

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



П.А. Машаров
2024 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ

Укрупненная группа направлений подготовки	27.00.00 Управление в технических системах
Программа высшего образования	Программа бакалавриата
Направление подготовки	27.03.03 Системный анализ и управление
Профиль подготовки	Системный анализ и управление
Квалификация	Бакалавр
Форма обучения	Очная

Рабочая программа адаптирована для лиц
с ограниченными возможностями здоровья и инвалидов

Рабочая программа дисциплины **«Объектно-ориентированное программирование»** для обучающихся по направлению подготовки 27.03.03 Системный анализ и управление (Профиль: Системный анализ и управление), составлена на основании Федерального государственного образовательного стандарта высшего образования – бакалавриат по направлению подготовки 27.03.03 Системный анализ и управление, утвержденного приказом Министерства образования и науки Российской Федерации от 07 августа 2020 г. № 902 (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:

доцент кафедры теории упругости
и вычислительной математики
им. акад. А.С. Космодамианского,
канд. физ.-мат. наук



А.И. Мирончук

Рабочая программа одобрена на заседании кафедры теории упругости и вычислительной математики им. акад. 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,
канд. экон. наук, доц.
26.03.2024 г.



А.М. Гизатулин

1. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

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

предметы программы среднего общего образования: Математика, Алгебра и начала математического анализа, Геометрия, Информатика;

дисциплины программы бакалавриата: Офисные прикладные программы: лабораторный практикум, Информационные технологии, Основы программирования на Python, Базы данных, Технологии и методы программирования.

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

Проектирование и архитектура программных систем, Управление проектами и автоматизацией, Тестирование и отладка программного обеспечения, Информационная безопасность и защита информации, Web-программирование.

2. ОПИСАНИЕ ДИСЦИПЛИНЫ

2.1. Общая характеристика

Наименование показателя	Значение показателя
Название образовательной программы	27.03.03 Системный анализ и управление (Профиль: Системный анализ и управление)
Шифр и название в соответствии с учебным планом	Б1.Б.М4.7 Объектно-ориентированное программирование
Часть образовательной программы	Базовая часть
Количество зачетных единиц / всего часов	5,5 / 198

2.2. Распределение часов по периодам обучения

Форма обучения	курс	семестр	Общее количество часов					Форма контроля
			лекционных	лабораторных	практических	самостоятельной работы + контроль	всего	
Очная	2	3	34	–	34	130	198	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++.

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

4.1. Компетенции

ОПК-6. Способен разрабатывать методы моделирования, анализа и технологии синтеза процессов и систем, а также алгоритмы и программы, основанные на этих методах, пригодные для практического применения в области техники и технологии

4.2. Индикаторы компетенций

ОПК-6.1. Способен применять языки, системы и инструментальные средства программирования в профессиональной деятельности

4.3. Результаты обучения

ОПК-1.3.1. Знает язык программирования высокого уровня; методы и приемы программирования задач в профессиональной деятельности.

ОПК-1.3.2. Умеет работать в среде программирования, которая поддерживает изучаемый язык; настраивать инструментальные средства программирования языка высокого уровня для наиболее удобного для себя интерфейса.

ОПК-1.3.3. Владеет основными средствами и методами разработки алгоритмов; основными приемами объектного программирования на языке C++.

5. ПРОГРАММА ДИСЦИПЛИНЫ

Название темы	Краткое содержание темы (вопросы темы)
Раздел 1. Инкапсуляция.	
1. Объектно-ориентированная парадигма	1.1. Краткий обзор основных парадигм программирования. 1.2. Процессно-ориентированная парадигма и язык С. 1.3. Краткий обзор ключевых конструкций С. 1.4. Аппликативная (функциональная) парадигма. 1.5. Парадигма логического программирования. 1.6. Объектно-ориентированная парадигма. 1.7. Методология разработки объектно-ориентированного программного обеспечения. 1.8. Принципы объектно-ориентированного подхода. 1.9. Этапы объектно-ориентированной методологии.
2. Базовые классы	2.1. Понятие класса, объекта, поля, метода. 2.2. Определение базового класса. 2.3. Секции доступа. Область видимости класса. 2.4. Инкапсуляция – объектно-ориентированная характеристика модульности. 2.5. Внешний интерфейс и внутренняя реализация инкапсулированного программного объекта. 2.6. Характерные признаки эффективной инкапсуляции: абстракция, общедоступный интерфейс и сокрытие реализации. 2.7. Доступ к элементам класса. Статические элементы класса. Дружественные классы и

	<p>функции. Пространства имен.</p> <p>2.8. Специальные методы класса. Назначение конструкторов и деструктора. Конструктор по умолчанию. Параметризованные конструкторы. Конструктор копирования. Деструктор.</p> <p>2.9. Функциональное замыкание как механизм работы специальных методов класса по созданию и уничтожению объектов. Ограничения на использование автоматически сгенерированных компилятором специальных методов.</p>
Раздел 2. Статический полиморфизм.	
3. Перегрузка операторов	<p>3.1. Статический полиморфизм. Перегруженные функции-члены. Концепция перегрузки операторов. Перегруженные операторы как методы класса и как дружественные функции.</p> <p>3.2. Перегрузка операторов преобразования типа. Спецификатор <i>explicit</i> для конструкторов. Перегрузка операторов <i>new</i> и <i>delete</i>.</p>
4. Потоки	<p>4.1. Файловые и строковые потоки.</p> <p>4.2. Средства ввода-вывода и работа с потоками.</p> <p>4.3. Перегрузка операторов потокового ввода/вывода.</p> <p>4.4. Библиотека потока C++. Общие функции ввода-вывода в поток.</p>
Раздел 3. Наследование и динамический полиморфизм.	
5. Наследование	<p>5.1. Наследование – базовое понятие объектно-ориентированного программирования. Наследование и повторное использование кода. Простое наследование. Синтаксис определения класса-потомка. Правила наследования.</p> <p>5.2. Преобразование типов (ссылок и указателей). Правила видимости при простом наследовании. Перекрытие имен и функциональное замыкание в случае простого наследования.</p>
6. Динамический полиморфизм	<p>6.1. Динамический полиморфизм (простое наследование). Реализация виртуальных функций. Перегруженные и переопределённые методы. Раннее и позднее связывание. Пустые и чистые виртуальные функции. Абстрактные классы.</p> <p>6.2. Множественное наследование. Синтаксис определения класса-потомка. Видимость при множественном наследовании. Виртуальные базовые классы. Функциональное замыкание в случае множественного наследования.</p> <p>6.3. Динамический полиморфизм (множественное наследование). Интерфейсы. Абстрактные классы. Динамическая информация о типе при простом и множественном наследовании.</p> <p>6.4. Объектно-ориентированная концепция обработки исключений. Блок <i>try</i>. Оператор <i>catch</i>. Классы исключений. Стандартные исключения. Последовательность действий при возникновении</p>

	исключительной ситуации (генерирование и переброска исключений).
--	--

6. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

6.1. Форма обучения – очная, курс – 2, семестр – 3

Наименования разделов и тем	Количество часов				
	Лекц.	Лабор.	Практ.	СРС	Всего
Раздел 1. Инкапсуляция.	10	-	10	42	62
1. Объектно-ориентированная парадигма	5	-	5	21	31
2. Базовые классы	5	-	5	21	31
Раздел 2. Статический полиморфизм.	12	-	12	44	68
3. Перегрузка операторов	6	-	6	22	34
4. Потoki	6	-	6	22	34
Раздел 3. Наследование и динамический полиморфизм.	12	-	12	44	68
5. Наследование	6	-	6	22	34
6. Динамический полиморфизм	6	-	6	22	34
ПО КОМПОНЕНТУ ОПОП	34	-	34	94	198

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

7.1. Контрольные вопросы

Раздел 1

1. Краткий обзор основных парадигм программирования, процессно-ориентированная парадигма и язык C, краткий обзор ключевых конструкций C, аппликативная (функциональная) парадигма, парадигма логического программирования, объектно-ориентированная парадигма, методология разработки объектно-ориентированного программного обеспечения, принципы объектно-ориентированного подхода, этапы объектно-ориентированной методологии.

2. 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, пространства имен, специальные методы класса, назначение конструкторов и деструктора, конструктор по умолчанию, параметризованные конструкторы, конструктор копирования, деструктор, функциональное замыкание как механизм работы специальных методов класса по созданию и уничтожению объектов, ограничения на использование автоматически сгенерированных компилятором специальных методов.

Раздел 2

3. Статический полиморфизм, перегруженные функции-члены, концепция перегрузки операторов, перегруженные операторы как методы класса и как дружественные функции, перегрузка операторов преобразования типа, спецификатор

explicit для конструкторов, перегрузка операторов *new* и *delete*.

4. 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.

Раздел 3

5. 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, перекрытие имен и функциональное замыкание в случае простого наследования, динамический полиморфизм (простое наследование), реализация виртуальных функций, перерегруженные и переопределённые методы, раннее и позднее связывание, пустые и чистые виртуальные функции, абстрактные классы, множественное наследование, синтаксис определения класса-потомка, видимость при множественном наследовании, виртуальные базовые классы, функциональное замыкание в случае множественного наследования.

6. Динамический полиморфизм (множественное наследование), интерфейсы, абстрактные классы, динамическая информация о типе при простом и множественном наследовании, объектно-ориентированная концепция обработки исключений, блок *try*, оператор *catch*, классы исключений, стандартные исключения, последовательность действий при возникновении исключительной ситуации (генерирование и переброска исключений).

7.2. Темы письменных работ (типы задач)

Контрольные работы по практике темам:

1. 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).

2. Создать шаблон функции, возвращающей значение минимального элемента массива. Аргументами функции должны быть имя и размер массива. Продемонстрировать работу шаблона на примере массива экземпляров созданного класса – треугольник, задаваемый координатами трех своих вершин, определив для него операции сравнения по площади: $S = \sqrt{p(p-a)(p-b)(p-c)}$, где $p = \frac{1}{2}(a+b+c)$.

7.3. Образец содержания экзаменационного билета (при наличии экзамена по дисциплине)

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 1

1. Какова очередность вызова и выполнения (тела) конструкторов/деструкторов при простом/множественном наследовании? А если присутствуют виртуальные классы? Как управлять передачей информации между конструкторами классов, расположенных на разных уровнях дерева наследования (синтаксис)? **Примеры.**

2. В чем состоит идея использования виртуальных классов? Чем отличается дерево наследования в случае наличия/отсутствия в нем виртуальных классов? Какова очередность вызова и выполнения (тела) конструкторов/деструкторов при наследовании в случае наличия/отсутствия виртуальных классов? Как разрешаются конфликты имен при наследовании в случае наличия виртуальных классов (правило доминирования)?

Примеры.

3. Создать шаблон функции, возвращающей среднее арифметическое значение элемента массива. Аргументами функции должны быть имя и размер массива. Продемонстрировать работу шаблона на примере массива экземпляров созданного класса – вектор на плоскости с координатами x , y , определив для него операции сложения и деления на число.

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 и практических занятий (участие в обсуждении текущего и пройденного материала, решение задач и т.п.).

Номера разделов	Виды работ	Максимальное количество баллов
1-3	Организационно-учебная работа в аудитории	10
	Самостоятельная работа	15
	Контрольные работы по практике	15
	Контрольная работа по теоретическому материалу	20
ИТОГО		60
Экзамен		40
Общий итог за семестр		100

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). Для проведения практических занятий требуется аудитория, оборудованная меловой или маркерной доской, мультимедийный проектор и экран, ноутбук, 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 учебно-методическом кабинете Главного корпуса (ауд.405).

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. – Москва: БИНОМ; Санкт-Петербург: 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. – Москва: Изд-во МГТУ им. Н. Э. Баумана, 2001. – 317 с.

3. Калоеров, С.А. Программирование на языке C++ : Учеб. пособие / С.А. Калоеров ; Донец. нац. ун-т. – Донецк : Юго-Восток, 2002. – 224 с.

4. Калоеров, С.А. Программирование на языке C++ : Учеб. пособие / С.А. Калоеров ; Донец. нац. ун-т. – 2-е изд. – Донецк : Юго-Восток, 2004. – 237 с.

5. 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.

6. Лафоре, Р. Объектно-ориентированное программирование в C++ [Текст] / Р. Лафоре; [пер. с англ. А. Кузнецова]. – 4-е изд. – Москва [и др.]:Питер, 2008. – 923 с.

7. Леоненко, А.В. Объектно-ориентированный анализ и проектирование с использованием UML и IBM Rational Rose : учеб. пособие / А. В. Леоненков. – Москва: Интернет-ун-т информ. технологий : Бином. Лаб. знаний, 2006. – 320 с.

8. Павловская, Т.А. C/C++ Программирование на языке высокого уровня : учебник для вузов по направлению "Информатика и вычисл. техника" / Т. А. Павловская. – Москва и др.:Питер, 2008. – 461 с.

9. 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, 2009. – 460 с.

10. 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. Пол, Айра Объектно-ориентированное программирование на C++ / Айра Пол; Пер. с англ. Д. Ковальчука. – 2-е изд. – Москва: БИНОМ; Санкт-Петербург: Невский диалект, 1999. – 464 с.

12. Страуструп, Б. Язык программирования Си / Б. Страуструп; пер. с англ. М.Г. Пиголкина, В.А. Яницкого. – Москва: Радио и связь, 1991. – 348 с.

13. Страуструп, Б. Язык программирования C++ / Бьерн Страуструп ; пер. с англ. С. Анисимова, М. Кононова ; под ред. Ф. Андреева, А. Ушакова. – спец. изд. – Москва: Бином-Пресс, 2008. – 1098 с.

14. 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. – Москва: БИНОМ; Санкт-Петербург: Невский диалект, 1999. – 560 с.

11.2. Дополнительная литература

15. Кнут, Д. Э. Искусство программирования для ЭВМ : [В 7 т.] : Пер. с англ. Т. 1: Основные алгоритмы / Пер. с англ. Г.П. Бабенко, Ю.М. Баяковского; Под ред. Г.П. Бабенко, В.С. Штаркмана. – Москва: Мир, 1976. – 735 с.

16. Кнут, Д. Э. Искусство программирования для ЭВМ : [В 7 т.] : Пер. с англ. Т. 2 : Получисленные алгоритмы / Пер. с англ. Г. П. Бабенко и др.; Под ред. Г. П. Бабенко. – Москва: Мир, 1977. – 724 с.

17. 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. – Москва: 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).