

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



**УТВЕРЖДЕНО**

решением Ученого совета факультета математики, информационных и авиационных технологий  
 «21» 05 2024г., протокол № 5/24  
 Председатель \_\_\_\_\_ Волков М.А.  
 «21» 05 2024 г.

## РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Дисциплина	<b>Технологии и методы программирования</b>
Факультет	Факультет математики, информационных и авиационных технологий
Кафедра	Кафедра информационных технологий
Курс	3 - очная форма обучения

Направление (специальность): 10.05.03 Информационная безопасность автоматизированных систем

Направленность (профиль/специализация): Безопасность открытых информационных систем

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
Шабалин Александр Станиславович	Кафедра информационных технологий	Доцент, Кандидат физико-математических наук

## 1. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

### Цели освоения дисциплины:

- ввести в круг понятий и задач информационной обработки данных, связанных с проблемами сбора, передачи, обработки и накопления информации с помощью компьютеров;
- формирование у студентов научных основ методологии курса программирования.
- освоение базовых основ информатизации и программирования.
- изучение программных методов защиты данных.

### Задачи освоения дисциплины:

- получение и использование навыков работы с техническими и программными средствами для реализации информационных процессов;
- получение навыков обработки текстовой и числовой информации;
- получение навыков защиты информации.

## 2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина «Технологии и методы программирования» относится к числу дисциплин блока Б1.О.1, предназначенного для студентов, обучающихся по направлению: 10.05.03 Информационная безопасность автоматизированных систем.

В процессе изучения дисциплины формируются компетенции: ОПК-7, ОПК-11.

Основные положения дисциплины используются в дальнейшем при изучении таких дисциплин как: Научно-исследовательская работа, Подготовка к сдаче и сдача государственного экзамена, Языки программирования, Математическая логика и теория алгоритмов, Системы управления базами данных, Операционные системы, Базы данных.

## 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
ОПК-11 Способен разрабатывать компоненты систем защиты информации автоматизированных систем;	<p><b>знать:</b> содержание и способы использования компьютерных и информационных технологий;</p> <p><b>уметь:</b> применять и модифицировать математические модели для решения задач в области профессиональной деятельности.</p> <p><b>владеть:</b> компьютерными программами проектирования</p>

Код и наименование реализуемой компетенции	Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с индикаторами достижения компетенций
	продукции, методами сбора и обработки информации в рамках производственных задач
ОПК-7 Способен создавать программы на языках общего назначения, применять методы и инструментальные средства программирования для решения профессиональных задач, осуществлять обоснованный выбор инструментария программирования и способов организации программ;	<p><b>знать:</b> содержание и способы использования компьютерных и информационных технологий;</p> <p><b>уметь:</b> применять и модифицировать математические модели для решения задач в области профессиональной деятельности.</p> <p><b>владеть:</b> компьютерными программами проектирования продукции, методами сбора и обработки информации в рамках производственных задач</p>

#### 4. ОБЩАЯ ТРУДОЕМКОСТЬ ДИСЦИПЛИНЫ

4.1. Объем дисциплины в зачетных единицах (всего): 4 ЗЕТ

4.2. Объем дисциплины по видам учебной работы (в часах): 144 часа

Форма обучения: очная

Вид учебной работы	Количество часов (форма обучения <u>очная</u> )	
	Всего по плану	В т.ч. по семестрам
		5
<b>1</b>	<b>2</b>	<b>3</b>
Контактная работа обучающихся с преподавателем в соответствии с УП	90	90
Аудиторные занятия:	90	90
Лекции	36	36
Семинары и практические занятия	18	18
Лабораторные работы, практикумы	36	36
Самостоятельная работа	18	18
Форма текущего контроля знаний и контроля самостоятельной работы: тестирование, контр. работа, коллоквиум, реферат и др. (не менее 2 видов)	Тестирование	Тестирование
Курсовая работа	-	-
Виды промежуточной аттестации (экзамен, зачет)	Экзамен (36)	Экзамен
Всего часов по дисциплине	144	144

### 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. Раздел 1. Современное программирование на C++</b>							
Тема 1.1. Введение в C++ и современное программирование	5	2	0	2	0	1	Тестирование
Тема 1.2. Строки в C++: работа с текстом	7	2	2	2	0	1	Тестирование
Тема 1.3. Контейнеры STL: обзор и использование	7	2	2	2	2	1	Тестирование
Тема 1.4. Векторы (vector): динамические массивы	7	2	2	2	2	1	Тестирование
Тема 1.5. Ассоциативные контейнеры: map и set	5	2	0	2	2	1	Тестирование
Тема 1.6. Итераторы: принцип работы и применение	7	2	2	2	2	1	Тестирование
Тема 1.7. Основы структур и классов в C++	7	2	2	2	2	1	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
Тема 1.8. Конструкторы и деструкторы	5	2	0	2	2	1	Тестирование
Тема 1.9. Наследование и полиморфизм	7	2	2	2	2	1	Тестирование
Тема 1.10. Шаблоны в C++	7	2	2	2	2	1	Тестирование
Тема 1.11. Современные подходы в STL	5	2	0	2	0	1	Тестирование
Тема 1.12. Ошибки и исключения в C++	7	2	2	2	2	1	Тестирование
Тема 1.13. Стандарты C++: C++11, C++14, C++17	5	2	0	2	2	1	Тестирование
Тема 1.14. Лямбда-функции и функциональное программирование в C++	7	2	2	2	2	1	Тестирование
Тема 1.15. Стандартная библиотека алгоритмов (STL)	6	2	0	2	2	2	Тестирование
Тема 1.16. Класс	6	2	0	2	0	2	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
string: расширенные возможности работы со строками							
Тема 1.17. Работа с файлами: чтение и запись	4	2	0	2	0	0	Тестирование
Тема 1.18. Проектирование и архитектура программ на С++	4	2	0	2	0	0	Тестирование
<b>Итого подлежит изучению</b>	108	36	18	36	24	18	

## 5. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

### Раздел 1. Раздел 1. Современное программирование на С++

#### Тема 1.1. Введение в С++ и современное программирование

Обзор языка С++, его особенности и преимущества. История языка и его использование в различных областях. Основные концепции объектно-ориентированного программирования (ООП).

#### Тема 1.2. Строки в С++: работа с текстом

Основы работы с классом string, методы для манипуляции строками, конкатенация, сравнение, поиск подстрок и форматирование текста.

#### Тема 1.3. Контейнеры STL: обзор и использование

Введение в стандартную библиотеку шаблонов (STL). Обзор различных типов контейнеров (последовательные и ассоциативные) и их применение.

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

## **Тема 1.4. Векторы (vector): динамические массивы**

Изучение контейнера vector: создание, добавление и удаление элементов, доступ к элементам, управление памятью и использование методов.

## **Тема 1.5. Ассоциативные контейнеры: map и set**

Подробное рассмотрение map и set: структура, операции вставки, поиска и удаления, работа с ключами и значениями.

## **Тема 1.6. Итераторы: принцип работы и применение**

Понимание итераторов как универсального механизма доступа к элементам контейнеров. Типы итераторов, их использование и преимущества.

## **Тема 1.7. Основы структур и классов в C++**

Введение в структуры и классы: определение, разница между ними, создание объектов, доступ к членам и инкапсуляция.

## **Тема 1.8. Конструкторы и деструкторы**

Роль конструкторов и деструкторов в управлении жизненным циклом объектов. Различные типы конструкторов и их использование.

## **Тема 1.9. Наследование и полиморфизм**

Основы наследования: создание иерархий классов, переопределение методов, полиморфизм и виртуальные функции.

## **Тема 1.10. Шаблоны в C++**

Введение в шаблоны: функции и классы с шаблонами, преимущества использования шаблонов для создания универсальных компонентов.

## **Тема 1.11. Современные подходы в STL**

Рассмотрение новых возможностей и улучшений в STL, доступных в стандартах C++11 и выше, включая расширенные функции и алгоритмы.

## **Тема 1.12. Ошибки и исключения в C++**

Обработка ошибок в C++: использование механизма исключений, создание собственных классов исключений, практика обработки ошибок в коде.

## **Тема 1.13. Стандарты C++: C++11, C++14, C++17**

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

Обзор новых функций и возможностей, введенных в последних стандартах C++. Применение новых возможностей в реальных задачах.

#### **Тема 1.14. Лямбда-функции и функциональное программирование в C++**

Изучение лямбда-функций: синтаксис, использование, преимущества. Основы функционального программирования и применение в C++.

#### **Тема 1.15. Стандартная библиотека алгоритмов (STL)**

Обзор стандартных алгоритмов: сортировка, поиск, модификация контейнеров. Как использовать алгоритмы для решения различных задач.

#### **Тема 1.16. Класс string: расширенные возможности работы со строками**

Глубокое изучение класса string: методы для обработки строк, работа с символами, регулярные выражения.

#### **Тема 1.17. Работа с файлами: чтение и запись**

Основы работы с файлами в C++: открытие, чтение, запись и закрытие файлов. Использование потоков для обработки данных.

#### **Тема 1.18. Проектирование и архитектура программ на C++**

Принципы проектирования программного обеспечения: модульность, тестирование, документация. Примеры архитектурных решений на C++.

### **6. ТЕМЫ ПРАКТИЧЕСКИХ И СЕМИНАРСКИХ ЗАНЯТИЙ**

#### **Тема 1.2. Строки в C++: работа с текстом**

Вопросы к теме:

Очная форма

Как создать и инициализировать строку в C++?


Какие методы класса string вы можете использовать для изменения содержимого строки?

Как можно сравнивать две строки и какие нюансы следует учитывать?

#### **Тема 2.3. Контейнеры STL: обзор и использование**

Вопросы к теме:



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

## Очная форма

В чем основные преимущества использования vector по сравнению с обычными массивами?  
Как динамически изменять размер вектора во время выполнения программы?  
Какие методы vector используются для добавления и удаления элементов?

### **Тема 3.4. Векторы (vector): динамические массивы**

Вопросы к теме:

#### Очная форма

В чем основные преимущества использования vector по сравнению с обычными массивами?  
Как динамически изменять размер вектора во время выполнения программы?  
Какие методы vector используются для добавления и удаления элементов?

### **Тема 4.6. Итераторы: принцип работы и применение**

Вопросы к теме:

#### Очная форма

Какова роль итераторов в STL и почему они важны?  
Какие типы итераторов существуют в C++ и как они различаются?  
Как можно использовать итераторы для обхода контейнеров?

### **Тема 5.7. Основы структур и классов в C++**

Вопросы к теме:

#### Очная форма

Какова основная разница между структурами и классами в C++?  
Как можно задать доступ к членам класса или структуры (публичный/приватный)?  
Какие методы вы можете создать в классе для управления объектами?

### **Тема 6.9. Наследование и полиморфизм**

Вопросы к теме:

#### Очная форма

Как определить базовый класс и производный класс в C++?  
Какие виды наследования существуют и в чем их отличия?  
Как реализовать полиморфизм с использованием наследования в C++?

### **Тема 7.10. Шаблоны в C++**

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

Вопросы к теме:

Очная форма

Что такое шаблоны в C++ и как они помогают в разработке?

Как объявить и использовать шаблон функции?

Как можно создать шаблонный класс и какие примеры его использования?

## **Тема 8.12. Ошибки и исключения в C++**

Вопросы к теме:

Очная форма

Какова структура обработки исключений в C++ (try, catch, throw)?

В каких случаях вы должны использовать обработку исключений?

Как создать собственный класс исключений и использовать его в коде?

## **Тема 9.14. Лямбда-функции и функциональное программирование в C++**

Вопросы к теме:

Очная форма

Что такое лямбда-функция и какова её основная структура?

Как вы можете передать лямбда-функцию в качестве аргумента другой функции?

В чем преимущества использования лямбда-функций по сравнению с обычными функциями?

## **7. ЛАБОРАТОРНЫЕ РАБОТЫ, ПРАКТИКУМЫ**

Изучение стандартной библиотеки шаблонов (STL) C++

Цели: Познакомиться с основными компонентами стандартной библиотеки шаблонов (STL), такими как контейнеры, алгоритмы и итераторы, и научиться их эффективно использовать для решения прикладных задач. В рамках работы студенты приобретут практические навыки использования STL для управления коллекциями данных, поиска, сортировки и изменения элементов с помощью встроенных механизмов библиотеки.

Содержание: Реализовать программу, использующую контейнеры vector, list, и map для хранения и обработки данных различного типа. Продемонстрировать использование алгоритмов сортировки и поиска для этих контейнеров. Лабораторная работа должна показать, как стандартизированные инструменты STL позволяют значительно упростить работу с динамическими структурами данных.

Результаты: Студенты должны продемонстрировать способность использовать различные контейнеры STL для хранения данных. Ожидается, что они создадут программу, которая правильно применяет алгоритмы поиска и сортировки, и обеспечит корректное управление элементами в контейнерах. В результате студенты смогут оценить преимущества использования STL для упрощения работы с данными.

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

Ссылка: <http://lib.ulsu.ru/ProtectedView/Book/ViewBook/6758>

### Контейнеры C++: Вектор, Map, Set

Цели: Научиться применять различные контейнеры STL, такие как `vector`, `map`, и `set`, для эффективного хранения и обработки данных. Разобраться с различиями в их поведении и применении в зависимости от задачи, например, когда важно хранить данные с уникальными ключами, динамически изменять размер коллекции или обеспечивать быстрый доступ по ключу.

Содержание: Реализовать систему хранения данных о пользователях с использованием контейнеров `vector`, `map`, и `set`. Необходимо продемонстрировать динамическое добавление пользователей, уникальное хранение идентификаторов, а также выполнение быстрых поисковых операций с использованием соответствующего контейнера.

Результаты: Студенты должны создать приложение, которое эффективно управляет коллекцией пользователей, используя контейнеры `vector`, `map` и `set`. Ожидается, что они смогут корректно добавлять, удалять и искать пользователей, а также продемонстрируют, как разные контейнеры оптимизируют различные операции. В результате студенты увидят, как выбор подходящего контейнера влияет на производительность приложения.

Ссылка: <http://lib.ulsu.ru/ProtectedView/Book/ViewBook/6758>

### Структуры и классы в C++

Цели: Изучить основные принципы объектно-ориентированного программирования на примере создания структур и классов в C++. В рамках работы студенты осvoят создание классов, конструкторов, методов и деструкторов, а также научатся работать с инкапсуляцией, наследованием и другими ключевыми концепциями ООП.

Содержание: Создать класс "Студент", содержащий поля для хранения имени, возраста и среднего балла, а также методы для вывода данных о студенте и расчета стипендии. Необходимо показать практическую пользу инкапсуляции, а также научиться правильно использовать конструкторы и деструкторы для управления объектами.

Результаты: Студенты должны разработать класс "Студент", который включает методы для управления данными и выполнения расчетов. Ожидается, что они смогут инкапсулировать данные, использовать конструкторы и деструкторы, а также применять принципы ООП для создания читаемого и поддерживаемого кода. Результатом будет демонстрация понимания принципов объектно-ориентированного программирования.

Ссылка: <http://lib.ulsu.ru/ProtectedView/Book/ViewBook/6758>

### Бинарный поиск и алгоритмы STL

Цели: Изучить алгоритмы STL на примере бинарного поиска и научиться использовать стандартные функции для поиска и сортировки данных. В ходе работы студенты осvoят принципы работы с отсортированными массивами и приобретут практические навыки использования алгоритмов поиска.

Содержание: Написать программу, которая применяет бинарный поиск для нахождения заданного элемента в отсортированном массиве. Использовать стандартную функцию `binary_search` и продемонстрировать, как алгоритмы STL позволяют реализовывать сложные задачи поиска с минимальными усилиями.

Результаты: Студенты должны реализовать программу, использующую бинарный поиск для нахождения элементов в отсортированном массиве. Ожидается, что они продемонстрируют понимание принципов работы бинарного поиска и алгоритмов STL. Результатом станет успешное выполнение поиска с минимальными временными затратами и демонстрация преимуществ применения стандартных алгоритмов.

Ссылка: <http://lib.ulsu.ru/ProtectedView/Book/ViewBook/6758>

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

## Итераторы в контейнерах STL

**Цели:** Изучить понятие итераторов и научиться их применять для эффективного обхода и изменения элементов контейнеров. В ходе работы студенты познакомятся с различными типами итераторов и научатся использовать их для различных операций, таких как добавление, удаление и модификация данных.

**Содержание:** Реализовать программу, которая использует итераторы для обхода контейнера list и выполнения операций добавления, изменения и удаления элементов. Необходимо продемонстрировать, как итераторы позволяют абстрагироваться от внутренней структуры данных и работать с контейнерами на высоком уровне.

**Результаты:** Студенты должны разработать программу, использующую итераторы для манипуляции данными в контейнере list. Ожидается, что они смогут продемонстрировать эффективный обход контейнера, а также добавление, удаление и модификацию элементов. Результатом будет понимание того, как итераторы упрощают работу с коллекциями и обеспечивают абстракцию при взаимодействии с контейнерами.

**Ссылка:** <http://lib.ulsu.ru/ProtectedView/Book/ViewBook/6758>

## Unit тестирование

**Цели:** Освоить методы Unit тестирования и научиться разрабатывать тесты для проверки корректности работы небольших фрагментов кода (функций, методов). В ходе работы студенты познакомятся с принципами тестирования и узнают, как Unit тесты помогают находить ошибки на ранних стадиях разработки.

**Содержание:** Создать несколько Unit тестов для проверки работы класса "Калькулятор", который выполняет основные математические операции. Использовать фреймворк для автоматизации тестов (например, Google Test). Лабораторная работа должна показать, как регулярное тестирование помогает поддерживать качество кода и уменьшать вероятность ошибок.

**Результаты:** Студенты должны создать и выполнить набор Unit тестов для класса "Калькулятор". Ожидается, что они смогут продемонстрировать правильность работы методов класса, выявить и исправить возможные ошибки. Результатом станет отчет о тестировании, который подтвердит, что все функции работают корректно, а также понимание важности автоматизированного тестирования в процессе разработки программного обеспечения.

**Ссылка:** <http://lib.ulsu.ru/ProtectedView/Book/ViewBook/6758>

## 8. ТЕМАТИКА КУРСОВЫХ, КОНТРОЛЬНЫХ РАБОТ, РЕФЕРАТОВ

Данный вид работы не предусмотрен УП.

## 9. ПЕРЕЧЕНЬ ВОПРОСОВ К ЭКЗАМЕНУ

1. Опишите основные особенности языка C++ и его преимущества по сравнению с другими языками программирования. Какие аспекты делают его популярным выбором для разработчиков?
2. Расскажите о истории языка C++. Как он развивался с момента своего создания и какие важные изменения были внесены в различные версии?
3. Объясните основные концепции объектно-ориентированного программирования (ООП). Как они реализуются в C++, и почему они важны для разработки программного обеспечения?
2. 1. Объясните, как работает класс string в C++. Какие основные методы предоставляются для манипуляции строками, и как они могут быть использованы на практике?
2. Как выполняется

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

конкатенация строк в C++ с использованием класса string? Приведите примеры кода, показывающие различные способы объединения строк.3.Опишите методы поиска подстрок в классе string. Как можно использовать эти методы для поиска и сравнения текстовых данных?

3. 1.Что такое стандартная библиотека шаблонов (STL), и как она улучшает работу с данными в C++? Объясните, как STL способствует модульности и переиспользованию кода.2.Перечислите и опишите основные типы контейнеров в STL. Каковы их особенности, и в каких ситуациях каждый из них следует использовать?3.Как можно использовать контейнеры STL для решения реальных задач? Приведите примеры задач, которые можно эффективно решить с помощью STL.

4. 1.Объясните, что такое контейнер vector в C++. Каковы его основные преимущества по сравнению с обычными массивами?2.Как осуществляется управление памятью в векторах? Объясните, как vector автоматически изменяет размер при добавлении или удалении элементов.3.Приведите примеры использования методов push\_back(), pop\_back() и erase() для работы с векторами. Как они помогают в управлении динамическими массивами?

5. 1.Объясните, как работают ассоциативные контейнеры map и set. Как они организуют свои элементы, и чем они отличаются друг от друга?2.Как выполняются операции вставки и поиска в контейнерах map и set? Приведите примеры кода, демонстрирующие эти операции.3.Каковы преимущества использования ассоциативных контейнеров по сравнению с последовательными контейнерами, такими как vector или list? В каких случаях стоит использовать map и set?

6. 1.Что такое итераторы в C++, и какую роль они играют в стандартной библиотеке? Как они обеспечивают единообразный доступ к элементам контейнеров?2.Опишите различные типы итераторов, доступные в C++. Каковы их основные характеристики и различия?3.Как использовать итераторы для обхода контейнеров в C++? Приведите примеры кода, показывающие, как итераторы могут быть использованы в сочетании с алгоритмами STL.

7. 1.Объясните разницу между структурами и классами в C++. Какие особенности и ограничения есть у каждой из них?2.Как создаются и используются объекты классов в C++? Приведите пример определения класса и создания объектов, а также объясните доступ к членам класса.3.Как инкапсуляция реализуется в C++? Почему она важна для проектирования программного обеспечения, и как она помогает скрыть внутренние детали реализации?

8. 1.Какова роль конструкторов и деструкторов в управлении жизненным циклом объектов в C++? Объясните, как они используются при создании и уничтожении объектов.2.Опишите различные типы конструкторов, такие как конструкторы копирования и перемещения. Как они помогают управлять ресурсами в C++?3.Приведите пример класса, демонстрирующего использование конструктора и деструктора. Как они взаимодействуют при создании и удалении объектов?

9. 1.Объясните, что такое наследование в C++ и как оно используется для создания иерархий классов. Каковы основные преимущества наследования?2.Что такое полиморфизм, и как он реализуется в C++? Приведите пример, показывающий, как полиморфизм может быть использован для обработки объектов различных классов.3.Каковы особенности виртуальных функций и абстрактных классов в C++? Как они помогают в реализации полиморфизма?

10. 1.Что такое шаблоны в C++, и как они позволяют создавать универсальные функции и классы? Объясните, как шаблоны способствуют переиспользованию кода.2.Приведите пример шаблона функции, который демонстрирует, как можно обрабатывать различные типы данных. Как компилятор генерирует код для конкретных типов?3.Каковы преимущества использования шаблонов в C++ по сравнению с обычными функциями и классами? В каких случаях их использование оправдано?

11. 1.Какие новые возможности были добавлены в стандартную библиотеку шаблонов (STL) в C++11 и выше? Как эти изменения повлияли на разработку программного

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

обеспечения?2.Объясните, как лямбда-функции изменили подход к написанию кода в C++. Как они упрощают работу с алгоритмами STL?3.Как новые алгоритмы и функции в STL, такие как `std::for_each`, `std::transform`, и `std::accumulate`, могут быть использованы для решения реальных задач? Приведите примеры.

12. 1.Как обрабатываются ошибки в C++ с помощью механизма исключений? Опишите, как использовать блоки `try`, `catch` и `throw` для управления исключениями.2.Как создать собственный класс исключений в C++? Объясните, как и когда следует использовать пользовательские исключения.3.Приведите пример кода, показывающий, как обрабатывать ошибки в программе. Как это влияет на надежность и устойчивость кода?

13. 1.Какие ключевые изменения и улучшения были внесены в стандарты C++11, C++14 и C++17? Как они повлияли на разработку программного обеспечения?2.Объясните, как новые функции, такие как `std::optional`, `std::variant`, и `std::any`, упрощают обработку данных в C++. Приведите примеры их использования.3.Каковы преимущества использования новых возможностей стандартов C++ в реальных задачах? Приведите примеры задач, которые можно решить с помощью этих возможностей.

14. 1.Объясните, что такое лямбда-функции в C++ и как они способствуют функциональному программированию. Каков их синтаксис и когда их следует использовать?2.Как лямбда-функции могут быть использованы с алгоритмами STL? Приведите примеры, демонстрирующие использование лямбда-функций для обработки контейнеров.3.Каковы преимущества использования лямбда-функций по сравнению с традиционными функциями? В каких ситуациях их использование оправдано?

15. 1.Объясните, как стандартная библиотека алгоритмов (STL) упрощает решение задач, связанных с обработкой данных. Какие типичные алгоритмы доступны в STL?2.Как использовать алгоритмы `sort()`, `find()`, и `copy()` для обработки контейнеров в C++? Приведите примеры кода, демонстрирующие их применение.3.Как алгоритмы STL могут быть комбинированы для решения более сложных задач? Приведите пример, где несколько алгоритмов используются в одной программе.

16. 1.Объясните, как класс `string` в C++ позволяет работать с текстовыми данными. Какие методы доступны для обработки строк и работы с символами?2.Как использовать регулярные выражения в сочетании с классом `string` для поиска и замены текстовых данных? Приведите примеры использования.3.Как выполнять операции форматирования текста с использованием класса `string`? Объясните, как это может быть полезно в реальных приложениях.

17. 1.Как открываются и закрываются файлы в C++? Опишите различные режимы открытия файлов и их использование при чтении и записи данных.2.Как использовать потоки для чтения и записи текстовых и бинарных файлов в C++? Приведите примеры кода, демонстрирующие работу с потоками.3.Как обработка ошибок может быть реализована при работе с файлами в C++? Объясните, как это влияет на надежность программы.

18. 1.Какие принципы проектирования программного обеспечения важны при разработке программ на C++? Объясните, как они помогают улучшить качество кода.2.Как модульность и тестирование влияют на архитектуру программ на C++? Приведите примеры, показывающие, как эти принципы реализуются на практике.3.Объясните, как документация помогает в разработке и поддержке программного обеспечения на C++. Какие инструменты и методы могут быть использованы для создания документации?

## 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

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.)
<b>Раздел 1. Раздел 1. Современное программирование на C++</b>			
Тема 1.1. Введение в C++ и современное программирование	Проработка учебного материала с использованием ресурсов учебно-методического и информационного обеспечения дисциплины.	1	Вопросы к экзамену, Тестирование
Тема 1.2. Строки в C++: работа с текстом	Проработка учебного материала с использованием ресурсов учебно-методического и информационного обеспечения дисциплины.	1	Вопросы к экзамену, Тестирование
Тема 1.3. Контейнеры STL: обзор и использование	Проработка учебного материала с использованием ресурсов учебно-методического и информационного обеспечения дисциплины.	1	Вопросы к экзамену, Тестирование
Тема 1.4. Векторы (vector): динамические массивы	Проработка учебного материала с использованием ресурсов учебно-методического и информационного обеспечения дисциплины.	1	Вопросы к экзамену, Тестирование
Тема 1.5. Ассоциативные контейнеры: map и set	Проработка учебного материала с использованием ресурсов учебно-методического и информационного обеспечения дисциплины.	1	Вопросы к экзамену, Тестирование
Тема 1.6. Итераторы: принцип работы и применение	Проработка учебного материала с использованием ресурсов учебно-методического и информационного обеспечения дисциплины.	1	Вопросы к экзамену, Тестирование
Тема 1.7. Основы структур и	Проработка учебного материала с	1	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.)
классов в C++	использованием ресурсов учебно-методического и информационного обеспечения дисциплины.		Тестирование
Тема 1.8. Конструкторы и деструкторы	Проработка учебного материала с использованием ресурсов учебно-методического и информационного обеспечения дисциплины.	1	Вопросы к экзамену, Тестирование
Тема 1.9. Наследование и полиморфизм	Проработка учебного материала с использованием ресурсов учебно-методического и информационного обеспечения дисциплины.	1	Вопросы к экзамену, Тестирование
Тема 1.10. Шаблоны в C++	Проработка учебного материала с использованием ресурсов учебно-методического и информационного обеспечения дисциплины.	1	Вопросы к экзамену, Тестирование
Тема 1.11. Современные подходы в STL	Проработка учебного материала с использованием ресурсов учебно-методического и информационного обеспечения дисциплины.	1	Вопросы к экзамену, Тестирование
Тема 1.12. Ошибки и исключения в C++	Проработка учебного материала с использованием ресурсов учебно-методического и информационного обеспечения дисциплины.	1	Вопросы к экзамену, Тестирование
Тема 1.13. Стандарты C++: C++11, C++14, C++17	Проработка учебного материала с использованием ресурсов учебно-методического и информационного обеспечения дисциплины.	1	Вопросы к экзамену, Тестирование
Тема 1.14. Лямбда-функции и функциональное программирование в C++	Проработка учебного материала с использованием ресурсов учебно-методического и информационного обеспечения дисциплины.	1	Вопросы к экзамену, Тестирование
Тема 1.15. Стандартная библиотека алгоритмов (STL)	Проработка учебного материала с использованием ресурсов учебно-методического и информационного обеспечения дисциплины.	2	Вопросы к экзамену, Тестирование
Тема 1.16. Класс string:	Проработка учебного материала с	2	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.)
расширенные возможности работы со строками	использованием ресурсов учебно-методического и информационного обеспечения дисциплины.		Тестирование

## 11. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

### а) Список рекомендуемой литературы

#### основная

1. Терехов, А. Н. Технология программирования : учебное пособие / А. Н. Терехов ; А. Н. Терехов. - Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. - 148 с. - Книга находится в премиум-версии ЭБС IPR BOOKS. - Текст. - Гарантированный срок размещения в ЭБС до 28.07.2022 (автопродлонгация). - электронный. - Электрон. дан. (1 файл). - URL: <http://www.iprbookshop.ru/97587.html>. - Режим доступа: ЭБС IPR BOOKS; для авторизир. пользователей. - ISBN 978-5-4497-0702-4. / .— ISBN 0\_155104

2. Львович, И. Я. Информационные технологии моделирования и оптимизации. Краткая теория и приложения : монография / И. Я. Львович, Я. Е. Львович, В. Н. Фролов ; И. Я. Львович, Я. Е. Львович, В. Н. Фролов. - Воронеж : Воронежский институт высоких технологий, Научная книга, 2016. - 444 с. - Книга находится в премиум-версии ЭБС IPR BOOKS. - Текст. - Весь срок охраны авторского права. - электронный. - Электрон. дан. (1 файл). - URL: <http://www.iprbookshop.ru/67365.html>. - Режим доступа: ЭБС IPR BOOKS; для авторизир. пользователей. - ISBN 978-5-4446-0836-4. / .— ISBN 0\_140044

#### дополнительная

1. Самарский Александр Андреевич. Математическое моделирование : Идеи. Методы. Примеры / А.А. Самарский, А.П. Михайлов. - 2-е изд.,испр. - Москва : Физматлит, 2002. - 320 с. - ISBN 5-9221-0120-X (В пер.). / .— ISBN 1\_78334

2. Зайцев, М. Г. Современные технологии программирования : практикум / М. Г. Зайцев ; М. Г. Зайцев. - Новосибирск : Сибирский государственный университет телекоммуникаций и информатики, 2008. - 31 с. - Книга находится в премиум-версии ЭБС IPR BOOKS. - Текст. - Гарантированный срок размещения в ЭБС до 14.10.2021 (автопродлонгация). - электронный. - Электрон. дан. (1 файл). - URL: <http://www.iprbookshop.ru/55460.html>. - Режим доступа: ЭБС IPR BOOKS; для авторизир. пользователей. - ISBN 2227-8397. / .— ISBN 0\_134846

#### учебно-методическая

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

1. Шабалин А. С. Методические указания для самостоятельной работы студентов по дисциплине «Технологии и методы программирования» по направлению специалитета 10.05.03 «Информационная безопасность» / А. С. Шабалин ; УлГУ, ФМИиАТ. - 2019. - Загл. с экрана. - Неопубликованный ресурс. - Электрон. текстовые дан. (1 файл : 301 КБ). - Режим доступа: ЭБС УлГУ. - Текст : электронный. / .— ISBN 0\_41602.

## **б) Программное обеспечение**

- Операционная система "Альт образование"
- Офисный пакет "Мой офис"

## **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].

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

**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: <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;

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

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	Шабалин Александр Станиславович
	Должность, ученая степень, звание	ФИО