

Министерство науки и высшего образования РФ Ульяновский государственный университет Ф – Рабочая программа дисциплины	Форма	
--	-------	--

**УТВЕРЖДЕНО**  
решением Ученого совета факультета математики,  
информационных и авиационных технологий  
от 21 мая 2024 г., протокол № 5/24  
Председатель \_\_\_\_\_ / М.А. Волков  
«21» мая 2024 г.



## РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Дисциплина	<b>Объектно-ориентированное программирование</b>
Факультет	Факультет математики, информационных и авиационных технологий
Кафедра	Кафедра информационных технологий
Курс	3 - очная форма обучения

Направление (специальность): 09.03.03 Прикладная информатика

Направленность (профиль/специализация): Информационная сфера

Форма обучения: очная

Дата введения в учебный процесс УлГУ: 01.09.2024 г.

Программа актуализирована на заседании кафедры: протокол № \_\_\_\_\_ от \_\_\_\_\_ 20\_\_ г.

Программа актуализирована на заседании кафедры: протокол № \_\_\_\_\_ от \_\_\_\_\_ 20\_\_ г.

Программа актуализирована на заседании кафедры: протокол № \_\_\_\_\_ от \_\_\_\_\_ 20\_\_ г.

Сведения о разработчиках:

ФИО	КАФЕДРА	Должность, ученая степень, звание
Шевченко Татьяна Валентиновна	Кафедра информационных технологий	Доцент, Кандидат физико-математических наук

Министерство науки и высшего образования РФ Ульяновский государственный университет Ф – Рабочая программа дисциплины	Форма	
--	-------	--

## **1. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ**

### **Цели освоения дисциплины:**

- обучить студентов принципам объектно-ориентированного программирования;
- обучить студентов принципам декомпозиции при решении поставленных задач;
- обучить студентов принципам современным методам написания кода.
- приобретение представлений о новейших тенденциях развития технологий программирования.

### **Задачи освоения дисциплины:**

- получить навыки работы объектно-ориентированного программирования;
- получить основу методов объектно-ориентированного проектирования программного обеспечения.

## **2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП**

Дисциплина «Объектно-ориентированное программирование» относится к числу дисциплин блока Б1.В.1, предназначенного для студентов, обучающихся по направлению: 09.03.03 Прикладная информатика.

В процессе изучения дисциплины формируются компетенции: ПК-2, ПК-7.

Основные положения дисциплины используются в дальнейшем при изучении таких дисциплин как: Преддипломная практика, Эксплуатационная практика, Проектная деятельность, Подготовка к сдаче и сдача государственного экзамена, Выполнение и защита выпускной квалификационной работы, Открытые технологии разработки программного обеспечения, Разработка мобильных приложений, Высокопроизводительные вычисления, Программирование на языке Java, Методы разработки программного обеспечения, Информационные сети, 1С: Предприятие для программистов и системных администраторов, Системы реального времени, Прикладная статистика, Имитационное моделирование, Распределенные вычисления, Современные системы автоматизации разработки информационных систем, Компьютерная геометрия и графика, Информационный менеджмент, Информатизация общества, Методы машинного обучения, Интеллектуальный анализ данных, Обнаружение вторжений и защита информации, Администрирование информационных систем, Сетевые технологии, Операционные системы, Базы данных, Технологическая (проектно-технологическая) практика, Представление знаний, Параллельное программирование, Методы и системы обработки больших данных, Функциональное программирование, Графический дизайн, Интеллектуальные системы и технологии, Программирование для Интернет, Технология программирования, Программная инженерия.

## **3. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ, СООТНЕСЕННЫХ С ПЛАНИРУЕМЫМИ РЕЗУЛЬТАТАМИ ОСВОЕНИЯ ОСНОВНОЙ**

## ПРОФЕССИОНАЛЬНОЙ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Код и наименование реализуемой компетенции	Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с индикаторами достижения компетенций
ПК-7 Способен настраивать, эксплуатировать и сопровождать информационные системы и сервисы	<p><b>знать:</b> типы прикладного ПО; методологию объектно-ориентированного программирования.</p> <p><b>уметь:</b> осуществлять выбор методологий программирования в зависимости от решаемых задач; практически использовать прикладное ПО.</p> <p><b>владеть:</b> способами внедрения ПО; приёмами сопровождения ПО</p>
ПК-2 Способен разрабатывать и адаптировать прикладное программное обеспечение	<p><b>знать:</b> методологию применения низкоуровневого и высокоуровневого программирования; методологию объектно-ориентированного.</p> <p><b>уметь:</b> осуществлять выбор низкоуровневого и высокоуровневого программирования в зависимости от решаемых задач; практически использовать процедурное, функциональное, модульное, программирование.</p> <p><b>владеть:</b> основными алгоритмами</p>

### 4. ОБЩАЯ ТРУДОЕМКОСТЬ ДИСЦИПЛИНЫ

**4.1. Объем дисциплины в зачетных единицах (всего): 6 ЗЕТ**

**4.2. Объем дисциплины по видам учебной работы (в часах): 216 часов**

Форма обучения: очная

Вид учебной работы	Количество часов (форма обучения <u>очная</u> )	
	Всего по плану	В т.ч. по семестрам
		<b>6</b>
<b>1</b>	<b>2</b>	<b>3</b>
Контактная работа обучающихся с преподавателем в соответствии с УП	72	72
Аудиторные занятия:	72	72
Лекции	36	36
Семинары и практические занятия	-	-
Лабораторные работы, практикумы	36	36
Самостоятельная работа	108	108
Форма текущего контроля знаний и контроля самостоятельной работы: тестирование, контр. работа, коллоквиум, реферат и др. (не менее 2 видов)	Тестирование	Тестирование

Министерство науки и высшего образования РФ Ульяновский государственный университет Ф – Рабочая программа дисциплины	Форма	
--	-------	--

Вид учебной работы	Количество часов (форма обучения <u>очная</u> )	
	Всего по плану	В т.ч. по семестрам
		6
1	2	3
Курсовая работа	Курсовая работа	Курсовая работа
Виды промежуточной аттестации (экзамен, зачет)	Экзамен (36)	Экзамен
Всего часов по дисциплине	216	216

### 4.3. Содержание дисциплины. Распределение часов по темам и видам учебной работы

Форма обучения: очная

Название разделов и тем	Всего	Виды учебных занятий					Форма текущего контроля знаний
		Аудиторные занятия			Занятия в интерактивной форме	Самостоятельная работа	
		Лекции	Практические занятия, семинары	Лабораторные работы, практикумы			
1	2	3	4	5	6	7	8
<b>Раздел 1. Концепции.</b>							
Тема 1.1. Сложность ПО. Структуры сложных систем.	20	4	0	4	0	12	
<b>Раздел 2. Классы и объекты.</b>							
Тема 2.1. Классы и объекты.	20	4	0	4	0	12	Тестирование
Тема 2.2. Знакомство с объектно-ориентированными языками программирования.	20	4	0	4	0	12	
<b>Раздел 3. Введение в ООП.</b>							
Тема 3.1. Программа	20	4	0	4	0	12	Тестирование

Название разделов и тем	Всего	Виды учебных занятий					Форма текущего контроля знаний
		Аудиторные занятия			Занятия в интерактивной форме	Самостоятельная работа	
		Лекции	Практические занятия, семинары	Лабораторные работы, практикумы			
1	2	3	4	5	6	7	8
и ее сборка.							
Тема 3.2. Объекты. Классы. Перегрузка.	20	4	0	4	0	12	Тестирование
Тема 3.3. Принципы ООП.	20	4	0	4	0	12	Тестирование
Тема 3.4. Интерфейсы. Делегаты.	20	4	0	4	0	12	
Тема 3.5. Коллекции. Параметризованные классы. Шаблоны.	20	4	0	4	0	12	
Тема 3.6. Потoki.	20	4	0	4	0	12	
<b>Итого подлежит изучению</b>	180	36	0	36	0	108	

## 5. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

### Раздел 1. Концепции.

#### Тема 1.1. Сложность ПО. Структуры сложных систем.

Современные технологии программирования. Парадигмы программирования. Процедурный и объектный подход к программированию. Объектно-ориентированная технология разработки программных систем и её принципы. Принципы проектирования сложных систем.

### Раздел 2. Классы и объекты.

#### Тема 2.1. Классы и объекты.

Природа объектов. Отношения между объектами. Отношения между классами. Взаимосвязь

Министерство науки и высшего образования РФ Ульяновский государственный университет Ф – Рабочая программа дисциплины	Форма	
--	-------	--

классов и объектов. Способы описания классов. Создание объектов. Обращение к атрибутам и методам объектов. Определение методов класса.

## **Тема 2.2. Знакомство с объектно-ориентированными языками программирования.**

Языки ООП и их история. С++, Java, С#. Плюсы и минусы. Сравнительная характеристика. Выбор языка ООП.

## **Раздел 3. Введение в ООП.**

### **Тема 3.1. Программа и ее сборка.**

Основные понятия языка, принципы работы Microsoft.NET. Пространство имен. Класс и структура. Модификаторы доступа. Типы данных. Литералы. Переменные, объявление и инициализация. Константы. Операции и выражения. Приведение типов. Тип object. Арифметические операции. Контроль за переполнением. Перечисления. Область видимости переменных, время жизни. Операторы и методы. Обработка исключений. Входные потоки. Массивы. Ввод/вывод.

### **Тема 3.2. Объекты. Классы. Перегрузка.**

Способы передачи параметров при вызове метода. Ссылка и ссылка на ссылку как параметры. Main как точка входа. Создание объекта. Конструктор, его виды. Операция new. Выражение this. Деструктор и сборка мусора. Перегрузка методов и операций.

### **Тема 3.3. Принципы ООП.**

Принципы ООП. Наследование. Проблемы доступа. Переопределение членов базового класса. New-модификатор. Прекращение наследования, sealed-модификатор. Абстрактные функции и абстрактные классы. Операции is, as. Виртуальные функции. Принцип полиморфизма. Инкапсуляция.

### **Тема 3.4. Интерфейсы. Делегаты.**

Объявление интерфейса. Интерфейс IComparable. Интерфейс IDisposable. Делегаты и события. Анонимные методы и делегаты для них. Различия.

### **Тема 3.5. Коллекции. Параметризованные классы. Шаблоны.**

Пространство имен System.Collection. ArrayList, Stack, Queue, IEnumerator. Шаблоны классов и функций. Ограничения. Явное и неявное преобразование.

### **Тема 3.6. Потоки.**

Многопоточные приложения. Потоки байтов, символов. Работа с каталогами и файлами. Сериализация.

Министерство науки и высшего образования РФ Ульяновский государственный университет Ф – Рабочая программа дисциплины	Форма	
--	-------	--

## 6. ТЕМЫ ПРАКТИЧЕСКИХ И СЕМИНАРСКИХ ЗАНЯТИЙ

### 7. ЛАБОРАТОРНЫЕ РАБОТЫ, ПРАКТИКУМЫ

#### Лабораторная работа 1. КЛАССЫ И ОБЪЕКТЫ

Цели: получение практических навыков работы по созданию классов и объектов.

Содержание: Определить пользовательский класс в соответствии с вариантом задания. Добавить в класс конструкторы: без параметров, с параметрами, копирования. Определить в классе деструктор (если необходимо), а также функции для просмотра и установки полей данных. Написать демонстрационную программу, в которой создаются и разрушаются объекты пользовательского класса и каждый вызов конструктора и деструктора сопровождается выдачей соответствующего сообщения. Вариант 1. Создать класс `worker`. Класс должен включать поле типа `int` для хранения номера сотрудника и поле типа `float` для хранения величины его зарплаты. Программа просит пользователя ввести данные для пяти сотрудников и выводит полученную информацию на экран. Вариант 2. Создать класс `Int`, имитирующий стандартный тип `int`. Единственное поле этого класса должно иметь тип `int`. Создать методы, которые будут устанавливать значение поля равным нулю, инициализировать его целым значением, выводить значение поля на экран и складывать два значения типа `Int`. Написать программу, в которой будут созданы три объекта класса `Int`, два из которых – инициализированы. Сложить два инициализированных объекта, присвоить результат третьему, а затем отобразить результат на экране. Вариант 3. Создать класс типа двухсвязный список. Поля-данные: указатель на область памяти типа `void`, указатели на следующий и предыдущий. Функции-члены добавляют элемент к списку, удаляют элемент из списка, выводят элементы списка от начала и от конца, ищут заданный элемент в списке. Вариант 4. Создать класс типа дата с полями: день (1–31), месяц (1–12), год (целое число). Класс имеет конструктор, функции установки дня, месяца и года, функции-члены получения дня, месяца и года, а также две функции вывода на экран: печать по шаблону «17 мая 2007 года» и «17.05.2007». Функции установки полей класса должны проверять корректность задаваемых параметров. Вариант 5. Создать класс типа прямоугольник. Поля – высота и ширина. Функции вычисляют площадь, периметр, устанавливают поля и возвращают значения. Функции установки полей класса должны проверять корректность задаваемых параметров. Вариант 6. Создать класс типа круг. Поля-данные: радиус, координаты центра. Функции вычисляют площадь, длину окружности, устанавливают поля и возвращают значения. Функции установки полей класса должны проверять корректность задаваемых параметров (не равны нулю, не отрицательные). Вариант 7. Создать класс множество `Set`. Функции реализуют добавление и удаление элемента, пересечение и разность множеств. Вариант 8. Создать класс типа квадрат. Поля-данные: сторона. Функции вычисляют площадь, периметр, устанавливают поля и возвращают значения. Функции установки полей класса должны проверять корректность задаваемых параметров. Вариант 9. Создать класс типа время с полями: час (0–23), минуты (0–59), секунды (0–59). Класс имеет конструктор. Функции установки времени, получения часа, минуты и секунды, а также две функции вывода на экран: печать по шаблону «15 часов 57 минут 30 секунд» и «3 p.m. 57 минут 30 секунд». Функции установки полей класса должны проверять корректность задаваемых параметров. Вариант 10. Создать класс одномерный массив целых чисел (вектор) с полями – количество фактических элементов, массив (динамический). Функции: обращения к отдельному элементу массива, вывода массива на экран, поэлементного сложения и вычитания со

Министерство науки и высшего образования РФ Ульяновский государственный университет Ф – Рабочая программа дисциплины	Форма	
--	-------	--

скаляром, вывода элемента по заданному индексу.

Результаты: Проект согласно заданию. Отчет о ходе выполнения.

Ссылка: <http://lib.ulsu.ru/ProtectedView/Book/ViewBook/7232>

### Лабораторная работа 2. НАСЛЕДОВАНИЕ

Цели: получение практических навыков по созданию иерархии классов.

Содержание: Определить иерархию классов (в соответствии с вариантом – выделить базовый и производные). Реализовать классы (самостоятельно задать члены-данные и методы класса). Написать демонстрационную программу, в которой создаются объекты различных классов. Определение классов, их реализацию, демонстрационную программу поместить в отдельные модули. Вариант 1. Классы – человек (имя, дата рождения), абитуриент (количество баллов ЕГЭ), студент (курс, группа, факультет), преподаватель (должность, кафедра) Вариант 2. Классы – растение (название, вид), дерево (возраст), цветок (длина стебля), тюльпан (цвет) Вариант 3. Классы – кадры (имя), рабочий (специальность, цех), инженер (квалификация, подразделение), администрация (должность) Вариант 4. Классы – УлГУ (адрес), факультет (название), группа (номер, староста, курс), подгруппа (номер, количество студентов) Вариант 5. Классы – печатное издание (издательство, год, название), журнал (номер, месяц), книга (тематика, автор, количество страниц), учебник (назначение) Вариант 6. Классы – млекопитающие (год), парнокопытные (среда обитания), птицы (хищники), животное (вид, род, вес) Вариант 7. Классы – место (площадь, название), область (количество населенных пунктов, руководство), город (область, количество жителей, мэр), деревня (район) Вариант 8. Классы – товар (название), радиотовары (назначение), продукт (отдел), хлебный продукт (разновидность, дата изготовления), транзистор (тип, номер) Вариант 9. Классы – автомобиль (марка, номер), поезд (номер, количество вагонов, количество пассажиров в вагоне), транспортное средство (средняя скорость, вид топлива, год выпуска) Вариант 10. Классы – республика (вид, правительство), монархия (вид, имя монарха), королевство (король), государство (название, денежная единица, символика)

Результаты: Проект согласно заданию. Отчет о ходе выполнения.

Ссылка: <http://lib.ulsu.ru/ProtectedView/Book/ViewBook/7232>

### Лабораторная работа 3. АБСТРАКТНЫЕ КЛАССЫ

Цели: получение практических навыков по работе с абстрактными классами.

Содержание: Использовать проект, созданный в лабораторной работе № 2. Расширить иерархию классов с использованием виртуального абстрактного класса в качестве основы иерархии (использовать чисто виртуальные функции). Определить в классе статическую компоненту – указатель на начало связанного списка объектов и статическую функцию для просмотра списка (инициализировать вне определения класса, в глобальной области). Статический метод просмотра списка вызывать не через объект, а через класс. Написать демонстрационную программу, в которой создаются и помещаются в список (методом класса) объекты различных классов, после чего список просматривается.

Результаты: Проект согласно заданию. Отчет о ходе выполнения.

Ссылка: <http://lib.ulsu.ru/ProtectedView/Book/ViewBook/7232>

### Лабораторная работа 4. ДИНАМИЧЕСКАЯ ИДЕНТИФИКАЦИЯ ТИПОВ

Цели: получение практических навыков по работе с итераторами.

Содержание: Использовать проект, созданный в лабораторной работе № 3. Определить методы-итераторы в неабстрактных классах на основе следующих запросов: Вариант 1. Имена студентов указанного курса, имена и должность преподавателей указанной кафедры Вариант 2. Все чайные тюльпаны с длиной стебля большей, чем задал пользователь, количество тюльпанов заданного цвета Вариант 3. Имена рабочих заданного цеха, количество инженеров в заданном подразделении

Министерство науки и высшего образования РФ Ульяновский государственный университет Ф – Рабочая программа дисциплины	Форма	
--	-------	--

Вариант 4. Количество групп заданного курса, старост всех групп определенного факультета  
 Вариант 5. Наименования всех книг в библиотеке/магазине, вышедших не ранее указанного года; суммарное количество учебников в библиотеке/магазине  
 Вариант 6. Средний вес животных заданного вида в зоопарке, количество хищных птиц  
 Вариант 7. Названия всех городов заданной области, суммарное количество жителей всех городов в области  
 Вариант 8. Наименования всех товаров в заданном отделе магазина, суммарная стоимость товара заданного наименования  
 Вариант 9. Количество указанных транспортных средств в автопарке/на автостоянке, количество мест во всех вагонах поездов  
 Вариант 10. Имена всех монархов на заданном континенте, количество демократических государств

Результаты: Проект согласно заданию. Отчет о ходе выполнения.

Ссылка: <http://lib.ulsu.ru/ProtectedView/Book/ViewBook/7232>

#### Лабораторная работа 5. ПЕРЕГРУЗКА ОПЕРАЦИЙ

Цели: получение практических навыков по перегрузке операций.

Содержание: Создать заданный в варианте класс. Определить в классе конструкторы, деструктор, необходимые функции и заданные перегруженные операции. Написать программу тестирования, в которой проверяется использование всех перегруженных операций.

Вариант 1. Класс – одномерный массив. Дополнительно перегрузить следующие операции: \* – умножение массивов; [] – доступ по индексу, int() – размер массива; == – проверка на равенство; <= – сравнение  
 Вариант 2. Класс – одномерный массив. Дополнительно перегрузить следующие операции: [] – доступ по индексу; == – проверка на равенство; != – проверка на неравенство; + – объединение массивов  
 Вариант 3. Класс – множество Cset. Дополнительно перегрузить следующие операции: + – добавить элемент в множество; + – объединение множеств; \* – пересечение множеств, == – проверка множеств на равенство; int () – мощность множества  
 Вариант 4. Класс – очередь Cqueue. Дополнительно перегрузить следующие операции: + – добавить элемент; -- – извлечь элемент; bool () – проверка, пустая ли очередь; < – копирование одной очереди в другую с сортировкой в убывающем порядке; int () – мощность  
 Вариант 5. Класс – множество Cset. Дополнительно перегрузить следующие операции: () – конструктор множества (в стиле конструктора для множественного типа); + – объединение множеств; <= – сравнение множеств; int()– мощность множества; [] – доступ к элементу в заданной позиции  
 Вариант 6. Класс – однонаправленный список list. Дополнительно перегрузить следующие операции: + – объединить два списка; -- – удалить элемент из начала; == – проверка на равенство; bool () – проверка, пустой ли список  
 Вариант 7. Класс – матрица. Дополнительно перегрузить следующие операции: + – сложение матриц; () – доступ по индексу M(c, i); > – сравнение матриц; == – копирование матриц  
 Вариант 8. Класс – время CVector. Дополнительно перегрузить следующие операции: + – добавление элемента; ++ – увеличение всех элементов вектора на один; != – проверка на неравенство; [] – доступ к элементу в заданной позиции  
 Вариант 9. Класс – время CVector. Дополнительно перегрузить следующие операции: - – разность двух векторов; + – сложение элементов вектора с целым; != – проверка на неравенство; [] – доступ к элементу в заданной позиции  
 Вариант 10. Класс – стек Cstack. Дополнительно перегрузить следующие операции: + – добавить элемент в стек; -- – извлечь элемент из стека; bool() – проверка, пустой ли стек; > – копирование одного стека в другой с сортировкой в возрастающем порядке

Результаты: Проект согласно заданию. Отчет о ходе выполнения.

Ссылка: <http://lib.ulsu.ru/ProtectedView/Book/ViewBook/7232>

#### Лабораторная работа 6. ШАБЛОНЫ КЛАССОВ И ФУНКЦИЙ

Цели: получение практических навыков по работе с шаблонами.

Содержание: Модифицировать проект, созданный в лабораторной работе № 5. Создать шаблон заданного по варианту класса. Проверить использование шаблона для стандартных типов данных

Министерство науки и высшего образования РФ Ульяновский государственный университет Ф – Рабочая программа дисциплины	Форма	
--	-------	--

(int, char/double). Определить пользовательский класс, который будет использоваться в качестве параметра шаблона. Для пользовательского типа взять класс из проекта задания № 1 или № 2 своего варианта (например, реализация шаблона – множество прямоугольников). В пользовательском классе перегрузить операции, необходимые для работы шаблона. Добавить функцию-шаблон в соответствии с вариантом. Проверить работу шаблон-функции с int, double и пользовательским классом. Если имеется несколько полей, для манипуляций выбрать одно из полей класса. Вариант 1. Написать функцию-шаблон, вычисляющую среднее значение в массиве. Вариант 2. Написать функцию-шаблон поиска минимального значения. Вариант 3. Написать функцию-шаблон удаления каждого второго элемента в множестве. Вариант 4. Написать функцию-шаблон поиска максимального элемента очереди. Вариант 5. Написать функцию-шаблон поиска наиболее близкого элемента множества к заданному. Аргументами функции должны быть значение и множество. Вариант 6. Создать шаблон функции прореживания списка. Функция имеет три параметра: откуда выбирать, куда выбирать и через сколько выбирать. Вариант 7. Создать шаблон функции, которая возвращает сумму минимального и максимального из значений матрицы. Вариант 8. Написать функцию-шаблон, вычисляющую произведение максимального и минимального значения в векторе. Вариант 9. Создать шаблон функции, которая возвращает максимально близкое к заданному элементу из значений вектора. Аргументами функции должны быть значение и вектор. Вариант 10. Написать функцию-шаблон, которая подсчитывает количество элементов, больших заданного, в стеке.

Результаты: Проект согласно заданию. Отчет о ходе выполнения.

Ссылка: <http://lib.ulsu.ru/ProtectedView/Book/ViewBook/7232>

#### Лабораторная работа 7. РАБОТА С ПОТОКАМИ, ОБРАБОТКА ИСКЛЮЧЕНИЙ

Цели: получение практических навыков по работе с потоками и обработке исключений.

Содержание: Модифицировать проект, созданный в лабораторной работе № 6. Реализовать ввод-вывод данных на экран и в файл через потоковые классы. Например, считать из текстового файла количество и типы создаваемых объектов, а также их значения. Сгенерировать и обработать как минимум пять различных исключительных ситуаций. Например, не позволять при инициализации объектов передавать конструкторам неверные данные, обрабатывать ошибки памяти и ошибки работы с файлами. Построить иерархию классов исключений.

Результаты: Проект согласно заданию. Отчет о ходе выполнения.

Ссылка: <http://lib.ulsu.ru/ProtectedView/Book/ViewBook/7232>

#### Лабораторная работа 8. КОНТЕЙНЕРЫ

Цели: получение практических навыков по работе с контейнерами.

Содержание: Проект должен содержать демонстрации: использования контейнерных классов для хранения встроенных типов данных; использования контейнерных классов для хранения пользовательских типов данных; использования алгоритмов STL. Выполнить пункты а)-з): а. Создать объект-контейнер в соответствии с вариантом задания и заполнить его данными согласно варианту. Просмотреть контейнер. б. Изменить контейнер, удалив из него одни элементы и заменив другие. в. Просмотреть контейнер, используя для доступа к элементам итераторы. г. Изменить контейнер, удалив из него n элементов после заданного. Просмотреть контейнер. д. Отсортировать контейнер по убыванию элементов. Просмотреть контейнер. е. Выполнить пункты а)–г), но для данных пользовательского типа (в качестве пользовательского типа данных использовать пользовательский класс лабораторной работы № 1). Для вставки и удаления элементов контейнера использовать соответствующие операции, определенные в классе контейнера. Для ввода-вывода объектов пользовательского класса следует перегрузить операции >> и <. ж. Используя подходящий алгоритм, найти в контейнере пользовательского типа заданный элемент (заданное значение).

Министерство науки и высшего образования РФ Ульяновский государственный университет Ф – Рабочая программа дисциплины	Форма	
--	-------	--

самостоятельно). 3. Подсчитать, сколько элементов, удовлетворяющих заданному условию, содержат все контейнеры проекта Вариант 1. Контейнер STL – stack, встроенный тип данных - char. Вариант 2. Контейнер STL – multimap, встроенный тип данных - int. Вариант 3. Контейнер STL – list, встроенный тип данных - long. Вариант 4. Контейнер STL – deque, встроенный тип данных - float. Вариант 5. Контейнер STL – stack, встроенный тип данных - double. Вариант 6. Контейнер STL – queue, встроенный тип данных - char. Вариант 7. Контейнер STL – vector, встроенный тип данных - string. Вариант 8. Контейнер STL – set, встроенный тип данных - int. Вариант 9. Контейнер STL – vector, встроенный тип данных - double. Вариант 10. Контейнер STL – map, встроенный тип данных - int.

Результаты: Проект согласно заданию. Отчет о ходе выполнения.

Ссылка: <http://lib.ulsu.ru/ProtectedView/Book/ViewBook/7232>

#### Лабораторная работа 9. ИНТЕРФЕЙСЫ И НАСЛЕДОВАНИЕ

Цели: получение практических навыков по работе с интерфейсами.

Содержание: Определить иерархию классов в соответствии с вариантом. Классы должны содержать конструкторы (с параметрами и без), статический конструктор, свойства get, set, переопределение методов (override) в производных классах. Создать в одном из классов статический метод вывода количества объектов и статическое поле – номер объекта. В проекте должны быть или интерфейс (при множественном наследовании), или абстрактный класс (при единичном наследовании). В методах класса для работы с аргументами используйте ref - и out-параметры. При возникновении ошибок должны выбрасываться исключения. Выполните обработку исключений, используя try, catch и finally. Создайте бесплодный класс, указанный в варианте. Напишите демонстрационную программу, в которой создаются объекты различных классов и добавляются в коллекцию из пространства имен System.Collections. Вариант 1. Классы – человек, абитуриент, студент, преподаватель, зав. кафедрой ИТ (бесплодный) Вариант 2. Классы – растение, дерево, цветок, роза, margella (бесплодный) Вариант 3. Классы – кадры, рабочий, инженер, администрация, зам. ген. директора (бесплодный) Вариант 4. Классы – университет, факультет, группа, подгруппа, факультет ФМИАТ (бесплодный) Вариант 5. Классы – печатное издание, журнал, книга, учебник, журнал «IT-тренды»(бесплодный) Вариант 6. Классы – млекопитающие, парнокопытные, птицы, животное, тибетский волк (бесплодный) Вариант 7. Классы – место, область, город, деревня, Ульяновск (бесплодный) Вариант 8. Классы – товар, радиотовары, продукт, хлебный продукт, транзистор, фен (бесплодный) Вариант 9. Классы – тест, экзамен, выпускной экзамен, испытание, ООП-экзамен (бесплодный) Вариант 10. Классы – республика, монархия, королевство, государство, РФ (бесплодный)

Результаты: Проект согласно заданию. Отчет о ходе выполнения.

Ссылка: <http://lib.ulsu.ru/ProtectedView/Book/ViewBook/7232>

#### Лабораторная работа 10. ОБОБЩЕНИЯ И ПЕРЕГРУЗКА ОПЕРАТОРОВ

Цели: получение практических навыков по обобщению изученного материала.

Содержание: Создать заданный в варианте обобщенный класс. Определить в классе индексы, конструкторы, деструктор, необходимые функции и заданные перегруженные операции. В методах класса для работы с аргументами используйте ref- и out-параметры и операторы is или as, foreach. Используйте обработку исключительных ситуаций. Вариант 1. Класс – одномерный массив. Дополнительно перегрузить следующие операции: \* – умножение массивов; int() – преобразование; ==, != – проверка на равенство, неравенство; <=, >= – сравнение Вариант 2. Класс – одномерный массив. Дополнительно перегрузить следующие операции: == – проверка на равенство; != – проверка на неравенство; ++ – инкремент массивов; --- декремент массивов Вариант 3. Класс – множество Cset. Дополнительно перегрузить следующие операции: + – добавить элемент в

Министерство науки и высшего образования РФ Ульяновский государственный университет Ф – Рабочая программа дисциплины	Форма	
--	-------	--

множество (типа set + item); \* – пересечение множеств, ==, != – проверка множеств на равенство, неравенство  
**Вариант 4.** Класс – очередь Queue. Дополнительно перегрузить следующие операции: + – добавить элемент; - – извлечь элемент; bool () – проверка, пустая ли очередь; < – копирование одной очереди в другую с сортировкой в убывающем порядке; > – копирование одной очереди в другую с сортировкой в возрастающем порядке  
**Вариант 5.** Класс – множество Cset. Дополнительно перегрузить следующие операции: + – объединение множеств; <=, >= – сравнение множеств; int () – мощность множества; & – пересечение  
**Вариант 6.** Класс – однонаправленный список list. Дополнительно перегрузить следующие операции: + – объединить два списка; - – удалить элемент; ==, != – проверка на равенство, неравенство; ~ – проверка, пустой ли список  
**Вариант 7.** Класс – матрица. Дополнительно перегрузить следующие операции: + – сложение матриц; () – доступ по индексу M(c, i); >, < – сравнение матриц; == – копирование матриц, != – проверка соответствия размеров матриц  
**Вариант 8.** Класс – вектор CVector. Дополнительно перегрузить следующие операции: + – добавление элемента; ++, -- – увеличение и уменьшения всех элементов вектора на один (CVector++); !=, == – проверка на неравенство, равенство  
**Вариант 9.** Класс – вектор CVector. Дополнительно перегрузить следующие операции: - – разность двух векторов; + – сложение элементов вектора с целым; !=, == – проверка на неравенство, равенство  
**Вариант 10.** Класс – стек Cstack. Дополнительно перегрузить следующие операции: + – добавить элемент в стек; - – извлечь элемент из стека; ~ – проверка, пустой ли стек; > и < – копирование одного стека в другой с сортировкой в возрастающем порядке

Результаты: Проект согласно заданию. Отчет о ходе выполнения.

Ссылка: <http://lib.ulsu.ru/ProtectedView/Book/ViewBook/7232>

## 8. ТЕМАТИКА КУРСОВЫХ, КОНТРОЛЬНЫХ РАБОТ, РЕФЕРАТОВ

### Темы курсовой работы

Тема 1. Объектно-ориентированный анализ и программная реализация прикладной задачи

Тема 2. Разработка объектной модели движения на перекрестке дорог.

Тема 3. Разработка объектной модели Солнечной системы

Тема 4. Разработка программы моделирования случайных процессов с заданными характеристиками.

Тема 5. Разработка программы поиска выхода из лабиринта.

Тема 6. Разработка приложения на языке C# и движке Unity 3D.

Тема 7. Перегрузка стандартных операций языка программирования для пользовательских объектов.

Тема 8. Разработка бота для социальной сети «ВКонтакте».

Тема 9. Разработка аркадной игры.

Тема 10. Мониторинг приложений, работающих с виртуальной реальностью.

Тема 11. Использование языка программирования Python для создания web-чата.

Тема 12. Разработка критериев проверки корректности учебного расписания.

Тема 13. Информационная система «Учебное расписание».

Тема 14. Создание игрового приложения «Тетрис» на языке программирования C#.

Тема 15. Создание программного комплекса поддержки принятия решения о выборе транспортного средства обслуживания населения.

Тема 16. Объектная реализация контейнера на основе комбинированной структуры "Упорядоченный массив динамических очередей".

Министерство науки и высшего образования РФ Ульяновский государственный университет Ф – Рабочая программа дисциплины	Форма	
--	-------	--

Тема 17. Объектная реализация контейнера на основе комбинированной структуры «Двунаправленный список массивов-стеков»

Тема 18. Объектная реализация контейнера на основе комбинированной структуры "Упорядоченный массив динамических очередей"

Тема 19. Разработка объектной программы для задачи управления новостным Internet-порталом

Тема 20. Разработка объектной программы для задачи учета компьютерной техники в организации

Тема 21. Основы объектно-ориентированного визуального программирования

Тема 22. Проектирование библиотеки классов для реализации оконного интерфейса на платформе MS Windows

Тема 23. Методология объектно-ориентированного программирования. Свойства экторной модели параллельных вычислений Хьюита

Тема 24. Разработка механизма сборки мусора объектов на языке программирования C++

Тема 25. Исследование графических возможностей Windows. Создание приложения (на любом языке программирования), которое использует возможности интерфейса Aero.

Тема 26. Программирование гаджетов (Gadgets) в Windows. Создание гаджета для отображения картинки с произвольным текстом поверх нее.

Тема 27. Использование возможностей видеокарты для выполнения вычислений (CUDA от nVidia, ATI Stream и OpenCL). Написание программы шифрации и дешифрации с использованием возможностей видеокарты.

Тема 28. Исследование возможностей стандарта web-разработки HTML5 и CSS3 (каскадные таблицы стилей 3-й версии). Создание web-странички отображающей графики и диаграммы средствами HTML5 (без применения JavaScript или с его минимальным использованием).

Тема 29. Использование хранимых процедур CLR в SQL Server 2008 R2. Исследование возможностей и сравнение с конкурирующими технологиями (обычные хранимые процедуры в TransactSQL, PL/SQL в Oracle и в PostgreSQL,...). Создание программы, у которой вся бизнес-логика реализована средствами хранимых процедур CLR (т.е. в программе имеются только вызовы этих процедур).

Тема 30. Библиотека классов, реализующих графические примитивы на плоскости с возможностью аффинных преобразований.

Тема 31. Библиотека классов, реализующих графические примитивы на плоскости с реализацией операций над множествами.

Тема 32. Библиотека классов, реализующих графические примитивы в трёхмерном пространстве с возможностью аффинных преобразований.

Тема 33. Классы для описания оконного графического интерфейса, аналогичного интерфейсу MS Windows. Должны быть реализованы обработчики событий с использованием виртуальных функций.

Тема 34. Библиотека классов, реализующих комплексную арифметику.

Тема 35. Библиотека классов, реализующих векторы в n-мерном пространстве.

Тема 36. Реализация строк и операций над ними, включая работу с регулярными выражениями.

Тема 37. «Обобщённый массив» (позволяющий хранить данные произвольных типов).

Тема 38. Ассоциативный массив (хэш-массив) (с возможностью хранения данных произвольных типов).

Тема 39. Реализация различных типов графов и операций над ними.

Тема 40. Система классов для обеспечения работы с абонентами телефонной компании.

Тема 41. Система классов для обеспечения работы деканата.

Тема 42. Система классов, описывающих сотрудников предприятия/организации с их функциями (сотрудник, менеджер, ...).

Министерство науки и высшего образования РФ Ульяновский государственный университет Ф – Рабочая программа дисциплины	Форма	
--	-------	--

- Тема 43. Система классов, описывающих различные транспортные средства.  
Тема 44. Моделирование замкнутой биологической системы (корм, травоядное, хищник).  
Тема 45. Моделирование муравейника (несколько типов муравьёв, источники питания, внешние раздражители,...).  
Тема 46. Моделирование дорожного движения на заданной карте дорог.  
Тема 47. Моделирование компьютерной сети (стационарной).  
Тема 48. Моделирование компьютерной сети с изменяющейся топологией.  
Тема 49. Моделирование планетарной системы.  
Тема 50. Система классов для реализации матричной арифметики.  
Тема 51. Библиотека классов для реализации игрового приложения.

### 9. ПЕРЕЧЕНЬ ВОПРОСОВ К ЭКЗАМЕНУ

1. Сложность ПО. Структуры сложных систем1.
2. Проектирование сложных систем
3. Природа объектов. Отношения между объектами
4. Отношения между классами. Взаимосвязь классов и объектов
5. Объектно-ориентированные языки программирования. История. Виды. Сравнительная характеристика.
6. Введение в ООП. Основные понятия языка.
7. Объявление класса. Объявление объекта. Перегрузка методов и операций.
8. Принципы ООП. Наследование. Полиморфизм.
9. Интерфейсы и делегаты.
10. Коллекции. Параметризованные классы.
11. Шаблоны.
12. Потоки. Работа с файлами.

### 10. САМОСТОЯТЕЛЬНАЯ РАБОТА ОБУЧАЮЩИХСЯ

*Содержание, требования, условия и порядок организации самостоятельной работы обучающихся с учетом формы обучения определяются в соответствии с «Положением об организации самостоятельной работы обучающихся», утвержденным Ученым советом УлГУ (протокол №8/268 от 26.03.2019г.).*

*По каждой форме обучения: очная/заочная/очно-заочная заполняется отдельная таблица*

Форма обучения: очная

<b>Название разделов и тем</b>	<b>Вид самостоятельной работы (проработка учебного материала, решение задач, реферат, доклад, контрольная работа, подготовка к сдаче зачета, экзамена и др).</b>	<b>Объем в часах</b>	<b>Форма контроля (проверка решения задач, реферата и др.)</b>
<b>Раздел 1. Концепции.</b>			
Тема 1.1. Сложность ПО. Структуры сложных систем.	Проработка учебного материала с использованием ресурсов учебно-методического и информационного обеспечения дисциплины.	12	Вопросы к экзамену
<b>Раздел 2. Классы и объекты.</b>			
Тема 2.1. Классы и объекты.	Проработка учебного материала с использованием ресурсов учебно-методического и информационного обеспечения дисциплины.	12	Вопросы к экзамену, Тестирование
Тема 2.2. Знакомство с объектно-ориентированными языками программирования.	Проработка учебного материала с использованием ресурсов учебно-методического и информационного обеспечения дисциплины.	12	Вопросы к экзамену
<b>Раздел 3. Введение в ООП.</b>			
Тема 3.1. Программа и ее сборка.	Проработка учебного материала с использованием ресурсов учебно-методического и информационного обеспечения дисциплины.	12	Вопросы к экзамену, Тестирование
Тема 3.2. Объекты. Классы. Перегрузка.	Проработка учебного материала с использованием ресурсов учебно-методического и информационного обеспечения дисциплины.	12	Вопросы к экзамену, Тестирование
Тема 3.3. Принципы ООП.	Проработка учебного материала с использованием ресурсов учебно-методического и информационного обеспечения дисциплины.	12	Вопросы к экзамену, Тестирование
Тема 3.4. Интерфейсы. Делегаты.	Проработка учебного материала с использованием ресурсов учебно-методического и информационного обеспечения дисциплины.	12	Вопросы к экзамену

Министерство науки и высшего образования РФ Ульяновский государственный университет Ф – Рабочая программа дисциплины	Форма	
--	-------	--

Название разделов и тем	Вид самостоятельной работы (проработка учебного материала, решение задач, реферат, доклад, контрольная работа, подготовка к сдаче зачета, экзамена и др).	Объем в часах	Форма контроля (проверка решения задач, реферата и др.)
Тема 3.5. Коллекции. Параметризованные классы. Шаблоны.	Проработка учебного материала с использованием ресурсов учебно-методического и информационного обеспечения дисциплины.	12	Вопросы к экзамену
Тема 3.6. Поток.	Проработка учебного материала с использованием ресурсов учебно-методического и информационного обеспечения дисциплины.	12	Вопросы к экзамену

## 11. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

### а) Список рекомендуемой литературы

#### основная

1. Зыков Сергей Викторович. Программирование. Объектно-ориентированный подход : Учебник и практикум для вузов / С.В. Зыков ; Зыков С. В. - Москва : Юрайт, 2022. - 155 с. - (Высшее образование). - URL: <https://urait.ru/bcode/490423> (дата обращения: 24.01.2022). - Режим доступа: Электронно-библиотечная система Юрайт, для авториз. пользователей. - Электрон. дан. - ISBN 978-5-534-00850-0 : 559.00. / .— ISBN 0\_319728

2. Гниденко И. Г. Технологии и методы программирования : учебное пособие / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. - 2-е изд. ; пер. и доп. - Москва : Юрайт, 2024. - 248 с. - (Высшее образование). - URL: <https://urait.ru/bcode/536903> . - Режим доступа: Электронно-библиотечная система Юрайт, для авториз. пользователей. - ISBN 978-5-534-18130-2 : 1059.00. / .— ISBN 0\_522596

3. Хорев Павел Борисович. Объектно-ориентированное программирование с примерами на C# : Учебное пособие / П.Б. Хорев ; Московский энергетический институт. - 1. - Москва : Издательство "ФОРУМ", 2023. - 200 с. - (Высшее образование: Бакалавриат). - ВО - Бакалавриат. - Режим доступа: ЭБС Znanium; по подписке. - ISBN 978-5-00091-680-3. - ISBN 978-5-16-103810-9. - ISBN 978-5-16-015117-5. / .— ISBN 0\_478815

#### дополнительная

1. Непейвода, Н. Н. Стили и методы программирования : учебное пособие / Н. Н. Непейвода ; Н. Н. Непейвода. - Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. - 295 с. - Книга находится в премиум-версии ЭБС IPR BOOKS. - Текст. - Гарантированный срок размещения в ЭБС до 24.12.2022 (автопродлонгация). - электронный. -

Министерство науки и высшего образования РФ Ульяновский государственный университет Ф – Рабочая программа дисциплины	Форма	
--	-------	--

Электрон. дан. (1 файл). - URL: <http://www.iprbookshop.ru/102065.html>. - Режим доступа: ЭБС IPR BOOKS; для авторизир. пользователей. - ISBN 978-5-4497-0938-7. / .— ISBN 0\_157088

2. Тузовский Анатолий Федорович. Объектно-ориентированное программирование : Учебное пособие для вузов / А.Ф. Тузовский. - Москва : Юрайт, 2021. - 206 с. - (Высшее образование). - <https://urait.ru/bcode/470223>. - <https://urait.ru/book/cover/5A5B4C4A-8623-4001-9D33-BFA775413E60>. - Режим доступа: Электронно-библиотечная система Юрайт, для авториз. пользователей. - Электрон. дан. - ISBN 978-5-534-00849-4 : 649.00. / .— ISBN 0\_282271

### **учебно-методическая**

1. Жаркова Г. А. Методические указания для выполнения лабораторных работ студентов по дисциплине «Объектно-ориентированное программирование» для студентов бакалавриата по направлениям подготовки 09.03.03 «Прикладная информатика», направленность (профиль/специализация) Информационная сфера и 02.03.03 «Математическое обеспечение и администрирование информационных систем», направленность (профиль/специализация) Технология программирования очной формы обучения / Г. А. Жаркова ; УлГУ, ФМИиАТ. - 2019. - Загл. с экрана. - Неопубликованный ресурс. - Электрон. текстовые дан. (1 файл : 355 КБ). - URL: <http://lib.ulsu.ru/MegaPro/Download/MObject/7235>. - Режим доступа: ЭБС УлГУ. - Текст : электронный. / .— ISBN 0\_40644.

2. Жаркова Г. А. Методические указания для выполнения курсовой работы студентов по дисциплине «Объектно-ориентированное программирование» для студентов бакалавриата по направлениям подготовки 09.03.03 «Прикладная информатика», направленность (профиль/специализация) Информационная сфера и 02.03.03 «Математическое обеспечение и администрирование информационных систем», направленность (профиль/специализация) Технология программирования очной формы обучения / Г. А. Жаркова ; УлГУ, ФМИиАТ. - 2019. - Загл. с экрана. - Неопубликованный ресурс. - Электрон. текстовые дан. (1 файл : 350 КБ). - URL: <http://lib.ulsu.ru/MegaPro/Download/MObject/7238>. - Режим доступа: ЭБС УлГУ. - Текст : электронный. / .— ISBN 0\_40647.

3. Жаркова Г. А. Методические указания для самостоятельной работы студентов по дисциплине «Объектно-ориентированное программирование» для студентов бакалавриата по направлениям подготовки 09.03.03 «Прикладная информатика», направленность (профиль/специализация) Информационная сфера и 02.03.03 «Математическое обеспечение и администрирование информационных систем», направленность (профиль/специализация) Технология программирования очной формы обучения / Г. А. Жаркова ; УлГУ, ФМИиАТ. - 2019. - Загл. с экрана. - Неопубликованный ресурс. - Электрон. текстовые дан. (1 файл : 315 КБ). - URL: <http://lib.ulsu.ru/MegaPro/Download/MObject/7244>. - Режим доступа: ЭБС УлГУ. - Текст : электронный. / .— ISBN 0\_40655.

### **б) Программное обеспечение**

- Операционная система "Альт образование"
- Офисный пакет "Мой офис"
- Visual studio code

Министерство науки и высшего образования РФ Ульяновский государственный университет Ф – Рабочая программа дисциплины	Форма	
--	-------	--

## в) Профессиональные базы данных, информационно-справочные системы

### 1. Электронно-библиотечные системы:

1.1. Цифровой образовательный ресурс IPRsmart : электронно-библиотечная система : сайт / ООО Компания «Ай Пи Ар Медиа». - Саратов, [2024]. – URL: <http://www.iprbookshop.ru>. – Режим доступа: для зарегистрир. пользователей. - Текст : электронный.

1.2. Образовательная платформа ЮРАЙТ : образовательный ресурс, электронная библиотека : сайт / ООО Электронное издательство ЮРАЙТ. – Москва, [2024]. - URL: <https://urait.ru>. – Режим доступа: для зарегистрир. пользователей. - Текст : электронный.

1.3. База данных «Электронная библиотека технического ВУЗа (ЭБС «Консультант студента») : электронно-библиотечная система : сайт / ООО Политехресурс. – Москва, [2024]. – URL: <https://www.studentlibrary.ru/cgi-bin/mb4x>. – Режим доступа: для зарегистрир. пользователей. – Текст : электронный.

1.4. Консультант врача. Электронная медицинская библиотека : база данных : сайт / ООО Высшая школа организации и управления здравоохранением-Комплексный медицинский консалтинг. – Москва, [2024]. – URL: <https://www.rosmedlib.ru>. – Режим доступа: для зарегистрир. пользователей. – Текст : электронный.

1.5. Большая медицинская библиотека : электронно-библиотечная система : сайт / ООО Букап. – Томск, [2024]. – URL: <https://www.books-up.ru/ru/library/> . – Режим доступа: для зарегистрир. пользователей. – Текст : электронный.

1.6. ЭБС Лань : электронно-библиотечная система : сайт / ООО ЭБС Лань. – Санкт-Петербург, [2024]. – URL: <https://e.lanbook.com>. – Режим доступа: для зарегистрир. пользователей. – Текст : электронный.

1.7. ЭБС **Znanium.com** : электронно-библиотечная система : сайт / ООО Знаниум. - Москва, [2024]. - URL: <http://znanium.com> . – Режим доступа : для зарегистрир. пользователей. - Текст : электронный.

**2. КонсультантПлюс** [Электронный ресурс]: справочная правовая система. /ООО «Консультант Плюс» - Электрон. дан. - Москва : КонсультантПлюс, [2024].

**3. eLIBRARY.RU**: научная электронная библиотека : сайт / ООО «Научная Электронная Библиотека». – Москва, [2024]. – URL: <http://elibrary.ru>. – Режим доступа : для авториз. пользователей. – Текст : электронный

**4. Федеральная государственная информационная система «Национальная электронная библиотека»** : электронная библиотека : сайт / ФГБУ РГБ. – Москва, [2024]. – URL: <https://нэб.рф>. – Режим доступа : для пользователей научной библиотеки. – Текст : электронный.

**5. Российское образование** : федеральный портал / учредитель ФГАУ «ФИЦТО». – URL:

Министерство науки и высшего образования РФ Ульяновский государственный университет Ф – Рабочая программа дисциплины	Форма	
--	-------	--

<http://www.edu.ru>. – Текст : электронный.

**6. Электронная библиотечная система УлГУ :** модуль «Электронная библиотека» АБИС Мега-ПРО / ООО «Дата Экспресс». – URL: <http://lib.ulsu.ru/MegaPro/Web>. – Режим доступа : для пользователей научной библиотеки. – Текст : электронный.

## 12. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ:

Аудитории для проведения лекций, семинарских занятий, для выполнения лабораторных работ и практикумов, для проведения текущего контроля и промежуточной аттестации, курсового проектирования, групповых и индивидуальных консультаций (*выбрать необходимое*)

Аудитории укомплектованы специализированной мебелью, учебной доской. Аудитории для проведения лекций оборудованы мультимедийным оборудованием для представления информации большой аудитории. Помещения для самостоятельной работы оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа к электронной информационно-образовательной среде, электронно-библиотечной системе. Перечень оборудования, используемого в учебном процессе:

- Мультимедийное оборудование: компьютер/ноутбук, экран, проектор/телевизор
- Компьютерная техника

## 13. СПЕЦИАЛЬНЫЕ УСЛОВИЯ ДЛЯ ОБУЧАЮЩИХСЯ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ

В случае необходимости, обучающимся из числа лиц с ограниченными возможностями здоровья (по заявлению обучающегося) могут предлагаться одни из следующих вариантов восприятия информации с учетом их индивидуальных психофизических особенностей:

- для лиц с нарушениями зрения: в печатной форме увеличенным шрифтом; в форме электронного документа; в форме аудиофайла (перевод учебных материалов в аудиоформат); в печатной форме на языке Брайля; индивидуальные консультации с привлечением тифлосурдопереводчика; индивидуальные задания и консультации;

- для лиц с нарушениями слуха: в печатной форме; в форме электронного документа; видеоматериалы с субтитрами; индивидуальные консультации с привлечением сурдопереводчика; индивидуальные задания и консультации;

- для лиц с нарушениями опорно-двигательного аппарата: в печатной форме; в форме электронного документа; в форме аудиофайла; индивидуальные задания и консультации.

В случае необходимости использования в учебном процессе частично/исключительно дистанционных образовательных технологий, организация работы ППС с обучающимися с ОВЗ и инвалидами предусматривается в электронной информационно-образовательной среде с учетом их индивидуальных психофизических особенностей.

Разработчик	Доцент Кандидат физико-математических наук	Шевченко Татьяна Валентиновна
-------------	--	-------------------------------

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф – Рабочая программа дисциплины		

	Должность, ученая степень, звание	ФИО
--	-----------------------------------	-----