всего
Раздел 1. Объектно-ориентированное программирование и стандартная библиотека C++	34	34	17	95	180
1. Объектно-ориентированная парадигма	2	2	1	6	11
2. Базовые классы	2	2	1	6	11
3. Перегрузка операторов	2	2	1	6	11
4. Потoki	2	2	1	6	11
5. Наследование	2	2	1	6	11
6. Динамический полиморфизм	2	2	1	6	11

7. Параметрический полиморфизм для функций	2	2	1	6	11
8. Параметрический полиморфизм для классов	2	2	1	6	11
9. Архитектура библиотеки	2	2	1	6	11
10. Итераторы	2	2	1	6	11
11. Функциональные объекты	2	2	1	6	11
12. Последовательные контейнеры	2	2	1	6	11
13. Ассоциативные контейнеры	2	2	1	6	11
14. Операции с последовательностями	4	4	2	7	17
15. Специальные алгоритмы	4	4	2	10	20
ИТОГО ПО КОМПОНЕНТУ ОПОП	34	34	17	54	144

7. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ (СРЕДСТВА) ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ, ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

7.1. Контрольные вопросы

Раздел 1

1. Процессно-ориентированная парадигма и язык С. Краткий обзор ключевых конструкций С. Аппликативная (функциональная) парадигма. Парадигма логического программирования.

2. Объектно-ориентированная парадигма. Методология разработки объектно-ориентированного программного обеспечения.

3. Принципы объектно-ориентированного подхода. Этапы объектно-ориентированной методологии.

4. Понятие класса, объекта, поля, метода. Определение базового класса. Секции доступа. Область видимости класса. Инкапсуляция – объектно-ориентированная характеристика модульности.

5. Внешний интерфейс и внутренняя реализация инкапсулированного программного объекта.

6. Характерные признаки эффективной инкапсуляции: абстракция, общедоступный интерфейс и сокрытие реализации.

7. Доступ к элементам класса. Статические элементы класса. Дружественные классы и функции. Пространства имен.

8. Специальные методы класса. Назначение конструкторов и деструктора. Конструктор по умолчанию. Параметризованные конструкторы.

9. Конструктор копирования. Деструктор.

10. Функциональное замыкание как механизм работы специальных методов класса по созданию и уничтожению объектов. Ограничения на использование автоматически сгенерированных компилятором специальных методов.

11. Статический полиморфизм. Перегруженные функции-члены. Концепция перегрузки операторов. Перегруженные операторы как методы класса и как дружественные функции.

12. Перегрузка операторов преобразования типа. Спецификатор *explicit* для конструкторов. Перегрузка операторов *new* и *delete*.

13. Файловые и строковые потоки. Средства ввода-вывода и работа с потоками. Перегрузка операторов потокового ввода/вывода. Библиотека потока C++. Общие функции ввода-вывода в поток.

14. Наследование – базовое понятие объектно-ориентированного программирования. Наследование и повторное использование кода. Простое наследование. Синтаксис определения класса-потомка. Правила наследования.

15. Преобразование типов (ссылок и указателей). Правила видимости при простом наследовании. Перекрытие имен и функциональное замыкание в случае простого наследования.

16. Динамический полиморфизм (простое наследование). Реализация виртуальных функций. Перегруженные и переопределённые методы.

17. Раннее и позднее связывание. Пустые и чистые виртуальные функции. Абстрактные классы.

18. Множественное наследование. Синтаксис определения класса-потомка. Видимость при множественном наследовании. Виртуальные базовые классы. Функциональное замыкание в случае множественного наследования.

19. Динамический полиморфизм (множественное наследование). Интерфейсы. Абстрактные классы.

20. Динамическая информация о типе при простом и множественном наследовании.

21. Объектно-ориентированная концепция обработки исключений. Блок *try*. Оператор *catch*. Классы исключений. Стандартные исключения.

22. Последовательность действий при возникновении исключительной ситуации (генерирование и переброска исключений).

23. Параметрический полиморфизм для функций. Синтаксис определения шаблона функции. Механизмы генерирования и идентификации при функциональных обращениях.

24. Синтаксис определения шаблона класса. Параметры шаблона. Специализация шаблонов классов. Реализация шаблона класса.

25. Составные части библиотеки. Организация стандартной библиотеки шаблонов. Стандартные контейнеры.

26. Итераторы. Обратные итераторы. Итераторы вставки. Поточные итераторы.

27. Функциональные объекты. Арифметические функциональные объекты. Предикаты. Отрицатели.

28. Связыватели. Адаптеры указателей на функции. Адаптеры методов.

29. Последовательные контейнеры. Векторы (*vector*). Списки (*list*). Стеки (*stack*). Двусторонние очереди (*deque*).

30. Очереди (*queue*). Очереди с приоритетами (*priority_queue*).

31. Ассоциативные контейнеры. Множества (*set*). Множества с дубликатами (*multiset*). Битовые множества (*bitset*). Словари (*map*). Словари с дубликатами (*multimap*).

32. Алгоритмы. Немодифицирующие операции с последовательностями. Модифицирующие операции с последовательностями.

33. Алгоритмы, связанные с сортировкой. Алгоритмы работы с множествами и пирамидами.

7.2. Темы письменных работ (типы задач)

Контрольные работы по практике:

1. В чем состоит концепция полиморфизма? Как виртуальные функции помогают реализовать динамический полиморфизм?

2. Механизм реализации динамического полиморфизма. Опишите и сравните механизмы «раннего связывания» и «позднего связывания». Примеры.

3. Статические и нестатические члены класса, в чем различие? Особенности доступа к ним? Особенности создания и размещения в памяти статических и нестатических членов данных класса? Чем отличается интерфейс статического и нестатического метода класса? Примеры.

4. Определить класс *PointOnPlane* (точка на декартовой плоскости) и с его использованием класс геометрических фигур *QuadrangleOnPlane* – четырехугольник на декартовой плоскости (четырёхугольник задается координатами четырех его вершин в порядке против часовой стрелки) в котором определить методы: необходимые конструкторы (с проверкой корректности геометрической фигуры) и деструктор;

преобразование в тип double (трактуются как вычисление периметра); перегруженные операции сравнения == и != (проверяется совпадение размеров и расположения), сложение *четырёхугольника* и *точки на декартовой плоскости* (трактуются как смещение *четырёхугольника* на заданный *точкой* вектор).

Контрольная работа по проверке теоретических знаний – по всем темам, с использованием указанных выше контрольных вопросов.

7.3. Образец содержания экзаменационного билета

1. Статические и нестатические члены класса, в чем различие? Особенности доступа к ним? Особенности создания и размещения в памяти статических и нестатических членов данных класса? Чем отличается интерфейс статического и нестатического метода класса? Примеры.

2. Определить класс PointOnPlane (*точка на декартовой плоскости*) и с его использованием класс геометрических фигур QuadrangleOnPlane – *четырёхугольник на декартовой плоскости* (четырёхугольник задается координатами четырех его вершин в порядке против часовой стрелки) в котором определить методы: необходимые конструкторы (с проверкой корректности геометрической фигуры) и деструктор; преобразование в тип double (трактуются как вычисление периметра); перегруженные операции сравнения == и != (проверяется совпадение размеров и расположения), сложение *четырёхугольника* и *точки на декартовой плоскости* (трактуются как смещение *четырёхугольника* на заданный *точкой* вектор).

В случае ведения учебного процесса с использованием электронного обучения и дистанционных образовательных технологий, содержание билета может отличаться от приведенного.

8. РАСПРЕДЕЛЕНИЕ БАЛЛОВ, КОТОРЫЕ ПОЛУЧАЮТ ОБУЧАЮЩИЕСЯ

Общая оценка знаний обучающихся по дисциплине проводится по 100-балльной шкале исходя из максимума, приведенного в таблице ниже. Организационно-учебная работа в аудитории оценивается на основе таких критериев как посещаемость занятий, своевременное и качественное выполнение домашних заданий, активность во время проведения лекционных, лабораторных и практических занятий (участие в обсуждении текущего и пройденного материала, решение задач и т.п.).

Номера разделов	Виды работ	Максимальное количество баллов
1	Организационно-учебная работа студента в аудитории	5
	Самостоятельная работа и лабораторные работы	70
	Контрольная работа	25
ИТОГО		100
Общий итог за семестр		100

Соответствие баллов оценке

Количество баллов из 100	ECTS	Оценка по пятибалльной шкале	
		Экзамен, дифференцированный зачет	Зачет
90-100	A	отлично	зачтено
80-89	B	хорошо	зачтено

75-79	C	удовлетворительно	зачтено
70-74	D		зачтено
60-69	E		зачтено
35-59	FX	неудовлетворительно	не зачтено
0-34	F		не зачтено

9. ОБЕСПЕЧЕНИЕ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ДЛЯ ЛИЦ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ И ИНВАЛИДОВ

В ходе реализации дисциплины используются следующие дополнительные методы обучения, текущего контроля успеваемости и промежуточной аттестации обучающихся в зависимости от их индивидуальных особенностей:

- 1) для слепых и слабовидящих:
 - лекции оформляются в виде электронного документа, доступного с помощью компьютера со специализированным программным обеспечением;
 - для выполнения задания при необходимости предоставляется увеличивающее устройство; возможно также использование собственных увеличивающих устройств;
 - письменные задания оформляются увеличенным шрифтом.
- 2) для глухих и слабослышащих:
 - лекции оформляются в виде электронного документа;
 - письменные задания выполняются на компьютере в письменной форме;
 - экзамен проводится в письменной форме на компьютере; возможно проведение в форме тестирования.
- 3) для лиц с нарушениями опорно-двигательного аппарата:
 - лекции оформляются в виде электронного документа, доступного с помощью компьютера со специализированным программным обеспечением;
 - письменные задания выполняются на компьютере;
 - экзамен и зачёт проводятся в устной форме или выполняются в письменной форме на компьютере.

При необходимости предусматривается увеличение времени для подготовки ответа.

Процедура проведения промежуточной аттестации для обучающихся устанавливается с учётом их индивидуальных психофизических особенностей. Промежуточная аттестация может проводиться в несколько этапов.

Проведение процедуры оценивания результатов обучения допускается с использованием дистанционных образовательных технологий.

Обеспечивается доступ к информационным и библиографическим ресурсам в сети Интернет для каждого обучающегося в формах, адаптированных к ограничениям их здоровья и восприятия информации:

- 1) для слепых и слабовидящих:
 - в печатной форме увеличенным шрифтом;
 - в форме электронного документа;
- 2) для глухих и слабослышащих:
 - в печатной форме;
 - в форме электронного документа.
- 3) для обучающихся с нарушениями опорно-двигательного аппарата:
 - в печатной форме;
 - в форме электронного документа.

10. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ УЧЕБНОГО ПРОЦЕССА

Учебные занятия проводятся в Главном корпусе ДонГУ (г. Донецк, пр. Гурова, 6), в Учебно-практическом вычислительном центре ДонГУ (г. Донецк, пр. Гурова, 6, корпус 12).

Для проведения лекций требуется аудитория, оборудованная меловой или маркерной доской / сенсорным экраном / мультимедийный проектор с экраном и ноутбуком, комплект учебной мебели для студентов, рабочее место преподавателя.

Для проведения практических занятий требуется аудитория, оборудованная меловой или маркерной доской / сенсорным экраном / мультимедийный проектор с экраном и ноутбук, комплект учебной мебели для студентов, рабочее место преподавателя.

Для проведения лабораторных занятий требуется аудитория, оборудованная маркерной доской или сенсорным экраном / мультимедийный проектор с экраном и ноутбук, персональные компьютеры, комплект учебной мебели для студентов, рабочее место преподавателя, выход в Интернет – проводной или с использованием Wi-Fi.

Для самостоятельной работы используются текстовые и электронные ресурсы Научной библиотеки университета и других электронных библиотечных баз данных, учебно-методическое обеспечение, представленное в аудиториях Главного корпуса (ауд. 511, 605, 610).

Обучающиеся имеют возможность использовать учебные материалы по дисциплине, размещенные на платформе Moodle Центра дистанционного образования ФГБОУ ВО «ДонГУ». При изучении дисциплины применяются электронное обучение и дистанционные образовательные технологии.

С использованием ресурсов платформы дистанционного образования осуществляется текущий контроль знаний обучающихся на основе тестирования и проверки результатов самостоятельной работы.

11. РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

11.1. Основная литература

1. Буч Гради. Объектно-ориентированный анализ и проектирование с примерами приложений на C++ / Гради Буч ; Пер. с англ. под ред. И. Романовского, Ф. Андреева. - 2-е изд. - М. : БИНОМ ; СПб. : Невский диалект, 1999. – 560 с.
2. Иванова Г.С.. Объектно-ориентированное программирование : Учеб. для студентов вузов, обучающихся по направлению подготовки дипломированных специалистов "Информатика и вычислительная техника" / Г.С. Иванова, Т.Н. Ничушкина, Е.К. Пугачев; Под ред. Г.С. Ивановой. - М. : Изд-во МГТУ им. Н. Э. Баумана, 2001. - 317 с.
3. Калоеров С. А. Программирование на языке C++ : учеб. пособие / С. А. Калоеров ; Донецкий нац. ун-т. - Изд. 3-е. - Донецк : Юго-Восток, 2009. - 298 с.
4. Лафоре Р. Объектно-ориентированное программирование в C++ [Текст] / Р. Лафоре; [пер. с англ. А. Кузнецова]. - 4-е изд. - М. [и др.] : Питер, 2008. - 923 с.
5. Павловская Т. А. C/C++. Программирование на языке высокого уровня : учеб. для студентов вузов, обучающихся по направлению "Информатика и вычислит. техника" / Т. А. Павловская. - Москва [и др.] : Питер, 2010. - 460 с.

11.2. Дополнительная литература

6. Пол Айра. Объектно-ориентированное программирование на C++ / Айра Пол; Пер. с англ. Д. Ковальчука. - 2-е изд. - М. : БИНОМ ; СПб. : Невский диалект, 1999. - 464 с.
7. Страуструп Б. Язык программирования C++ / Бьерн Страуструп ; пер. с англ. С. Анисимова, М. Кононова ; под ред. Ф. Андреева, А. Ушакова. - спец. изд. - М. : Бином-Пресс, 2008. - 1098 с.
8. Кнут Д. Э. Искусство программирования: Пер. с англ. Т. 1 : Основные алгоритмы / Д. Э. Кнут ; Под общ. ред. Ю. В. Козаченко. - 3. изд. - М. : Вильямс ; СПб., 2001. - 714 с.

9. Кнут Д. Э. Искусство программирования : Пер. с англ. Т. 3 : Сортировка и поиск / Д. Э. Кнут ; Под общ. ред. Ю. В. Козаченко. - 2. изд. - М. : Вильямс ; СПб., 2001. - 824 с.
Кнут Д. Э. Искусство программирования для ЭВМ : [В 7 т.] : Пер. с англ. Т. 3 : Сортировка и поиск / Пер. с англ. Н. И. Вьюковой и др. ; Под ред. Ю. М. Баяковского. - М. : Мир, 1978. - 844 с.

12. ИНФОРМАЦИОННЫЕ РЕСУРСЫ

1. **Национальная электронная библиотека (НЭБ):** федеральная государственная информационная система / Министерство Культуры РФ; Российская государственная библиотека. – Москва, 2019- . – URL: <https://rusneb.ru/> (дата обращения: 01.09.2023). – Режим доступа: свободный, подписка. Необходима установка программного обеспечения. – Текст: электронный.
2. **eLIBRARY.RU:** научная электронная библиотека: сайт. – Москва, 2000- . – URL: <https://elibrary.ru> (дата обращения: 01.09.2023). – Режим доступа: для авторизов. пользователей. –Текст: электронный.
3. Научная электронная библиотека **«КиберЛенинка»:** сайт / Ассоциация «Открытая наука». – Москва, 2014- . – URL: <https://cyberleninka.ru/>. – Режим доступа: свободный. – Текст: электронный.
4. Электронно-библиотечная система **«Лань»:** [сайт]. – URL: <https://e.lanbook.com> (дата обращения: 01.09.2023). – Режим доступа: для авторизов. пользователей. – Текст: электронный.
5. **ЭБС Юрайт:** электронная библиотечная система: сайт. – Москва, 2013. – URL: <https://biblio-online.ru> (дата обращения: 01.09.2023). – Режим доступа: для авторизов. пользователей. – Текст: электронный.
6. **Электронно-библиотечная система ДонГУ:** сайт / ФГБОУ ВО «ДонГУ». – Донецк, 2016- . – URL: <http://library.donnu.ru/> (дата обращения: 01.09.2023). – Режим доступа: свободный. – Текст: электронный.
7. **Электронный каталог** Научной библиотеки ДонГУ: раздел сайта / НБ ДонГУ. – Текст: электронный // ЭБС ДонГУ: сайт. – URL: <http://library.donnu.ru/catalog/> (дата обращения: 01.09.2023). – Режим доступа: поиск свободный, электронные документы – для пользователей ДонГУ.
8. **Электронный архив ДонГУ:** раздел сайта / НБ ДонГУ. – Текст: электронный // ЭБС ДонГУ: сайт. – URL: <http://repo.donnu.ru/> (дата обращения: 01.09.2023). – Режим доступа: свободный.

13. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

1. Windows 7 PRO (корпоративная лицензия ДонГУ № 46484614)
2. Microsoft Office (корпоративная лицензия ДонГУ № 46472919)
3. Microsoft Visual Studio (лицензия программы Dream Spark для высших учебных заведений)
4. Антивирус Касперского, Adobe Acrobat Reader, xPDF (лицензии GPL, Apache, BSD для свободного программного обеспечения).