


Министерство науки и высшего образования Российской Федерации Ульяновский государственный университет	Форма	
Ф-Рабочая программа по дисциплине		



**УТВЕРЖДЕНО**

решением Ученого совета ФМИАТ  
от «16» мая 2023 г., протокол № 4/23

Приседатель \_\_\_\_\_ Волков М.А.  
*подпись, расшифровка подписи*

«16» мая 2023 г.

## РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Дисциплина	Языки программирования
Факультет	Математики, информационных и авиационных технологий
Кафедра	Информационной безопасности и теории управления
Курс	1-2

Направление (специальность): 10.05.03 «Информационная безопасность автоматизированных систем»

*код направления (специальности), полное наименование*

Направленность (профиль/специализация): «Безопасность открытых информационных систем»

*полное наименование*

Форма обучения: очная

*очная, заочная, очно-заочная (указать только те, которые реализуются)*

Дата введения в учебный процесс УлГУ: «01» сентября 2023 г.

Программа актуализирована на заседании кафедры: протокол № 12 от апреля 2023 г.

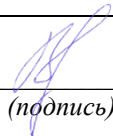

Программа актуализирована на заседании кафедры: протокол № \_\_\_\_\_ от \_\_\_\_\_ 20\_\_\_\_ г.


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
Цыганова Юлия Владимировна	ИТ	профессор, д.ф-м.н, доцент

СОГЛАСОВАНО	СОГЛАСОВАНО
Заведующий кафедрой, реализующей дисциплину	Заведующий выпускающей кафедрой
/  / <u>Волков М.А.</u> / <i>(подпись)</i> <i>(Ф.И.О.)</i>	/  / <u>Андреев А.С.</u> / <i>(подпись)</i> <i>(Ф.И.О.)</i>
«15» мая 2023 г.	«15» мая 2023 г.

Министерство науки и высшего образования Российской Федерации Ульяновский государственный университет	Форма	
Ф-Рабочая программа по дисциплине		

## 1. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Дисциплина «Языки и методы программирования» знакомит студентов с базовыми методами программирования и принципами разработки и реализации языков программирования и непосредственно связана с курсами «Информатика», «Объектно-ориентированное программирование» и «Технологии и методы программирования».

Предметом изучения являются языки и методы программирования, а также общие свойства языков программирования, методы анализа и разработки новых языков программирования, различные подходы к программированию, парадигмы программирования, методы трансляции языков программирования.

### Цели освоения дисциплины:

- изучение принципов базового и объектно-ориентированного программирования на языках C/C++;
- овладение навыками правильного программирования;
- получение студентами фундаментальных и практических знаний по теории программирования, методам программирования и трансляции программ.

### Задачи освоения дисциплины:

- выработка у студентов навыков использования языков программирования для создания систем обработки данных;
- обоснованного выбора методов и средств программирования;
- понимания механизмов трансляции программ.

## 2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина относится к базовой части цикла Б1 образовательной программы и читается в 5-м и в 6-м семестрах студентам специальности «Информационная безопасность автоматизированных систем» очной формы обучения.


Данная дисциплина базируется на входных знаниях, умениях, навыках и компетенциях студента, полученных им при изучении предшествующих учебных дисциплин, указанных в Приложении к данной рабочей программе (в фондах оценочных средств – далее ФОС, пункт 1).

Результаты освоения дисциплины будут необходимы для дальнейшего процесса обучения в рамках поэтапного формирования компетенций при изучении последующих дисциплин (указаны в ФОС, пункт 1), а также для прохождения всех видов практик и государственной итоговой аттестации.

## 3. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ (МОДУЛЮ), СООТНЕСЕННЫХ С ПЛАНИРУЕМЫМИ РЕЗУЛЬТАТАМИ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Процесс изучения дисциплины «Языки программирования» направлен на формирование следующих компетенций.

Код и наименование реализуемой компетенции	Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с индикаторами достижения компетенций
ОПК-7 – Способен создавать программы на языках общего назначения, применять методы и инструментальные средства програм-	Знать: методы создания программ на языках общего назначения, методы и инструментальные средства программирования для решения профессиональных задач. Уметь:

Министерство науки и высшего образования Российской Федерации Ульяновский государственный университет	Форма	
Ф-Рабочая программа по дисциплине		


мирования для решения профессиональных задач, осуществлять обоснованный выбор инструментария программирования и способов организации программ.	создавать программы на языках общего назначения, применять методы и инструментальные средства программирования для решения профессиональных задач, обоснованно выбирать инструментарий программирования и способы организации программ. Владеть: навыками создания программ на языках общего назначения, применения методов и инструментальных средств программирования для решения профессиональных задач, навыками обоснованного выбора инструментария программирования и способов организации программ.
--	--

#### 4. ОБЩАЯ ТРУДОЕМКОСТЬ ДИСЦИПЛИНЫ

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

4.2. Объем дисциплины по видам учебной работы:

Вид учебной работы	Количество часов (форма обучения – дневная)			
	Всего по плану	В т.ч. по семестрам		
		5	6	
Контактная работа обучающихся с преподавателем	144	90/90*	54/54*	
Аудиторные занятия:				
• Лекции	54	36/36*	18/18*	
• Практические и семинарские занятия	18	18/18*		
• Лабораторные работы (лабораторный практикум)	72	36/36*	36/36*	
Самостоятельная работа	62	26	36	
Экзамен	72	36/36*	36/36*	
Всего часов по дисциплине	216	108	108	
Форма текущего контроля знаний и контроля самостоятельной работы		Лабораторные работы, проверка решения задач	Лабораторные работы, проверка решения задач	
Виды промежуточной аттестации (экзамен, зачет)		зачёт	экзамен	
Общая трудоемкость в зач. ед.	8	3	3	


Министерство науки и высшего образования Российской Федерации Ульяновский государственный университет	Форма	
Ф-Рабочая программа по дисциплине		

*\*В случае необходимости использования в учебном процессе частично/исключительно дистанционных образовательных технологий в таблице через слеш указывается количество часов работы ППС с обучающимися при проведении занятий в дистанционном формате с применением электронного обучения.*


#### 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	
<b>Раздел 1. Основы программирования на языке C++</b>							
1.1. Выражения и операторы языка C++	6	2	1	1	1	1	Лабораторные работы. Домашние задания
1.2. Управление ходом выполнения программы	7	2	1	2	1	1	Лабораторные работы. Домашние задания
1.3. Функции	6	2	1	1	1	1	Лабораторные работы. Домашние задания
1.4. Ввод/вывод данных в C++	6	2	1	1	1	1	Лабораторные работы. Домашние задания
1.5. Введение в объектно-ориентированное программирование	7	2	1	2	1	1	Лабораторные работы. Домашние задания
1.6. Применение шаблонов C++	6	2	1	1	1	1	Лабораторные работы. Домашние задания
1.7. Исключения в C++	6	2	1	1	1	1	Лабораторные работы. Домашние задания
<b>Раздел 2. Основные концепции языков программирования</b>							
2.1. История	4	2	-	-	-	1	Проработка

Министерство науки и высшего образования Российской Федерации Ульяновский государственный университет	Форма	
Ф-Рабочая программа по дисциплине		

развития языков программирования							теоретического материала
2.2. Парадигмы языков программирования	3	1	-	1	-	1	Лабораторная работа
2.3. Критерии оценки языков программирования	4	2	1	-	1	1	Проработка теоретического материала
2.4. Объекты данных в языках программирования	5	2	1	1	1	1	Лабораторная работа
2.5. Механизмы типизации	4	2	1	-	1	1	Проработка теоретического материала
2.6. Время жизни переменных	4	1	1	1	1	1	Лабораторная работа
2.7. Область видимости переменных	3	1	1	-	1	1	Проработка теоретического материала
2.8. Типы данных	6	2	1	2	1	1	Лабораторная работа
2.9. Выражения и операторы присваивания	4	2	1	-	1	1	Проработка теоретического материала
2.10. Структуры управления на уровне операторов	6	2	1	2	1	1	Лабораторная работа
2.11. Подпрограммы	3	1	1	-	1	1	Проработка теоретического материала
Зачёт							Экзамен
Итого	108	36	18	36	16	18	
<b>Раздел 3. Методы трансляции программ</b>							
3.1. Общие сведения о трансляторах	3	2	1			1	Проработка теоретического материала
3.2. Основы теории языков и формальных грамматик	8	2	1	9	4	1	Проработка теоретического материала. Лабораторная работа
3.3. Распознаватели	3	4	2			2	Проработка теоретического материала
3.4. Лексический	8	4	2	9	4	2	Проработка

Министерство науки и высшего образования Российской Федерации Ульяновский государственный университет	Форма	
Ф-Рабочая программа по дисциплине		

анализ							теоретического материала. Лабораторная работа
3.5. Синтаксический анализ	8	4	2	9	6	2	Проработка теоретического материала. Лабораторная работа
3.6. Организация таблиц идентификаторов	3	4	2			2	Проработка теоретического материала
3.7. Семантический анализ и подготовка к генерации кода	4	4	2			2	Проработка теоретического материала
3.8. Распределение памяти	3	4	22			2	Проработка теоретического материала
3.9. Генерация кода. Методы генерации кода	8	4	2	9	4	2	Проработка теоретического материала. Лабораторная работа
3.10. Оптимизация кода. Основные методы оптимизации.	6	4	2			2	Проработка теоретического материала
Экзамен	36						Экзамен
Всего	216	54	18	72	34	36	

## 5. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

### *Раздел 1. Основы программирования на языке C++*

#### **Тема 1.1. Выражения и операторы языка C++.**


Структура программы на языке C. Комментарии и пробелы. Идентификаторы. Ключевые слова. Базовые типы данных. Константы. Строки. Указатели. Проблема двойного указания. Допустимые операции над указателями. Составные типы данных. Перечисления. Массивы. Структуры. Объединения. Объявление typedef. Арифметические операции. Операции отношения. Логические операции. Операции сдвига. Поразрядные логические операции. Операции присваивания. Операции над указателями. Операции над структурами и объединениями. Дополнительные операции. Преобразование типов данных. Приоритеты и порядок выполнения операций.

#### **Тема 1.2. Управление ходом выполнения программы.**

Оператор while. Оператор do while. Оператор for. Переходы в программе. Выбор вариантов. Оператор switch.

#### **Тема 1.3. Функции.**

Функции, вызов функции. Объявление функции. Определение функции. Инструкция return. Прототипы функций. Аргументы по умолчанию. Область видимости и классы па-

Министерство науки и высшего образования Российской Федерации Ульяновский государственный университет	Форма	
Ф-Рабочая программа по дисциплине		

мьюти (auto, register, extern, static, volatile). Пространства имен. Применение void. Передача массивов функциям.

#### **Тема 1.4. Ввод/вывод данных в C++.**

Основные понятия. Классификация функций ввода/вывода верхнего уровня. Обработка ошибок ввода/вывода. Открытие, закрытие и удаление файлов. Бесформатный ввод/вывод данных текстовых файлов. Форматный ввод/вывод данных текстовых файлов. Ввод/вывод записей. Анализ положения и управление положением указателя файла.

#### **Тема 1.5. Введение в объектно-ориентированное программирование.**

Определение класса. Создание экземпляра класса. Доступ к членам класса. Преимущества инкапсуляции. Классы с конструкторами (конструктор по умолчанию, инициализация конструктора, конструкторы как преобразователи). Классы с деструкторами. Вызов конструкторов и деструкторов. Функции-компоненты inline. Организация исходных файлов. Указатель this. Статические классы-члены. Производные классы. Конструкторы производных классов. Доступ к наследуемым членам. Создание иерархии классов. Преимущества наследования. Применение виртуальных функций. Применение виртуальных функций для управляемых объектов классов. Применение виртуальных функций для модификации базовых классов. Перегружаемые операторы. Определение дополнительных функций операторов. Перегрузка операторов. Перегрузка оператора присваивания. Применение конструкторов копирования и преобразования. Инициализация массивов.

#### **Тема 1.6. Применение шаблонов C++.**

Определение шаблонов функций. Замещение шаблонов. Определение шаблонов классов. Создание объектов по шаблонам. Включение конструкторов в шаблон функции.

#### **Тема 1.7. Исключения в C++.**

Обработка инициированных исключений. Составление кодов захвата (Catch-блоки). Размещение обработчиков исключений. Обработчики вложенных исключений. Обработка Win32 исключений.

### ***Раздел 2. Основные концепции языков программирования***

#### **Тема 2.1. История развития языков программирования.**

Основные этапы развития языков программирования. Классификация языков программирования. Системы программирования. Технологии программирования. Основные понятия языка программирования (алфавит, синтаксис, семантика).

#### **Тема 2.2. Парадигмы языков программирования.**

Императивные языки. Языки функционального программирования. Декларативные языки. Объектно-ориентированные языки.

#### **Тема 2.3. Критерии оценки языков программирования.**

Понятность. Надежность. Гибкость. Простота. Естественность. Мобильность. Стоимость.

#### **Тема 2.4. Объекты данных в языках программирования.**

Имена. Константы. Переменные.

#### **Тема 2.5. Механизмы типизации.**

Статические и динамические типы данных. Слабая типизация. Строгая типизация. Производные типы. Эквивалентность типов. Наследование атрибутов. Ограничения. Подтипы. Анонимные типы и подтипы.


#### **Тема 2.6. Время жизни переменных.**

#### **Тема 2.7. Область видимости переменных.**

#### **Тема 2.8. Типы данных.**

Элементарные типы данных. Символьные строки. Перечислимые типы. Ограниченные типы. Векторы и массивы. Записи. Объединения. Множества. Списки.

#### **Тема 2.9. Выражения и операторы присваивания.**

Министерство науки и высшего образования Российской Федерации Ульяновский государственный университет	Форма	
Ф-Рабочая программа по дисциплине		

Арифметические выражения. Логические выражения. Операторы присваивания.

### **Тема 2.10. Структуры управления на уровне операторов.**

Составные операторы. Условные операторы. Операторы цикла.

### **Тема 2.11. Подпрограммы.**

Определение подпрограммы. Формальные и фактические параметры подпрограммы. Процедуры и функции. Методы передачи параметров. Сопрограммы.

## **Раздел 3. Методы трансляции программ**

### **Тема 3.1. Общие сведения о трансляторах.**

Основные понятия и определения. Общие особенности языков программирования и трансляторов. Обобщенная структура транслятора. Варианты взаимодействия блоков транслятора.

### **Тема 3.2. Основы теории языков и формальных грамматик.**

Способы определения языков. Алфавиты, цепочки и языки. Проблема представления языков. Формальные грамматики. Отношения вывода. Примеры грамматик. Типы грамматик и их свойства. Классификация Хомского. Определения формального языка. Способы записи синтаксиса языка. Метаязык Хомского. Метаязык БНФ. Метаязык РБНФ. Диаграммы Вирта. Принцип рекурсии в правилах грамматики.

### **Тема 3.3. Распознаватели.**

Определение распознавателя. Структура распознавателя. Виды распознавателей. Задача разбора. Классификация распознавателей. Сентенциальная форма грамматики. Левосторонний и правосторонний выводы. Дерево вывода. Методы построения дерева вывода. Проблема однозначности и эквивалентности грамматик. Проверка однозначности и эквивалентности грамматик. Правила, задающие неоднозначность в грамматиках.

### **Тема 3.4. Лексический анализ.**

Понятие лексемы. Сканер. Транслитератор. Классы символов. Грамматики и распознаватели для лексического анализа. Связь между диаграммой Вирта и конечным автоматом. Минимизация коющего автомата. Связь между диаграммами Вирта и праволинейными грамматиками. Преобразование правой рекурсии в итерацию. Связь между диаграммами Вирта и грамматиками с левой рекурсией. Преобразование левой рекурсии в итерацию. Методы лексического анализа. Организация непрямого лексического анализатора. Организация прямого лексического анализатора. Практические вопросы построения лексических анализаторов.


### **Тема 3.5. Синтаксический анализ.**

Классификация методов синтаксического разбора. Необходимость использования автоматов с магазинной памятью. Организация автомата с магазинной памятью. Распознаватель скобочных выражений. Общая связь между грамматиками и автоматами с магазинной памятью. Связь между S-грамматикой и автоматом с магазинной памятью. Обобщенный алгоритм построения нисходящего АМП для S-грамматики. S-грамматика и распознавание вложенности скобок. Построение автомата с магазинной памятью по q-грамматике. Понятие множества выбора. Построение нисходящего автомата. Примеры построения АМП по q-грамматике. Распознавание вложенности скобок и q-грамматика. LL(1) – грамматики. Программная реализация нисходящего автомата с магазинной памятью. Разработка программы по таблице переходов АМП. Разработка программы с использованием метода рекурсивного спуска. Правила преобразования диаграмм Вирта в программные функции.

### **Тема 3.6. Организация таблиц идентификаторов.**

Назначение и особенности построения таблиц идентификаторов. Простейшие методы построения таблиц идентификаторов. Построение таблиц идентификаторов по методу



Министерство науки и высшего образования Российской Федерации Ульяновский государственный университет	Форма	
Ф-Рабочая программа по дисциплине		

бинарного дерева. Хэш-функции и хэш-адресация. Построение таблиц идентификаторов на основе хэш-функции. Построение таблиц идентификаторов по методу цепочек. Комбинированные способы построения таблиц идентификаторов.

### **Тема 3.7. Семантический анализ и подготовка к генерации кода.**

Назначение семантического анализа. Этапы семантического анализа. Проверка соблюдения во входной программе семантических соглашений. Дополнение внутреннего представления программы. Проверка смысловых норм языков программирования. Идентификация лексических единиц языков программирования.

### **Тема 3.8. Распределение памяти.**

Принципы распределения памяти. Виды переменных и областей памяти. Распределение памяти для переменных скалярных типов. Распределение памяти для сложных структур данных. Выравнивание границ областей памяти. Виды областей памяти. Статическое и динамическое связывание. Глобальная и локальная память. Статическая и динамическая память. Менеджеры памяти. Дисплей памяти процедуры (функции). Стековая организация дисплея памяти. Память для типов данных (RTTI-информация).

### **Тема 3.9. Генерация кода. Методы генерации кода.**

Общие принципы генерации кода. Синтаксически управляемый перевод. СУ-компиляция. Способы внутреннего представления программ. Синтаксические деревья. Преобразование дерева разбора в дерево операций. Многоадресный код с явно именуемым результатом (тетрады). Многоадресный код с неявно именуемым результатом (триады). Обратная польская запись операций. Вычисление выражений с помощью обратной польской записи. Схема СУ-компиляции для перевода выражений в обратную польскую запись. Схемы СУ-перевода. Пример схемы СУ-перевода дерева операций на язык ассемблера. Пример схемы СУ-перевода дерева операций в последовательность триад.

### **Тема 3.10. Оптимизация кода. Основные методы оптимизации.**

Общие принципы оптимизации кода. Оптимизация линейных участков программы. Удаление бесполезных присваиваний. Исключение избыточных вычислений (лишних операций). Свертка объектного кода. Перестановка операций. Арифметические преобразования. Оптимизация вычисления логических выражений. Оптимизация передачи параметров в процедуры и функции. Метод передачи параметров через регистры процессора. Метод подстановки кода функции в вызывающий объектный код. Оптимизация циклов. Машинно-зависимые методы оптимизации. Распределение регистров процессора. Оптимизация кода для процессоров, допускающих распараллеливание вычислений.

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

Практические занятия (семинары) предусматривают решение задач по тематике Раздела 1 и обсуждение найденных решений.

**Тема 1.** Общие сведения о языке C++.

**Тема 2.** Основные конструкции языка C++.

**Тема 3.** Обработка массивов на C++.

**Тема 4.** Указатели и динамические массивы.

**Тема 5.** Обработка матриц на C++.


**Тема 6.** Организация ввода-вывода в C++.

**Тема 7.** Обработка строк.

**Тема 8.** Структуры в C++.


**Тема 9.** Объектно-ориентированное программирование на C++.

## **7. ЛАБОРАТОРНЫЕ РАБОТЫ (ЛАБОРАТОРНЫЙ ПРАКТИКУМ)**

Министерство науки и высшего образования Российской Федерации Ульяновский государственный университет	Форма	
Ф-Рабочая программа по дисциплине		

## Раздел 1. Основы программирования на языке C++

<b>Темы лабораторных работ</b>
<p>Лабораторная работа № 1. Общие сведения о языке C++. Задание: Написать, отладить и протестировать программу на языке C++ по заданному варианту. Пример: Заданы два катета прямоугольного треугольника. Найти гипотенузу и углы треугольника.</p>
<p>Лабораторная работа № 2. Основные конструкции в языке C++. Задание: Написать, отладить и протестировать программу на языке C++ по заданному варианту. Пример: Вычислить сумму натуральных нечетных чисел, не превышающих N.</p>
<p>Лабораторная работа № 3. Обработка массивов в C++. Задание: Написать, отладить и протестировать программу на языке C++ по заданному варианту с обязательным применением массивов. Пример: Определить, содержит ли массив группы элементов, расположенные в порядке возрастания их значений. Если да, то определить количество таких групп.</p>
<p>Лабораторная работа № 4. Указатели и динамические массивы. Задание: Написать, отладить и протестировать программу на языке C++ по заданному варианту с обязательным применением динамических массивов и функций. Пример: Из положительных значений двух целочисленных массивов различной размерности сформировать общий массив. Найти среднее арифметическое элементов, расположенных в массиве после последнего совершенного числа. Создать функции для вычисления среднего значения и определения совершенного числа.</p>
<p>Лабораторная работа № 5. Обработка матриц в C++. Задание: Написать, отладить и протестировать программу на языке C++ по заданному варианту с обязательным применением динамических матриц. Пример: Решить СЛАУ <math>A(n,n)X=B(n)</math> (матрица A и вектор B заданы) методом Гаусса. Сделать проверку <math>AX=B</math>.</p>
<p>Лабораторная работа № 6. Организация ввода-вывода в C++. Задание: Написать, отладить и протестировать программу на языке C++ по заданному варианту. Первая программа должна сформировать двоичный файл. Вторая – считать данные из этого файла, выполнить соответствующие вычисления (при этом используемые массивы должны быть динамическими) и записать их результаты в текстовый файл. Пример: Создать двоичный файл и записать в него n целых чисел. Из исходного файла сформировать массивы четных и нечетных чисел. Определить наибольший четный компонент файла и наименьший нечетный.</p>
<p>Лабораторная работа № 7. Обработка строк. Задание: Написать, отладить и протестировать программу на языке C++ по заданному варианту. Дана строка до точки, группа символов в которой между пробелами считается словом, знаки препинания от слова пробелом не отделяются. Пример: Подсчитать количество знаков препинания в строке.</p>
<p>Лабораторная работа № 8. Структуры. Задание: Написать, отладить и протестировать программу на языке C++ по заданному варианту с обязательным использованием структур. В программе должна быть реализована система меню, включающая функции для создания двоичного файла с информацией, добавления туда новой информации, обработки информации и просмотра информации с выводом на экран. Пример: Создать двоичный файл с информацией о перенесенных инфекционных</p>

Министерство науки и высшего образования Российской Федерации Ульяновский государственный университет	Форма	
Ф-Рабочая программа по дисциплине		

заболеваниях учащимися средней школы с января по июнь. Добавить в файл поле (дополнительный столбец в таблице) «Средняя заболеваемость за полугодие». Упорядочить информацию в файле в порядке возрастания средней заболеваемости.

Лабораторная работа № 9. Объектно-ориентированное программирование на языке C++.


Задание: Написать, отладить и протестировать программу на языке C++ по заданному варианту с обязательным использованием классов. Программа должна состоять из двух файлов: один с описанием классов и методов (с обязательными конструкторами и деструкторами), а второй – с демонстрацией работы всех возможностей класса на примере конкретных объектов. Для всех вариантов предусмотреть разработку конструктора и деструктора.

Пример: Класс – комплексное число в алгебраической форме  $z=x+yi$ . Данные класса – действительная часть ( $x$ ) и мнимая часть ( $y$ ) числа. Методы – вычисление корня комплексного числа, вывод комплексного числа. Перегружаемые операции – сложение (+), вычитание (-), умножение (\*), деление (/) двух чисел.

Методические указания: основное внимание должно быть уделено освоению базовых типов данных и алгоритмов работы с ними (см. раздел 11, учебно-методическая литература).

## Раздел 2: Основные концепции языков программирования

Темы лабораторных работ
<p><i>Тема 1. Визуальное и объектно-ориентированное программирование. Наследование и виртуальные методы</i></p> <p>Задания лабораторных работ индивидуальны. Необходимо реализовать базовый класс с заданными свойствами и затем реализовать его потомка с расширенным набором методов и перекрытием некоторых методов в классе-наследнике. Классы должны быть реализованы в виде одного модуля. Все методы и их параметры, а так же элементы данных класса должны быть полностью прокомментированы. Кроме этого, должна быть реализована простая программа, иллюстрирующая работу с обоими классами. В описании задания буквой А) обозначается задание на базовый класс, а буквой Б) - на класс наследник, или класс, использующий класс А как элемент(ы) своих данных. Работа должна быть выполнена на языке Visual Си++ или C++ Builder. При реализации в отдельных вариантах удобно использовать промежуточные классы и множественное наследование. Наличие визуального интерфейса обязательно! Использование механизма стандартных потоков C++ обязательно для стандартных операций ввода/вывода, а также желательно образование и использование их наследников с требуемыми для Вашего задания свойствами.</p>
<p><i>Тема 2. Создание и обработка файлов с данными в Delphi</i></p> <p>Системы управления базами данных позволяют хранить большие массивы разнообразной информации, быстро находить необходимые данные, создавать отчеты. Для хранения и обработки информации в базах данных их размещают на дисках в виде текстовых и типизированных файлов. Для обработки данных созданы специальные языки запросов, позволяющие описать поисковые условия, не задавая последовательности действий, необходимых для получения ответа.</p> <p>ЗАДАНИЕ: Создать программу для работы с базой данных, назначение которой указано в Вашем варианте. Наличие визуального интерфейса обязательно! Основной блок, в режиме меню, должен запускать рабочие процедуры следующего</p>

Министерство науки и высшего образования Российской Федерации Ульяновский государственный университет	Форма	
Ф-Рабочая программа по дисциплине		

назначения:

1. Процедура Create.

Создание типизированного файла данных, элементами которого являются записи со структурой, указанной в Вашем варианте. Процедура запрашивает имя (Name) и создает файл с этим именем.

2. Процедура Use.

Выбор файла данных из текущего каталога по имени (Name). После выбора файла все остальные процедуры работают с этим файлом, который считается активным или текущим.

3. Процедура Append.

Добавление в файл новых записей в формате карточки-шаблона, в которой каждое поле записи имеет название и фиксированную позицию для ввода значений. После заполнения карточки вся запись добавляется в конец файла, а карточка-шаблон очищается от прежних значений и подготавливается для ввода следующей записи.

4. Процедура Browse.

Просмотр записей в форме таблицы базы данных с применением вертикального скроллинга и с возможностью досрочного окончания просмотра. Обязательна индикация номера записи.

5. Процедура Work\_A и Work\_B.

Обработка записей файла данных по заданию Вашего варианта и вывод результатов на экран.

6. Процедура Quit.

Выход из программы. Все открытые файлы закрываются.

Примечания:

1. При сдаче работы наряду с программой следует представить заполненные файлы данных, состоящие из не менее чем 10 записей, заполненных так, чтобы показать правильное функционирование Вашего алгоритма.

*Тема 3. Научное программирование в среде MATLAB.*

Написать и отладить программу на языке MATLAB в соответствии со своим вариантом.


Темы заданий:

1. Массивы и матрицы в MATLAB.
2. Построение двумерных графиков.
3. Построение трехмерных графиков.
4. Нелинейные уравнения и системы.
5. Обработка экспериментальных данных.
6. Решение задач оптимизации.

Методические указания: основное внимание должно быть уделено освоению алгоритмов работы с динамическими структурами данных (см. раздел 11, учебно-методическая литература).

### **Раздел 3: Методы трансляции программ**

<b>Темы лабораторных работ</b>
<p><i>Тема 1. Описание синтаксиса языка программирования с использованием Диаграмм Вирта.</i></p> <p>Цель работы: Изучение основ теории языков и формальных грамматик, метаязыков, методов описания пользовательского синтаксиса. Использование Диаграмм Вирта для описания синтаксиса языка программирования.</p>

Министерство науки и высшего образования Российской Федерации Ульяновский государственный университет	Форма	
Ф-Рабочая программа по дисциплине		

Результат: Диаграммы Вирта для заданного варианта синтаксических конструкций и 4 содержательных примера.

*Тема 2. Разработка лексического анализатора заданного языка программирования.*

Цель работы: Изучение основ теории языков и формальных грамматик, методов построения лексических анализаторов (или сканеров).

Результат: программный код лексического анализатора для заданного варианта синтаксических конструкций.

*Тема 3. Разработка синтаксического анализатора заданного языка программирования.*

Цель работы: Изучение основ теории языков и формальных грамматик, методов построения синтаксических анализаторов.

Результат: программный код синтаксического анализатора для заданного варианта синтаксических конструкций.

*Тема 4. Разработка интерпретатора заданного языка программирования.*

Цель работы: Изучение основ теории языков и формальных грамматик, методов построения интерпретаторов.

Результат: программный код интерпретатора для заданного варианта синтаксических конструкций.

Полное описание заданий для лабораторных работ и методические указания к ним содержится в учебно-методическом пособии [1] (см. раздел 11, учебно-методическая литература).

## 8. ТЕМАТИКА КУРСОВЫХ, КОНТРОЛЬНЫХ РАБОТ, РЕФЕРАТОВ

Курсовые работы и рефераты не предусмотрены учебным планом дисциплины.

**Комплект задач для текущего контроля и контроля самостоятельной работы обучающихся**


### Разделы 1 и 2.

№ задания	Условия задачи (формулировка задания)
1.	1) Туристы. За $A$ часов поездки на автомашине и $P$ часов на поезде туристы проехали $S$ км. Какова скорость поезда, если она на $N$ км/час больше скорости автомашины? 2) Стройка. На стройке работает $N$ бригад, причем на объекте $Dom1$ их в $K$ раз больше, чем на объекте $Dom2$ . Сколько их работает на каждом объекте? ( $K$ - целое число). 3) Автопарк. В автопарке находилось $N$ автобусов, причем неисправных было в $K$ раз меньше, чем исправных ( $K$ – целое число). Сколько автобусов вышло на линию?
2.	1) Распечатать все числа от 1 до $N$ , у которых остатки от деления на число $Z$ не превышают числа $M$ . 2) Распечатать квадраты только 3-значных чисел от 1 до $N$ . 3) Распечатать все числа от 1 до $N$ , не оканчивающиеся на цифру 3.
3.	1) Определить, есть ли равные цифры в числе. 2) Определить, является ли первая и последняя цифра числа одинаковой. 3) Найти наибольшую нечетную цифру. Если ее нет, вернуть 0.

4.	<p>1) Вычислить: <math>y = 1 + \frac{x \ln a}{1!} + \frac{(x \ln a)^2}{2!} + \frac{(x \ln a)^3}{3!} + \dots</math>; (N слагаемых).</p> <p>2) Вычислить: <math>y = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots</math>; (N слагаемых).</p> <p>3) Вычислить: <math>y = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots</math>; (N слагаемых).</p>
5.	<p>1) Удалить из массива числа, которые являются полным квадратом и имеют нечетную сумму цифр.</p> <p>2) Вместо чисел, имеющих в составе цифру P, вставить сумму цифр этого числа.</p> <p>3) Удалить из массива числа, имеющие не простую сумму цифр.</p>
6.	<p>1) Имеется ли в строке St изображение двузначного числа? Например: в строке 'mite37sim' имеется, а в 'wer673i' – нет.</p> <p>2) В строке St находится изображение целого числа, состоящее из цифр, например: 'abc12347654387de'. Напечатать это число, делая пробелы между триадами цифр, отделяя таким образом тысячи, миллионы и т.д. Пример: 12 347 654 387.</p> <p>3) Определить, имеется ли в строке St хотя бы две цифры и один из знаков '+', '-', '*', '/'. Пример: a7+b14=c8.</p>
7.	<p>Дана квадратная матрица целых чисел размером <math>N \leq 12</math>. Заполнить матрицу случайными числами от 0 до 100. Найти максимальное значение, среднее значение и количество тех чисел, которые расположены:</p> <p>1) выше главной и обратной диагонали;</p> <p>2) выше главной и ниже обратной диагонали;</p> <p>3) ниже главной и выше обратной диагонали;</p>
8.	<p>1) Дан текстовый файл с изображениями целых чисел. В выходной файл поместить те из них, которые имеют только четные цифры. Пример: 2462 8204 44 и т.д. На экран вывести количество чисел во входном файле, количество чисел в выходном файле.</p> <p>2) Дан текстовый файл с изображениями целых чисел. В выходной файл поместить те из них, у которых старшая цифра больше младшей. Пример: 3462 804 41 и т.д. На экран вывести количество чисел во входном файле, количество чисел в выходном файле.</p> <p>3) Дан текстовый файл с изображениями целых чисел. В выходной файл поместить те из них, которые состоят из возрастающих цифр. Пример: 246 12478 45 и т.д. На экран вывести количество чисел во входном файле, количество чисел в выходном файле.</p>

### Раздел 3.

№ задания	Условие задачи (формулировка задания)
1.	<p>Вычислите следующие выражения, записанные в форме обратной польской записи:</p> <p>1 2 * 3 5 4 * + +</p> <p>1 2 3 5 4 * * + +</p> <p>1 2 + 3 * 5 + 4 -</p> <p>1 2 + 3 * 5 4 + -</p> <p>Напишите для каждого из них выражение в форме инфиксной (обычной)</p>


Министерство науки и высшего образования Российской Федерации Ульяновский государственный университет	Форма	
Ф-Рабочая программа по дисциплине		

	записи.
2.	<p>Дана грамматика: <math>G = (\{“ ”, (, ), o, r, a, n, d, t, b, \}, \{S, T, E, F\}, P, S)</math></p> <p>P:</p> <p><math>S \rightarrow S \text{ or } T \mid T</math></p> <p><math>T \rightarrow T \text{ and } E \mid E</math></p> <p><math>E \rightarrow \text{not } E \mid F</math></p> <p><math>F \rightarrow (S) \mid b</math></p> <p>Постройте для нее диаграммы Вирта. Исключите левую рекурсию. Постройте диаграммы Вирта без рекурсии.</p>
3.	<p>Дана грамматика: <math>G = (\{“ ”, (, ), o, r, a, n, d, t, b, \}, \{S, T, E, F\}, P, S)</math></p> <p>P:</p> <p><math>S \rightarrow S \text{ or } T \mid T</math></p> <p><math>T \rightarrow T \text{ and } E \mid E</math></p> <p><math>E \rightarrow \text{not } E \mid F</math></p> <p><math>F \rightarrow (S) \mid b</math></p> <p>Постройте вывод цепочки “b or b and not b and (b or not not b)”. Постройте дерево разбора для этой цепочки.</p>
4.	<p>Дана грамматика: <math>G = (\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, -, +\}, \{S, T, F\}, P, S)</math></p> <p>P:</p> <p><math>S \rightarrow T \mid +T \mid -T</math></p> <p><math>T \rightarrow F \mid TF</math></p> <p><math>F \rightarrow 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9</math></p> <p>Запишите левосторонний вывод терминальной цепочки +3456. Запишите правосторонний вывод терминальной цепочки -7890. Запишите комбинированный вывод терминальной цепочки 12345.</p>
5.	<p>Дана грамматика <math>G = (\{(, ), ^, \&amp;, \sim, a\}, \{S, T, E\}, P, S)</math>:</p> <p>P:</p> <p><math>S \rightarrow S^{\wedge}T \mid T</math></p> <p><math>T \rightarrow T\&amp;E \mid E</math></p> <p><math>E \rightarrow \sim E \mid (S) \mid a</math></p> <p>Запишите левосторонний вывод терминальной цепочки, содержащей не менее семи символов. Постройте дерево разбора для этой цепочки.</p>

## 9.1 ПЕРЕЧЕНЬ ВОПРОСОВ К ЗАЧЁТУ

### Раздел 1: Основы программирования на языке C++

1. Опишите структуру программы на языке C++.
2. Назовите отличительные особенности языка C++ по сравнению с Паскалем.
3. Опишите базовые типы данных C++.
4. Опишите типы констант в C++.
5. Что такое указатель? Виды указателей. Приведите пример.
6. Перечислите допустимые операции над указателями. Приведите пример.
7. Перечисление в языке C++. Пример.
8. Приведите способы задания массивов в C++.
9. Приведите способы задания структуры в C++.
10. Объединение в языке C++. Пример.
11. Перечислите виды операций в C++. Приведите примеры.


Министерство науки и высшего образования Российской Федерации Ульяновский государственный университет	Форма	
Ф-Рабочая программа по дисциплине		

12. Правила преобразования типов данных в C++.
13. Структура оператора цикла while. Пример.
14. Структура оператора цикла do while. Пример.
15. Структура оператора цикла for. Пример.
16. Назовите операторы переходов в C++. Приведите примеры.
17. Структура и виды условного оператора в C++. Примеры.
18. Структура оператора множественного выбора в C++. Пример.
19. Дайте развернутое определение функции в C++.
20. Раскройте понятие класса памяти. Перечислите классы памяти в C++.
21. Опишите автоматический класс памяти.
22. Опишите регистровый класс памяти.
23. Опишите внешний класс памяти.
24. Опишите статический класс памяти.
25. Определение и назначение препроцессора. Что такое директива препроцессора?
26. Опишите особенности ввода/вывода данных в языке Си.
27. Дайте определение оператора расширения области видимости в C++. Приведите пример.
28. Определение и особенности встраиваемых функций в C++.
29. Опишите задание параметров функции по умолчанию.
30. Определение и особенности использования ссылок в C++.
31. Определение и особенности перегружаемых функций в C++.
32. Операторы работы с динамической памятью в C++. Пример.
33. Дайте развернутое определение понятия класса в C++.
34. Перечислите спецификации доступа к компонентам данных класса. Приведите пример.
35. Инкапсуляция в C++. Пример.
36. Дайте определение и назовите виды конструкторов в C++. Что такое деструктор?
37. Объясните назначение указателя this.
38. Дайте определение и объясните назначение производных классов в C++.
39. Объясните назначение и особенности создания иерархий классов.
40. Дайте определение и объясните отличие виртуальных функций от обычных функций C++.
41. Объясните механизм перегрузки операторов в C++. Приведите пример.
42. Опишите особенности перегрузки оператора присваивания.
43. Дайте определение и объясните назначение конструкторов копирования в C++.
44. Дайте определение и объясните назначение конструкторов преобразования в C++.
45. Дайте определение шаблона функции в C++. Приведите пример.
46. Определение шаблона класса в C++. Приведите пример.
47. Дайте определение и объясните назначение механизма исключений в C++.
48. Опишите способы составления кодов захвата в механизме исключений C++.
49. Объясните понятие раскручивания стека в механизме исключений C++.


## ***Раздел 2: Основные концепции языков программирования***

1. Какие языки называются императивными?
2. Какие языки относят к языкам функционального программирования?
3. Какие языки являются декларативными?
4. Назовите три основных свойства объектно-ориентированных языков программирования.
5. Какую пользу можно извлечь из знания свойств языка программирования?



Министерство науки и высшего образования Российской Федерации Ульяновский государственный университет	Форма	
Ф-Рабочая программа по дисциплине		

6. Как можно увеличить надежность языка программирования?
7. Как влияет удобочитаемость языка программирования на легкость создания программ на этом языке?
8. Что понимается под естественностью языка программирования?
9. Из чего складывается суммарная стоимость языка программирования?
10. Какое свойство языка программирования дает возможность более просто переносить программы с одной платформы на другие?
11. Что означает концептуальная целостность языка программирования?
12. Что понимается под объектом данных в языках программирования?
13. Чем отличается литерал от именованной константы?
14. С помощью каких атрибутов можно охарактеризовать переменную?
15. Что означает предопределенное имя?
16. Что такое неявное определение типа?
17. Когда осуществляется статическое связывание типа?
18. Назовите недостатки слабой типизации языка программирования.
19. Назовите основные признаки строгой типизации.
20. Каким образом производный тип наследует атрибуты?
21. Как определяется эквивалентность типов?
22. Чем подтип отличается от производного типа?
23. Что понимают под анонимным типом?
24. С какими проблемами связано использование указателей?
25. Что определяет область видимости переменных?
26. Из каких частей состоит среда ссылок?
27. Каким образом реализуется статическая область видимости имен?
28. Что понимается под динамической областью видимости имен?
29. Какие существуют разновидности числовых типов, и чем определяется их многообразие?
30. Какие операции определены для указателей?
31. Что содержит дескриптор вектора?
32. Как представляются в памяти многомерные массивы?
33. Какими атрибутами характеризуются записи?
34. Какие существуют способы представления множеств в памяти?
35. Каким образом разрешаются коллизии при использовании хеширования?
36. Какие операции над списками наиболее часто используются?
37. Чем определяется порядок вычисления операций в арифметических выражениях?
38. Что означает приведение типа?
39. Что такое перегруженный оператор?
40. Что означают сужающее и расширяющее преобразование типа?
41. Что представляет собой составной оператор присваивания?
42. Какие типы управляющих структур имеются в языках программирования?
43. Как реализуются правила интерпретации вложенных условных операторов?
44. Какие проблемы возникают при разработке операторов цикла с параметром?
45. Какие проблемы возникают при разработке операторов цикла с логическим управлением?
46. Как выполняется оператор цикла, управляемый пользователем?
47. Что представляет собой спецификация подпрограммы?
48. В каком отношении между собой находятся формальные и фактические параметры?
49. Какие существуют режимы и механизмы передачи параметров?
50. Перечислите основные достоинства и недостатки передачи параметров по ссылке.


Министерство науки и высшего образования Российской Федерации Ульяновский государственный университет	Форма	
Ф-Рабочая программа по дисциплине		

51. Перечислите основные достоинства и недостатки передачи параметров по значению, по результату.


## 9.2 ПЕРЕЧЕНЬ ВОПРОСОВ К ЭКЗАМЕНУ

### Раздел 3: Методы трансляции программ


1. Назовите отличия: интерпретатора от компилятора.
2. Назовите отличия: компилятора от ассемблера.
3. Назовите отличия: перекодировщика от транслятора.
4. Назовите отличия: эмулятора от интерпретатора.
5. Назовите отличия: синтаксиса от семантики.
6. Приведите конкретные примеры компилируемых языков программирования; интерпретируемых языков программирования; языков программирования, для которых имеются как компиляторы, так и интерпретаторы.
7. Основные достоинства и недостатки компиляторов.
8. Основные достоинства и недостатки интерпретаторов.
9. Опишите основные различия в синтаксисе и семантике двух известных Вам языков программирования.
10. Назовите основные фазы процесса трансляции и их назначение.
11. Назовите специфические особенности однопроходной трансляции.
12. Назовите специфические особенности многопроходной трансляции.
13. Приведите примеры возможных комбинаций однопроходной и многопроходной трансляции. Расскажите о практическом использовании этих схем.
14. Назовите основные способы определения формальных языков и их отличия.
15. Дайте определение формальной грамматики. Приведите пример.
16. Для чего нужны метаязыки? Приведите пример метаязыка.
17. Чем является формальный язык, порожаемый грамматикой?
18. Определите отношения вывода и назовите отличия, существующие между ними.
19. Для грамматики  $G_3 = (\{E, T, F\}, \{a, +, *, (\, )\}, P, E)$  где  $P$ : 1.  $E \rightarrow T \mid E + T$ , 2.  $T \rightarrow F \mid T * F$ , 3.  $F \rightarrow (E)$  а приведите пример вывода терминальной цепочки, содержащей три знака умножения и два знака сложения.
20. Приведите пример цепочки для грамматики  $G_3 = (\{E, T, F\}, \{a, +, *, (\, )\}, P, E)$ , где  $P$ : 1.  $E \rightarrow T \mid E + T$ , 2.  $T \rightarrow F \mid T * F$ , 3.  $F \rightarrow (E)$  а, содержащей пять операндов. Осуществите вывод этой цепочки из начального нетерминала.
21. Дайте определение цепочки, языка. Какие операции можно выполнять над цепочками символов? Что такое синтаксис и семантика языка?
22. Какие из перечисленных ниже тождеств являются истинными для двух произвольных цепочек символов  $\alpha$  и  $\beta$ , а какие нет: 1.  $|\alpha\beta| = |\alpha| + |\beta| = |\beta\alpha|$ ; 2.  $\alpha\beta = \beta\alpha$ ; 3.  $|\alpha^R| = |\alpha|$ ; 4.  $(\alpha^2\beta^2)^R = (\beta^R \alpha^R)^2$ ; 5.  $(\alpha^2\beta^2)^R = (\beta^R)^2(\alpha^R)^2$ .
23. Какие существуют методы задания языков? Почему метод перечисления всех допустимых цепочек языка не находит практического применения?
24. Какие дополнительные вопросы необходимо решить при задании языка программирования? Почему любой язык программирования не является чисто формальным языком?
25. Кто (или что) для любого языка программирования выступает в роли генератора цепочек языка? Кто (или что) выступает в роли распознавателя цепочек?
26. Как формулируется задача разбора? Всегда ли она разрешима и как она решается?

Министерство науки и высшего образования Российской Федерации Ульяновский государственный университет	Форма	
Ф-Рабочая программа по дисциплине		


27. Как выглядит описание грамматики в форме Бэкуса-Наура? Какие еще формы описания грамматик существуют?
28. Что такое распознаватель? Какие виды распознавателей существуют? Приведите структуру распознавателя.
29. Классификация распознавателей. Соотношение классификации распознавателей с классификацией языков и грамматик.
30. Назовите известные Вам классы грамматик с ограничениями на правила. Дайте их определения.
31. Чем отличается язык, определяемый формальной грамматикой, от языка, определяемого распознавателем?
32. Назовите эквивалентные соотношения между определениями формальных языков с помощью распознавателей и грамматик, заданных иерархией Хомского.
33. Дайте определение выводимости цепочки, непосредственной выводимости, нетривиальной выводимости, длины вывода.
34. Что такое сентенциальная форма грамматики? Что представляет собой множество конечных сентенциальных форм грамматики?
35. Что такое левосторонний и правосторонний выводы? Можно ли построить еще какие-нибудь варианты цепочек вывода?
36. Дайте определение дерева вывода. Приведите пример.
37. Дайте определение эквивалентности и однозначности грамматик. Сформулируйте проблему эквивалентности и однозначности.
38. Для чего нужен лексический анализатор? Что порождает лексический анализатор? Можно ли обойтись без сканера?
39. Назначение транслитератора.
40. Какая связь между сканером и конечным автоматом?
41. Существует ли связь между конечным автоматом и диаграммами Вирта?
42. Существует ли связь между конечным автоматом и праволинейными грамматиками?
43. Существует ли связь между конечным автоматом и грамматиками с левой рекурсией?
44. Как преобразовать грамматику с правой рекурсией в итеративную диаграмму Вирта?
45. Как преобразовать грамматику с левой рекурсией в итеративную диаграмму Вирта?
46. Назовите основные методы лексического анализа.
47. Приведите обобщенную структуру непрямого лексического анализатора.
48. Достоинства и недостатки непрямого лексического анализатора.
49. Приведите обобщенную структуру прямого лексического анализатора.
50. Достоинства и недостатки прямого лексического анализатора.
51. Перечислите конструкции конкретного языка программирования, которые целесообразно распознать на фазе лексического анализа.
52. Назначение синтаксического разбора.
53. Что является результатом синтаксического разбора?
54. Назовите основные критерии классификации синтаксического разбора.
55. Какие существуют методы разбора?
56. Связь методов разбора с выводом входной цепочки.
57. Особенности нисходящего разбора.
58. Особенности восходящего разбора.

Министерство науки и высшего образования Российской Федерации Ульяновский государственный университет	Форма	
Ф-Рабочая программа по дисциплине		

59. Особенности комбинированного разбора.
60. Какие существуют последовательности разбора?
61. Связь между методами разбора и последовательностью разбора.
62. Особенности разбора с просмотром вперед.
63. Дополнительная классификация контекстно-свободных грамматик.
64. Особенности разбора с возвратами.
65. Зачем при синтаксическом разборе нужны автоматы с магазинной памятью?
66. Как организован автомат с магазинной памятью?
67. Основные операции автомата с магазинной памятью.
68. Каким образом ограничения, накладываемые на грамматику, определяют реализацию автомата?
69. Приведите примеры известных Вам ограниченных грамматик.
70. Каким образом можно построить АМП по s-грамматике? Недостатки S-грамматики.
71. Разработка АМП по q-грамматике.
72. Приведите определения и краткие характеристики S-, q-, L-грамматик.
73. Методы программной реализации нисходящих автоматов с магазинной памятью.
74. Какие возможны варианты программной реализации нисходящего распознавателя при непосредственном использовании таблицы переходов АМП?
75. В чем специфика распознавателя, построенного с применением рекурсивного спуска?
76. Что такое таблица идентификаторов и для чего она предназначена? Какая информация может храниться в таблице идентификаторов?
77. Какие цели преследуются при организации таблицы идентификаторов? Исходя из каких характеристик оценивается эффективность того или иного метода организации таблицы?
78. Какие существуют способы организации таблиц идентификаторов?
79. Что такое коллизия? Почему она происходит при использовании хэш-функций для организации таблиц идентификаторов?
80. В чем заключаются преимущества и недостатки метода цепочек при организации таблиц идентификаторов?
81. Метод логарифмического поиска позволяет значительно сократить время поиска идентификатора в таблице. Однако он же значительно увеличивает время на помещение нового идентификатора в таблицу. Почему, тем не менее, можно говорить о преимуществах этого метода по сравнению с поиском метода прямого выбора?
82. Как могут быть скомбинированы различные методы организации таблиц идентификаторов?
83. Чем различаются таблица лексем и таблица идентификаторов? В какую из этих таблиц лексический анализатор не должен помещать ключевые слова, разделители и знаки операций?
84. Какие задачи в компиляторе решает семантический анализ? Можно ли построить компилятор без семантического анализатора?
85. Цепочка символов, принадлежащая любому языку программирования, может быть распознана с помощью распознавателя для контекстно-зависимых языков. При этом не будет требоваться дополнительный семантический анализ цепочки. Почему такой подход не применяется в компиляторах на практике?

Министерство науки и высшего образования Российской Федерации Ульяновский государственный университет	Форма	
Ф-Рабочая программа по дисциплине		


86. Почему распределение памяти не может быть выполнено до выполнения семантического анализа? Как работает процесс распределения памяти с таблицей идентификаторов?
87. Может ли компилятор для языков C и Pascal использовать статическую схему организации дисплея памяти процедуры (функции)? А динамическую схему?
88. От чего зависит состав информации, хранящейся в таблице RTTI? Зависит ли эта информация от синтаксиса и семантики исходного языка программирования, от архитектуры целевой вычислительной системы?
89. Почему в большинстве компиляторов помимо генерации результирующего объектного кода выполняется еще и его оптимизация? Можно ли построить компилятор, исключив фазу оптимизации кода?
90. В чем заключается основной принцип СУ-перевода? Является ли СУ-перевод наиболее эффективным методом порождения результирующего кода?
91. Перечислите основные способы внутреннего представления программы. На каких этапах компиляции лучше всего использовать каждый из этих способов?
92. От чего зависит код, порождаемый процедурой СУ-перевода для каждого узла дерева операций? Должна ли процедура СУ-перевода обращаться к таблице идентификаторов, должна ли она учитывать тип операндов?
93. Какой из двух основных методов оптимизации, машинно-зависимый или машинно-независимый, может породить более эффективный результирующий код? Как сочетаются эти два метода в компиляторе?
94. Почему линейные участки легче поддаются оптимизации, чем все другие части внутреннего представления программы и объектного кода?
95. Какие варианты распараллеливания выполнения операций может использовать компилятор при порождении объектного кода, необходимого для вычисления выражения  $A*B*C+A*D/E-V/E-F$ ?

Министерство науки и высшего образования Российской Федерации Ульяновский государственный университет	Форма	
Ф-Рабочая программа по дисциплине		


## 10. САМОСТОЯТЕЛЬНАЯ РАБОТА СТУДЕНТОВ

### Распределение по часам и видам самостоятельной работы

Название разделов и тем	Вид самостоятельной работы	Объем в часах	Форма контроля
1.1. Выражения и операторы языка C++	Проработка учебного материала, лабораторные работы, подготовка к сдаче экзамена, решение задач	1	Зачёт, проверка лабораторных работ, проверка решения задач
1.2. Управление ходом выполнения программы	Проработка учебного материала, лабораторные работы, подготовка к сдаче экзамена, решение задач	1	Зачёт, проверка лабораторных работ, проверка решения задач
1.3. Функции	Проработка учебного материала, лабораторные работы, подготовка к сдаче экзамена, решение задач	1	Зачёт, проверка лабораторных работ, проверка решения задач
1.4. Ввод/вывод данных в C++	Проработка учебного материала, лабораторные работы, подготовка к сдаче экзамена, решение задач	1	Зачёт, проверка лабораторных работ, проверка решения задач
1.5. Введение в объектно-ориентированное программирование	Проработка учебного материала, лабораторные работы, подготовка к сдаче экзамена, решение задач	1	Зачёт, проверка лабораторных работ, проверка решения задач
1.6. Применение шаблонов C++	Проработка учебного материала, лабораторные работы, подготовка к сдаче экзамена, решение задач	1	Зачёт, проверка лабораторных работ, проверка решения задач
1.7. Исключения в C++	Проработка учебного материала, лабораторные работы, подготовка к сдаче экзамена, решение задач	1	Зачёт, проверка лабораторных работ, проверка решения задач
2.1. История развития языков программирования	Проработка учебного материала, подготовка к сдаче экзамена	1	Зачёт
2.2. Парадигмы языков программирования	Проработка учебного материала, подготовка к сдаче экзамена	1	Зачёт
2.3. Критерии оценки языков программирования	Проработка учебного материала, подготовка к сдаче экзамена	1	Зачёт
2.4. Объекты данных в	Проработка учебного	1	Зачёт, проверка ла-

Министерство науки и высшего образования Российской Федерации Ульяновский государственный университет	Форма	
Ф-Рабочая программа по дисциплине		


языках программирования	материала, лабораторные работы, подготовка к сдаче экзамена		бораторных работ
2.5. Механизмы типизации	Проработка учебного материала, лабораторные работы, подготовка к сдаче экзамена	1	Зачёт, проверка лабораторных работ
2.6. Время жизни переменных	Проработка учебного материала, лабораторные работы, подготовка к сдаче экзамена	1	Зачёт, проверка лабораторных работ
2.7. Область видимости переменных	Проработка учебного материала, лабораторные работы, подготовка к сдаче экзамена	1	Зачёт, проверка лабораторных работ
2.8. Типы данных	Проработка учебного материала, лабораторные работы, подготовка к сдаче экзамена	1	Зачёт, проверка лабораторных работ
2.9. Выражения и операторы присваивания	Проработка учебного материала, лабораторные работы, подготовка к сдаче экзамена	1	Зачёт, проверка лабораторных работ
2.10. Структуры управления на уровне операторов	Проработка учебного материала, лабораторные работы, подготовка к сдаче экзамена	1	Зачёт, проверка лабораторных работ
2.11. Подпрограммы	Проработка учебного материала, лабораторные работы, подготовка к сдаче экзамена	1	Зачёт, проверка лабораторных работ
3.1. Общие сведения о трансляторах	Проработка учебного материала, лабораторные работы, подготовка к сдаче экзамена	1	Экзамен, проверка лабораторных работ
3.2. Основы теории языков и формальных грамматик	Проработка учебного материала, лабораторные работы, подготовка к сдаче экзамена	1	Экзамен, проверка лабораторных работ
3.3. Распознаватели	Проработка учебного материала, лабораторные работы, подготовка к сдаче экзамена	2	Экзамен, проверка лабораторных работ
3.4. Лексический анализ	Проработка учебного материала, лабораторные работы, подготовка к сдаче экзамена	2	Экзамен, проверка лабораторных работ
3.5. Синтаксический анализ	Проработка учебного материала, лабораторные работы, подготовка к	2	Экзамен, проверка лабораторных работ

Министерство науки и высшего образования Российской Федерации Ульяновский государственный университет	Форма	
Ф-Рабочая программа по дисциплине		

	сдаче экзамена		
3.6. Организация таблиц идентификаторов	Проработка учебного материала, лабораторные работы, подготовка к сдаче экзамена	2	Экзамен, проверка лабораторных работ
3.7. Семантический анализ и подготовка к генерации кода	Проработка учебного материала, лабораторные работы, подготовка к сдаче экзамена	2	Экзамен, проверка лабораторных работ
3.8. Распределение памяти	Проработка учебного материала, лабораторные работы, подготовка к сдаче экзамена	2	Экзамен, проверка лабораторных работ
3.9. Генерация кода. Методы генерации кода	Проработка учебного материала, лабораторные работы, подготовка к сдаче экзамена	2	Экзамен, проверка лабораторных работ
3.10. Оптимизация кода. Основные методы оптимизации	Проработка учебного материала, лабораторные работы, подготовка к сдаче экзамена	2	Экзамен, проверка лабораторных работ





Министерство науки и высшего образования Российской Федерации Ульяновский государственный университет	Форма	
Ф-Рабочая программа по дисциплине		

## б) Программное обеспечение

Для образовательного процесса по данной дисциплине необходим стационарный класс ПК с установленным следующим программным обеспечением:

- операционная среда ОС Windows/Linux;
- системы программирования на языках Си/C++ (Code::Blocks).

## 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, [2023]. – 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, [2023]. - 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, [2023]. – 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, [2023]. – URL: <https://www.rosmedlib.ru>. – Режим доступа: для зарегистрир. пользователей. – Текст : электронный.

1.5. Большая медицинская библиотека : электронно-библиотечная система : сайт / ООО «Букап». – Томск, [2023]. – 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, [2023]. – URL: <https://e.lanbook.com>. – Режим доступа: для зарегистрир. пользователей. – Текст : электронный.

1.7. ЭБС Znanium.com : электронно-библиотечная система : сайт / ООО «Знаниум». - Москва, [2023]. - 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, [2023].

### 3. Базы данных периодических изданий:


3.1. eLIBRARY.RU: научная электронная библиотека : сайт / ООО «Научная Электронная Библиотека». – Москва, [2023]. – URL: <http://elibrary.ru>. – Режим доступа : для авториз. пользователей. – Текст : электронный

3.2. Электронная библиотека «Издательского дома «Гребенников» (Grebinnikon) : электронная библиотека / ООО ИД «Гребенников». – Москва, [2023]. – URL: <https://id2.action-media.ru/Personal/Products>. – Режим доступа : для авториз. пользователей. – Текст : электронный.

4. Федеральная государственная информационная система «Национальная электронная библиотека» : электронная библиотека : сайт / ФГБУ РГБ. – Москва, [2023]. – 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»

Министерство науки и высшего образования Российской Федерации Ульяновский государственный университет	Форма	
Ф-Рабочая программа по дисциплине		

АБИС Мега-ПРО / ООО «Дата Экспресс». – URL: <http://lib.ulsu.ru/MegaPro/Web>. – Режим доступа : для пользователей научной библиотеки. – Текст : электронный.

Согласовано:

Инженер ведущий / Щуренко Ю.В. /  / 04.05.2023  
Должность сотрудника УНТТ                      ФИО                      подпись                      дата

Министерство науки и высшего образования Российской Федерации Ульяновский государственный университет	Форма	
Ф-Рабочая программа по дисциплине		

## 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.

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, указывается в соответствии со сведениями о материально-техническом обеспечении и оснащенности образовательного процесса, размещенными на официальном сайте УлГУ в разделе «Сведения об образовательной организации».

## 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  профессор каф. ИТ /Цыганова Ю.В./  
подпись должность ФИО