

МИНПРОСВЕЩЕНИЯ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования
"Тульский государственный педагогический университет им. Л.Н. Толстого"
(ФГБОУ ВО "ТГПУ им. Л.Н. Толстого")

**МОДУЛЬ "АЛГОРИТМИЗАЦИЯ И
ПРОГРАММИРОВАНИЕ"
Программирование**

рабочая программа дисциплины (модуля)

Закреплена за кафедрой	институт передовых информационных технологий
ОПОП	Направление 02.03.02 Фундаментальная информатика и информационные технологии направленность (профиль) Инженерия программного обеспечения
Квалификация	Бакалавр
Год начала подготовки	2023
Форма обучения	очная
Общая трудоемкость	5 з.е.

Виды контроля по семестрам:
экзамен 2

Семестр(Курс.Номер семестра на курсе)	2(1.2)		Итого	
	УП	РПД	УП	РПД
Лекции	34	34	34	34
Лабораторные	34	34	34	34
Итого ауд.	68	68	68	68
КСР	4	4	4	4
Контактная работа	72	72	72	72
Сам. работа	72	72	72	72
Часы на контроль	36	36	36	36
Практическая подготовка	0	0	0	0
Семинары	0	0	0	0
Консультации	0	0	0	0
Итого трудоемкость в часах	180	180	180	180

Программу составил(и):

к.ф.-м.н., доцент, Ванькова Валентина Сергеевна; к.п.н., доцент, Мартынюк Юлия Михайловна

Рабочая программа дисциплины

Программирование

разработана в соответствии с ФГОС:

Федеральный государственный образовательный стандарт высшего образования - бакалавриат по направлению подготовки 02.03.02 Фундаментальная информатика и информационные технологии (приказ Минобрнауки России от 23.08.2017 г. № 808)

составлена на основании учебного плана:

Направление 02.03.02 Фундаментальная информатика и информационные технологии
направленность (профиль) Инженерия программного обеспечения
утвержденного Учёным советом вуза от 27.10.2022 протокол № 13.

РПД утверждена Учёным советом университета
протокол от 27.10.2022 г. № 13

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Формирование способности программировать на языке C++, поддерживающем процедурную, объектно-ориентированную и обобщенную парадигмы.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП

Цикл (раздел) ООП:	Б1.О.07
2.1	Требования к предварительной подготовке обучающегося:
1.	К началу изучения дисциплины студенты по результатам изучения школьного курса «Информатика и ИКТ» должны знать определения и свойства модели, алгоритма и программы, иметь представление о способах записи алгоритмов и реализации основных алгоритмических конструкций следования, ветвления и повторения в одном из языков программирования или на псевдокоде.
2.	Введение в программирование
2.2	Дисциплины и практики, для которых освоение данной дисциплины (модуля) необходимо как предшествующее:
1.	Системы компьютерной математики
2.	Структуры и алгоритмы компьютерной обработки данных
3.	Теория и методика изучения информатических дисциплин в системе дополнительного образования
4.	Технологии программирования
5.	Компьютерное моделирование
6.	Подготовка к процедуре защиты и защита выпускной квалификационной работы
7.	Рекурсивно-логическое программирование
8.	Функциональное программирование
9.	Объектно-ориентированное программирование
10.	Алгоритмы и структуры данных
11.	Базы данных
12.	Веб-программирование
13.	Операционные системы
14.	Практикум по программированию
15.	Основы проектирования веб-интерфейсов
16.	Практикум по веб-программированию
17.	Разработка приложений для мобильных платформ
18.	Технологии параллельного программирования
19.	Экономические информационные системы
20.	Методы и технологии визуализации данных
21.	Применение нейронных сетей
22.	Разработка экономических информационных систем
23.	Экономико-правовые основы рынка программного обеспечения
24.	Аналитика больших данных
25.	Машинное обучение
26.	Проектирование и разработка пользовательских интерфейсов
27.	Технология IoT
28.	Производственная практика: научно-исследовательская работа

3. СООТНЕСЕНИЕ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ (МОДУЛЮ) С ИНДИКАТОРАМИ ДОСТИЖЕНИЯ КОМПЕТЕНЦИЙ

3.1 Компетенции обучающегося и индикаторы их достижения:

ОПК-2: Способен применять компьютерные/суперкомпьютерные методы, современное программное обеспечение, в том числе отечественного происхождения, для решения задач профессиональной деятельности

ОПК-2.1	Знает основные положения и концепции в области программирования, архитектуру языков программирования, теории коммуникации, знает основную терминологию, знаком с содержанием Единого Реестра Российских программ
	типы данных, алгоритмические конструкции ЯП; способы описания типов данных и функций пользователя в ЯП; теоретические основы работы с файлами в ЯП; интерфейс среды программирования VS, различные способы формирования массива; описание, использование механизма указателей для работы с данными различных базовых и производных типов. принципов организации, описания, инициализации и доступа к структурам данных; алгоритмов создания экземпляров класса, работы конструктора и деструктора класса,

	работы со статическими и константными методами класса
ОПК-2.2	Умеет анализировать типовые языки программирования, составлять программы
	умеет применять полученные знания при решении практических задач профессиональной деятельности
ОПК-2.3	Имеет практический опыт решения задач анализа, интеграции различных типов программного обеспечения, анализа типов коммуникаций
	имеет навык в декомпозиции сложных задач на независимые подзадачи; отладки и тестирования программ; организации модульного строения программ;
ОПК-3: Способен к разработке алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей, созданию информационных ресурсов глобальных сетей, образовательного контента, прикладных баз данных, тестов и средств тестирования систем и средств на соответствие стандартам и исходным требованиям	
ОПК-3.1	Знает методы теории алгоритмов, методы системного и прикладного программирования, основные положения и концепции в области математических, информационных и имитационных моделей
	Знает алгоритмы создания экземпляров класса, работы конструктора и деструктора класса, работы со статическими и константными методами класса.
ОПК-3.2	Умеет соотносить знания в области программирования, интерпретацию прочитанного, определять и создавать информационные ресурсы глобальных сетей, образовательного контента, средств тестирования систем
	использовать в программах файлы в качестве источника и приемника данных; применять базовые алгоритмы сортировки и поиска при решении конкретных задач; выполнять операции обработки данных строкового типа. выбирать метод программирования в зависимости от разработанной математической модели и модели данных;
ОПК-3.3	Имеет практический опыт применения разработки программного обеспечения
	декомпозиции сложных задач на независимые подзадачи; отладки и тестирования программ; обработки многомерных массивов; использования динамической памяти; работы в коллективах разработчиков; владеет рекурсией, как методом решения задач
ОПК-6: Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности	
ОПК-6.1	Знает принципы работы современных информационных технологий; современные тенденции развития информатики и вычислительной техники, цифровых компьютерных технологий
	знает различные способы формирования массива; описание, использование механизма указателей для работы с данными различных базовых и производных типов. принципов организации, описания, инициализации и доступа к структурам данных
ОПК-6.2	Умеет применять современные информационные технологии и использовать возможности программных средств в профессиональной деятельности
	умеет осуществлять поиск необходимой информации в глобальных компьютерных сетях и LMS MOODLE
ОПК-6.3	Владеет основными методами, способами и средствами получения, хранения и обработки информации с использованием современных информационных технологий в условиях цифровой трансформации
	имеет навык в декомпозиции сложных задач на независимые подзадачи; отладки и тестирования программ; организации модульного строения программ; обработки многомерных массивов; использования динамической памяти; владеет рекурсией, как методом решения задач
3.2 Результаты обучения по дисциплине:	
В результате освоения дисциплины обучающийся должен:	
	Знать:
3.1	типы данных, алгоритмические конструкции ЯП; способы описания типов данных и функций пользователя в ЯП; теоретические основы работы с файлами в ЯП; интерфейс среды программирования VS, различные способы формирования массива; описание, использование механизма указателей для работы с данными различных базовых и производных типов. принципов организации, описания, инициализации и доступа к структурам данных; алгоритмов создания экземпляров класса, работы конструктора и деструктора класса, работы со статическими и константными методами класса.
	Уметь:
У.1	применять полученные знания при решении практических задач профессиональной деятельности;
У.2	решать типовые задачи по программированию с использованием базовых алгоритмических конструкций и подпрограмм;
У.3	осуществлять поиск необходимой информации в глобальных компьютерных сетях и LMS MOODLE;
У.4	использовать в программах файлы в качестве источника и приемника данных;
У.5	применять базовые алгоритмы сортировки и поиска при решении конкретных задач;

У.6	выполнять операции обработки данных строкового типа. выбирать метод программирования в зависимости от разработанной математической модели и модели данных;
У.7	осуществлять форматированный и неформатированный ввод-вывод данных в файл и в поток.
	Владеть:
В.1	декомпозиции сложных задач на независимые подзадачи; отладки и тестирования программ;
В.2	организации модульного строения программ;
В.3	обработки многомерных массивов;
В.4	использования динамической памяти;
В.5	работы в коллективах разработчиков;
В.6	владеет рекурсией, как методом решения задач

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

Код занятия	Наименование разделов и тем /вид занятия/	Семестр / Курс	Часов	Литература	Содержание
	Массивы				
1.1	Массивы Методы инициализации. Массивы символьных данных. /Лек/	2	4	Л1.1 Л1.2	Массивы и переменные с индексами. Статические массивы. Описание массивов. Организация памяти. Инициализация массивов (из файла, по формуле, случайными числами, с консоли). Массивы констант. Строковые константы. Функции пользователя для генерации массива. Массивы как параметры функций. Классические задачи поиска в линейных и двумерных массивах. Ввод-вывод символьных данных. Внутренние коды и упорядоченность символов. Строковые константы.
1.2	Алгоритмы поиска и сортировки в массивах /Лек/	2	2	Л1.1 Л1.2	Упорядочение в одномерных массивах. Методы сортировки одномерных массивов (простого выбора, простых обменов, простых включений, и т.п.).
1.3	Описание, инициализация массивов. Алгоритмы поиска в массивах /Лаб/	2	2	Л1.1 Л1.2	Массивы и переменные с индексами. Описание массивов. Инициализация массивов. Функции и массивы. Массивы в параметрах функций. Двумерные массивы
1.4	Сортировки одномерных массивов /Лаб/	2	2	Л1.1 Л1.2	Методы сортировки одномерных массивов (простого выбора, простых обменов, простых включений).
1.5	Массивы символов /Лаб/	2	2	Л1.1 Л1.2	Ввод-вывод символьных данных. Внутренние коды и упорядоченность символов. Строковые константы.
1.6	Описание массивов. Инициализация массивов. /Ср/	2	2	Л1.1 Л1.2	Описание массивов. Инициализация массивов. Базовый тип массива. Функции и массивы. Одномерные и двумерные массивы
1.7	Сортировки массивов /Ср/	2	2	Л1.1 Л1.2	Упорядочение в одномерных массивах. Методы сортировки одномерных массивов (простого выбора, простых обменов, простых включений). Идея метода сортировки. Реализация метода сортировки. Функции пользователя сортировки одномерного массива.
1.8	Размерность массива /Ср/	2	2	Л1.1 Л1.2	Взаимосвязь одномерных и двумерных массивов.
1.9	Индивидуальное задание /Ср/	2	4	Л1.1 Л1.2	Решение классических задач поиска и сортировки в соответствии с индивидуальным заданием.
	Массивы и указатели				

2.1	Указатели. Указатели и адресная арифметика. Строки и указатели. Сравнение и сортировка текстовых данных. /Лек/	2	2	Л1.1 Л1.2	Операции с указателями. Организация памяти в процессорах 80□86 и указатели языка С.
2.2	Массивы и указатели. Динамическое размещение массивов /Лек/	2	4	Л1.1 Л1.2	Массивы и указатели. Динамическое размещение одномерных и двумерных массивов. Указатели на многомерные массивы. Методы доступа к элементам массива.
2.3	Массивы и указатели. /Лаб/	2	2	Л1.1 Л1.2	Запись основных функций пользователя по обработке массивов в идеологии указателей
2.4	Динамическое размещение массивов /Лаб/	2	2	Л1.1 Л1.2	Динамическое размещение массивов. Указатели на многомерные массивы. Функции работы с памятью в языке С и языке С++
2.5	Строки и указатели. Сравнение и сортировка текстовых данных /Лаб/	2	2	Л1.1 Л1.2	Библиотеки функций по обработке строк. Решение задач
2.6	Указатели. Указатели и адресная арифметика. Операции с указателями. Организация памяти и указатели языка С++. /Ср/	2	4	Л1.1 Л1.2	Изучение теории. Решение задач
2.7	Массивы и указатели. /Ср/	2	2	Л1.1 Л1.2	Изучение теории. Решение задач
2.8	Динамическое размещение массивов. Указатели на многомерные массивы. Методы доступа к элементам массива. /Ср/	2	2	Л1.1 Л1.2	Изучение теории. Решение задач
2.9	Запись алгоритмов сортировки одномерного массива в синтаксисе указателей. Решение классических задач с использованием динамической памяти для двумерного массива. /Ср/	2	2	Л1.1 Л1.2	Изучение теории. Решение задач
2.10	Строки и указатели. Сравнение и сортировка текстовых данных. /Ср/	2	2	Л1.1 Л1.2	Изучение теории. Решение задач
	Указатели и функции				
3.1	Указатели в параметрах функций. /Лек/	2	2	Л1.1 Л1.2	Имитация подпрограмм. Массивы в параметрах функций. Указатели на функции как параметры функций пользователя. Указатель, как возвращаемое функцией значение
3.2	Указатели и алгоритмы обработки строк /Лек/	2	2		Работа со строками как массивами символов. Реализация базовых функций удаления лишних пробелов в строке, вычисления количества слов в строке, вставки подстроки в строку и т.д.
3.3	Функции с переменным количеством параметров. Перегрузка функций. /Лек/	2	2	Л1.1 Л1.2	Функции с переменным количеством параметров. Перегрузка функций.
3.4	Указатели в параметрах функций. /Лаб/	2	2	Л1.1 Л1.2	Указатели в параметрах функций. Имитация подпрограмм. Массивы в параметрах функций. Указатели на функции как параметры.

3.5	Функции с переменным количеством параметров. Перегрузка функций. /Лаб/	2	2	Л1.1 Л1.2	Функции с переменным количеством параметров. Перегрузка функций.
3.6	Указатели в параметрах функций. Имитация подпрограмм. Функции с переменным количеством параметров. Перегрузка функций. /Ср/	2	2	Л1.1 Л1.2	Указатели в параметрах функций. Имитация подпрограмм. Массивы в параметрах функций. Указатели на функции как параметры. Функции с переменным количеством параметров. Перегрузка функций.
3.7	Имитация подпрограмм. /Ср/	2	2	Л1.1 Л1.2	Параметры по ссылке. Параметры по значению
3.8	Функции с переменным количеством параметров. Перегрузка функций. /Ср/	2	2	Л1.1 Л1.2	Указатели в параметрах функций. Имитация подпрограмм. Массивы в параметрах функций. Указатели на функции как параметры. Функции с переменным количеством параметров. Перегрузка функций.
	Препроцессорные средства обработки программ.				
4.1	Препроцессорная обработка /Лек/	2	2	Л1.1 Л1.2	Стадии и команды препроцессорной обработки. Замены в тексте. Включение текстов из файлов. Условная компиляция. Макроподстановки средствами препроцессора.
4.2	Препроцессорная обработка /Лаб/	2	2	Л1.1 Л1.2	Стадии и команды препроцессорной обработки. Замены в тексте. Включение текстов из файлов. У
4.3	Препроцессорная обработка /Ср/	2	4	Л1.1 Л1.2	Стадии и команды препроцессорной обработки. Замены в тексте. Включение текстов из файлов.
4.4	Условная компиляция /Ср/	2	4	Л1.1 Л1.2	Условная компиляция
4.5	Макроподстановки средствами препроцессора /Ср/	2	2	Л1.1 Л1.2	Макроподстановки средствами препроцессора
	Обработка исключений				
5.1	Общие принципы механизма обработки исключений /Лек/	2	2	Л1.1 Л1.2	Общие принципы механизма обработки исключений. Синтаксис и семантика генерации и обработки исключений. Обработка исключений при динамическом выделении памяти.
5.2	Исследование алгоритмов работы механизма исключений /Лаб/	2	2	Л1.1 Л1.2	Синтаксис и семантика генерации и обработки исключений. Обработка исключений при динамическом выделении памяти.
5.3	Теоретические основы /Ср/	2	2	Л1.1 Л1.2	Общие принципы механизма обработки исключений. Синтаксис и семантика генерации и обработки исключений. Обработка исключений при динамическом выделении памяти.
5.4	Обработка исключений при динамическом выделении памяти /Ср/	2	4	Л1.1 Л1.2	Моделирование функций пользователя
	Рекурсия как метод решения задач				
6.1	Теоретические основы рекурсии /Лек/	2	4	Л1.1 Л1.2	Понятие рекурсии. Рекурсия в информатике. Рекурсивная триада. Рекурсивный спуск. Рекурсивный подъем. Хвостовая оптимизированная рекурсия.
6.2	Динамическая база рекурсии /Лек/	2	2	Л1.1 Л1.2	На примере вычисления чисел Фибоначчи и биномиальных коэффициентов

6.3	Классические задачи /Лаб/	2	2	Л1.1 Л1.2	Алгоритмы теории чисел (факториал, степень, моделирование арифметических операций, распознавание простого числа, вычисление количества и суммы делителей натурального числа).
6.4	Рекурсивные алгоритмы теории чисел /Лаб/	2	2	Л1.1 Л1.2	Классические задачи (Ханойские башни, числа Фибоначчи, Биномиальные коэффициенты, алгоритм Ламберта, функции Аккермана, Маккарти, Кадью).
6.5	Динамическая база рекурсии /Лаб/	2	2	Л1.1 Л1.2	Вычисление чисел Фибоначчи, биномиальных коэффициентов
6.6	Теоретические основы рекурсии /Ср/	2	4	Л1.1 Л1.2	Понятие рекурсии. Рекурсия в информатике. Рекурсивная триада. Рекурсивный спуск. Рекурсивный подъем. Стек. Рекурсивный взрыв. Хвостовая оптимизированная рекурсия.
6.7	Алгоритмы теории чисел /Ср/	2	4	Л1.1 Л1.2	Алгоритмы теории чисел (факториал, степень, моделирование арифметических операций, распознавание простого числа, вычисление количества и суммы делителей натурального числа).
6.8	Рекурсия с динамической базой /Ср/	2	4	Л1.1 Л1.2	Рекурсивный взрыв. Рекурсия с динамической базой. Вычисление чисел Фибоначчи и биномиальных коэффициентов
6.9	Классические задачи /Ср/	2	2	Л1.1 Л1.2	Классические задачи (Ханойские башни).
6.10	Специальные рекурсивные функции /Ср/	2	2	Л1.1 Л1.2	Алгоритм Ламберта, функции Аккермана, Маккарти, Кадью).
	Структуры				
7.1	Структура как тип данных /Лек/	2	2	Л1.1 Л1.2	Объявление и определение структурных объектов. Инициализация структур. Присваивание значений. Доступ к элементам структур. Массив, как структура данных.
7.2	Указатели на структуры как тип параметров и результата функции пользователя /Лек/	2	2	Л1.1 Л1.2	Массивы структур. Указатели на структуры. Указатели как средство доступа к компонентам структур.
7.3	Структуры как тип данных /Лаб/	2	2	Л1.1 Л1.2	Объявление и определение структурных объектов. Инициализация структур. Присваивание значений. Доступ к элементам структур.
7.4	Указатели на структуры как тип параметров и результата функции пользователя /Лаб/	2	2	Л1.1 Л1.2	Разработка функций пользователя на основе использования указателей на структуры
7.5	Использование структур для моделирования баз данных /Лаб/	2	2	Л1.1 Л1.2	Работа по индивидуальным заданиям
7.6	Структуры как тип данных /Ср/	2	2	Л1.1 Л1.2	Объявление и определение структурных объектов. Инициализация структур. Присваивание значений. Доступ к элементам структур.
7.7	Массивы структур. Указатели на структуры. /Ср/	2	4	Л1.1 Л1.2	Указатели как средство доступа к компонентам структур. Операции точка и стрелка. Приоритет операций. Составление программ по заданиям преподавателя
7.8	Разработка БД на основе структур /Ср/	2	4	Л1.1 Л1.2	Индивидуальные проекты в предметной области
	Средства обобщенного программирования				

8.1	Шаблоны функций пользователя /Лек/	2	2	Л1.1 Л1.2	Шаблоны. Понятие обобщённого программирования и введение в стандартную библиотеку шаблонов. Программирование и методы вычислений: Программирование простейших шаблонов функций. Понятие шаблона как средства параметризации алгоритмов.
8.2	Разработка шаблонов функций /Лаб/	2	2	Л1.1 Л1.2	Программирование простейших шаблонов функций. Понятие шаблона как средства параметризации алгоритмов.
8.3	Проектирование шаблонов /Ср/	2	2	Л1.1 Л1.2	Программирование простейших шаблонов функций. Понятие шаблона как средства параметризации алгоритмов.
8.4	/КСР/	2	4		

5. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

5.1. Типовые задания для проведения текущего контроля

2 семестр

Лабораторная работа № 1 (2 часа)

1. Из файла вводится последовательность символов. Определить сколько среди них цифр и чему равна их сумма.
2. Подсчитайте сколько раз среди последовательности символов встречается символ, введенный с клавиатуры. Количество символов определяется вводом с клавиатуры, но не превосходит 100.
3. Пусть даны целое число n ($n > 0$) и последовательность символов $s_1, s_2, s_3, \dots, s_n$ среди которых есть двоеточие. Получите все символы, расположенные от первого двоеточия до второго двоеточия. Если второго двоеточия нет, то получите все символы от первого двоеточия и до конца последовательности.
4. Пусть задан текст за которым следует точка. Поменяйте все строчные латинские буквы на прописные.
5. Составить программу, которая инвертирует слово, введенное с клавиатуры.
6. Задана строка, состоящая из слов, разделенных одним или несколькими пробелами. Удалить повторные вхождения каждого слова.

Лабораторная работа № 2 (2 часа)

1. Для решения task2_3 допишите фрагмент программы, который обеспечивает вычисление суммы и произведения элементов массива и построчный вывод этих характеристик в файл output.txt.
2. Для решения task2_4 допишите фрагмент программы, который обеспечивает изменение элементов массива следующим образом: каждый четный элемент массива делит на 2, а каждый нечетный – удваивает.
3. Для решения task2_5 допишите фрагмент программы, который обеспечивает
 - a. Подсчет в массиве mm количества элементов массива, кратных числу 3;
 - b. Подсчет в массиве qq количества элементов массива, являющихся простыми числами;
 - c. * Подсчет в массиве pp суммы цифр всех элементов массива.

5.2. Типовые задания для проведения промежуточной аттестации

Вопросы к экзамену во 2 семестре

1. Файлы. Потоки. Функции организации работы с файлами и потоками. Ввод и вывод данных различных типов.
2. Функции пользователя. Локальные и глобальные переменные. Примеры.
3. Параметры процедур и функций. Примеры.
4. Области действия и видимости переменных. Примеры.
5. Понятие массива. Описание, инициализация массивов различных базовых типов.
6. Алгоритмы сортировки массивов. Сортировка массива “пузырьком”. Сортировка массива выбором. Сортировка массива включением.
7. Работа с элементами одномерных и двумерных массивов. Обработка массивов. Алгоритмы поиска в массивах.
8. Адреса переменных. Понятие указателя. Адресная арифметика. Нулевой указатель.
9. Указатели в параметрах функций.
10. Массивы и указатели.
11. Объявление строк. Сравнение и сортировка текстовых данных.
12. Строки и указатели. Обработка фрагментов строк.
13. Массивы динамической памяти.
14. Массивы указателей и моделирование многомерных массивов.
15. Поточковый ввод-вывод. Открытие и закрытие потока. Ввод-вывод в потоке чисел, символов и строк. Строковый обмен с файлами на диске. Режим форматированного обмена с файлами.
16. Двоичный (бинарный) режим работы с файлами на диске. Позиционирование в потоке.
17. Директивы препроцессора. Замены в тексте. Препроцессорные операции в строке замещения.
18. Директивы препроцессора. Включение текстов из файлов. Макроподстановки средствами препроцессора.
19. Обработка исключений.

15. Понятие рекурсии. Рекурсивная триада. Рекурсивный спуск. Рекурсивный подъем.
16. Рекурсивные алгоритмы теории чисел (факториал, степень, моделирование арифметических операций, распознавание простого числа, вычисление количества делителей натурального числа).
17. Рекурсивный взрыв. Рекурсия с динамической базой. Демонстрация алгоритмов рекурсии с динамической базой на примере вычисления чисел Фибоначчи и биномиальных коэффициентов.
18. Постановка задачи и реализация классического алгоритма задачи о Ханойских башнях.
19. Структуры. Объявление и определение структурных объектов. Инициализация структур. Присваивание значений. Доступ к элементам структур.
20. Массивы (статические и динамические) структур. Формирование и обработка данных. Демонстрация примеров в предметной области, выбранной студентом.
21. Взаимосвязь структур, функций и указателей.
22. Ввод-вывод структур при работе с файлами на диске.
23. Шаблоны функций.
24. Перегрузка функций.
25. Класс файлового потока. Методы. Строковые потоки.
26. Сравнительный анализ потоков библиотек `iostream` и `stdio`.
27. Исключения. Обработка исключений.
28. Генерация исключений из функций пользователя.

5.3. Перечень видов оценочных средств

1. Список вопросов к экзамену.
2. Типовые задания для проведения лабораторных работ (представлены в LMS MOODLE).
3. Примерные задания для контрольных работ.

5.4. Процедура применения оценочных материалов

Промежуточная аттестация может проводиться с применением электронного обучения и (или) дистанционных образовательных технологий в соответствии с "Порядком проведения промежуточной аттестации с применением электронного обучения и /или дистанционных образовательных технологий".

Задания, выполнение которых предполагает общение по методике «мозгового штурма» с целью формирования умений осуществлять декомпозицию сложных задач на независимые подзадачи

1. Дано целое k от 1 до 180. Определить, какая цифра находится в k -ой позиции последовательности 1011121314151617181920 ...96979899?
2. Составить программу, которая по введенному натуральному числу n , обозначающему позицию в последовательности 1011121314151617181920... определяет цифру, стоящую на указанном месте.
3. Составить программу, которая по введенному натуральному числу n , обозначающему позицию в последовательности 101001000100001000001000000..... определяет цифру, стоящую на указанном месте.
4. Дана непустая последовательность ненулевых целых чисел, за которой следует 0. Определить сколько раз в этой последовательности меняется знак. Например, в последовательности 1, -34, 21, 34, -1, 9, 78, 0 знак меняется 4 раза?
5. Разложить заданное целое число на простые множители.
6. Разложить четное число на сумму простых чисел.
7. Составить программу, выводющую на экран все четырехзначные палиндромы, квадраты которых тоже являются палиндромами.
8. ПОСЛЕДОВАТЕЛЬНОСТЬ ЧИСЕЛ. Рассмотрим бесконечную в обе стороны последовательность целых чисел F_i , в которой для любого целого i элемент F_{i+2} вычисляется с использованием следующего условия Фибоначчи: $F_{i+2}=F_{i+1}+F_i$. Пусть заданы два различных члена этой последовательности – F_i и F_j с соответствующими номерами i и j , а также некоторое целое число n . Необходимо восстановить элемент этой последовательности F_n , соответствующий номеру n . Требуется написать программу, которая по заданным числам i , F_i , j , F_j , n вычисляет искомый элемент F_n описанной выше последовательности.
9. НАИБОЛЬШЕЕ ПРОИЗВЕДЕНИЕ. Дано N натуральных чисел. Требуется выбрать из них три таких числа, произведение которых максимально.
10. ТЕАТР. В театре N мест, пронумерованных целыми числами от 1 до N . Некоторые из зрителей опоздали на спектакль, поэтому после третьего звонка те зрители, которые имели билеты на неудобные места, пересели на более удобные. Опоздавшие зрители, которые пришли уже после третьего звонка, сели на первое попавшееся свободное место. В антракте один из опоздавших зрителей решил сесть на свое место. Если его место до этого было занято, то тот, кто там сидел, пересаживался на свое место. Если и там кто-то уже сидел, то и этот зритель также вынужден был вернуться на свое место. И так далее. Поскольку в театр попали только зрители, имевшие на руках билеты, то начавшийся в антракте процесс пересаживания зрителей обязательно заканчивался. Необходимо посчитать, сколько человек в результате такого пересаживания были вынуждены поменять свои места. Требуется написать программу, которая вычисляет количество зрителей, поменявших свои места из-за опоздания одного зрителя.
11. КИНОТЕАТР. X мальчиков и Y девочек пошли в кинотеатр и купили билеты на подряд идущие места в одном ряду. Требуется написать программу, которая выдаст, как нужно сесть мальчикам и девочкам, чтобы рядом с каждым мальчиком сидела хотя бы одна девочка, а рядом с каждой девочкой – хотя бы один мальчик.
12. ЗОЛОТОЙ ПЕСОК. Сотрудники завода по производству золотого песка из воздуха решили поправить свое финансовое положение. Они пробрались на склад завода, где хранился золотой песок трёх видов. Один килограмм песка первого вида они смогли бы продать за A_1 рублей, второго вида – за A_2 рублей, а третьего вида – за A_3 рублей. Так получилось, что у сотрудников оказались с собой только три ёмкости: первая была рассчитана на B_1 килограмм груза, вторая – на B_2 килограмм, а третья – на B_3 килограмм. Им надо было заполнить полностью все ёмкости таким образом,

чтобы получить как можно больше денег за весь песок. При заполнении ёмкостей следует выполнять правила:

А) нельзя смешивать песок разных видов, т.е. нельзя помещать в одну ёмкость песок разных видов;

Б) заполнять ёмкости песком так, чтобы один вид песка занимал не более чем одну ёмкость.

Требуется написать программу, которая определяет, за какую сумму предприимчивые сотрудники смогут продать весь песок в случае наилучшего для себя заполнения ёмкостей песком.

13. **НАИБОЛЬШЕЕ ПРОИЗВЕДЕНИЕ.** Дано N натуральных чисел. Требуется выбрать из них три таких числа, произведение которых максимально.

Задания, выполнение которых за фиксированный отрезок времени позволяет судить об уровне и качестве навыков самостоятельной работы студентов

1. Дана последовательность из N натуральных чисел, оканчивающаяся 0. Составить программу поиска среди них трех таких чисел, произведение которых максимально.
2. Дано натуральное m . Написать программу, которая без использования операции логарифмирования определит наименьшее целое k , при котором $5k > m$.
3. На отрезке $[a, b]$, где $a < b$, a и b целые числа. Написать программу поиска на данном интервале количества чисел, сумма делителей которых кратна 24.
4. Заданы размеры прямоугольного отверстия A, B и размеры кирпича X, Y, Z . Определить, пройдет ли кирпич в отверстие. Замечание: задачу решать в предположении, что $A > B$, $X > Y > Z$.
5. Составьте программу поиска наибольшего целого числа, меньшего 900, которое нацело делится на 47.
6. Дано целое от 1 до 9999. Составьте программу для вывода в строку словесного описания данного числа в виде: «четное двухзначное число», «Нечетное четырехзначное число».
7. Составьте программу, которая удвоит натуральное число, введенное с клавиатуры, если оно содержит в своей записи хотя бы одну цифру «единицу», и возведет в квадрат, в противном случае.
8. Дан массив из 20 целых чисел диапазона $(-50, 50)$. Найти наименьший из его положительных элементов.
9. Массив a сформирован из 30 целых чисел, считанных из файла input.txt. Составить массив b следующим образом: $b[1]=a[30]$, $b[2]=a[29]$, $b[3]=a[28]$, ..., $b[30]=a[1]$.
10. Дан массив из 15 целых чисел из диапазона $(-30, 40)$. Найти наибольший из его отрицательных элементов.
11. Массив a сформирован из 20 целых чисел, считанных из файла input.txt. Составить массив b следующим образом: $b[1]=a[11]$, $b[2]=a[12]$, ..., $b[10]=a[20]$, $b[11]=a[1]$, ..., $b[20]=a[10]$.
12. Дан массив из 20 целых чисел из диапазона $(-40, 20)$. Среди элементов, не превосходящих по модулю 15, найти наименьший.
13. Массив a сформирован из 30 целых чисел, считанных из файла input.txt. Составить массив b следующим образом: $b[1]=a[29]$, $b[2]=a[2]$, $b[3]=a[27]$, $b[4]=a[4]$, ..., $b[30]=a[30]$.
14. Дан массив из 17 целых чисел из диапазона $(-50, 30)$. Среди элементов, не превосходящих по модулю 20, найти наибольший.
15. Массив a сформирован из 20 целых чисел, считанных из файла input.txt. Составить массив b следующим образом: $b[1]=a[1]$, $b[2]=a[20]$, $b[3]=a[3]$, $b[4]=a[18]$, ..., $b[20]=a[2]$.
16. Составить программу, демонстрирующую механизм перегрузки функций для функции генерации одномерного массива с базовыми типами: целочисленным, вещественным, символьным.
17. Составить программу, демонстрирующую механизм перегрузки функций для функции сортировки одномерного массива с базовыми типами: целочисленным, вещественным, символьным (в каждом из алгоритмов приведите различные алгоритмы сортировки).
18. Составить программу, демонстрирующую механизм перегрузки функций для функции обмена значениями двух аргументов с базовыми типами: целочисленным, вещественным, символьным.
19. Составить программу, демонстрирующую механизм разработки шаблона для функции обмена значениями двух аргументов с несколькими базовыми типами. Протестируйте приложение.
20. Составить программу, демонстрирующую механизм разработки шаблона для сортировки одномерного массива с базовыми типами: целочисленным, вещественным, символьным (для сортировки используется любой известный метод).

6. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

6.1. Рекомендуемая литература

6.1.1. Основная литература

	Авторы, составители	Заглавие	Издательство, год (кол-во экземпляров для печатных изданий)	Ссылка на электронное издание
Л1.1	Березин Б. И., Березин С. Б.	Начальный курс C и C++: учебное пособие	, 2008	http://biblioclub.ru/index.php?page=book&id=54733
Л1.2	Александров Э. Э., Афонин В. В.	Программирование на языке C в Microsoft Visual Studio 2010: учебное пособие	Интернет-Университет Информационных Технологий, 2010	http://biblioclub.ru/index.php?page=book&id=233564

6.2. Перечень ресурсов информационно-телекоммуникационной сети "Интернет"

Э1	Поисковые системы
----	-------------------

Э2	Портал Министерства науки и высшего образования и РФ
Э3	Сайт ГОУ ВО ТПУ им. Л.Н. Толстого
Э4	Интуит. Национальный открытый университет [Электронный ресурс]
Э5	ИКТ [Электронный ресурс]: федеральный образовательный портал / ФГАУ ГНИИ ИТТ "Ин-форматика". - М. : [б. и.], 2003

6.3. Информационные технологии

6.3.1 Перечень лицензионного и свободно распространяемого программного обеспечения

1.	Операционная система Microsoft Windows 10 Professional Russian. Контракт № ПР/ФЕН/15/18 от 23.10.2015 г., договор № Пр/16/6 от 05 апреля 2016 г.
2.	Операционная система Microsoft Windows Professional 7 Russian. Лицензия №48497058 от 13.05.2011 г., договор № Пр/16/6 от 05 апреля 2016 г.
3.	Программное обеспечение Microsoft Office 2013 Professional. Контракт № 405535 от 2 ноября 2015 года, контракт № ПР/ФЕН/15/18 от 23.10.2015 г.
4.	Комплексная система антивирусной защиты Kaspersky Endpoint Security для бизнеса – стандартный Russian Edition. 500-999 Node 2 year Educational Renewal License. Лицензия № 13С8-190514-084943-783-1256 от 15.05.2019
5.	Браузеры Google Chrome, Mozilla, Opera. Свободно распространяемое ПО
6.	Файловый архиватор 7z. Свободно распространяемое ПО
7.	Программа просмотра файлов формата RPD Adobe Acrobat Reader DC. Свободно распространяемое ПО

6.3.2 Перечень информационных справочных систем и профессиональных баз данных

1.	Портал «Информационно-коммуникационные технологии в образовании» (http://www.ict.edu.ru)
2.	Полнотекстовый архив ведущих западных научных журналов на российской платформе Национального электронно-информационного консорциума (НЭИКОН)(http://neicon.ru)

7. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

Ауд.	Назначение	Оборудование и технические средства обучения	Вид
4-305	Компьютерный класс	аудиоколонки для проектора и интерактивной доски, аудиоколонки учебные, интерактивная доска, компьютеры, кондиционер, маркерная доска, проектор, столы компьютерные, столы учебные	Лек
4-307	Компьютерный класс	аудиоколонки, компьютеры, кондиционер, маркерная доска, столы компьютерные, столы учебные, телевизор	Лаб
4-306	Компьютерный класс	аудиоколонки для проектора и интерактивной доски, интерактивная доска, компьютеры, кондиционер, маркерная доска, проектор, столы компьютерные, столы учебные	Лек
4-338	Учебная аудитория	аудиоколонки, доска учебная, кондиционер, проектор, сабвуфер, стол преподавателя, столы учебные, стул преподавателя, экран настенный	Лек
4-305	Компьютерный класс	аудиоколонки для проектора и интерактивной доски, аудиоколонки учебные, интерактивная доска, компьютеры, кондиционер, маркерная доска, проектор, столы компьютерные, столы учебные	Лаб
4-305	Компьютерный класс	аудиоколонки для проектора и интерактивной доски, аудиоколонки учебные, интерактивная доска, компьютеры, кондиционер, маркерная доска, проектор, столы компьютерные, столы учебные	Зачёт
4-306	Компьютерный класс	аудиоколонки для проектора и интерактивной доски, интерактивная доска, компьютеры, кондиционер, маркерная доска, проектор, столы компьютерные, столы учебные	Лаб
4-306	Компьютерный класс	аудиоколонки для проектора и интерактивной доски, интерактивная доска, компьютеры, кондиционер, маркерная доска, проектор, столы компьютерные, столы учебные	Зачёт
4-307	Компьютерный класс	аудиоколонки, компьютеры, кондиционер, маркерная доска, столы компьютерные, столы учебные, телевизор	Зачёт
4-304	Лекционный с мультимедийным комплексом	доска учебная, проектор, стол преподавателя, столы учебные, стул преподавателя, экран	Лек
4-304	Лекционный с мультимедийным комплексом	доска учебная, проектор, стол преподавателя, столы учебные, стул преподавателя, экран	Экзамен

8. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)

Дисциплина «Программирование» направлена на формирование у студентов знаний, умений и владений в области

программирования на одном из языков процедурной парадигмы. В результате изучения данной дисциплины студенты должны научиться решать задачи с помощью ЭВМ, раскладывать сложные задачи на независимые подзадачи, представлять, как можно использовать возможности компьютера для решения прикладных задач. Преподавателю необходимо провести систематизацию и выравнивание знаний студентов в области программирования, поскольку они могут сильно варьироваться вследствие того, что часть студентов обучалась по базовому, а часть – по профильному курсу предмета «Информатика и ИКТ» в среднем звене школы.

При изучении темы «Массивы» важно уметь описывать массивы как пользовательский тип данных, отслеживать соответствие между выделенной для массива памятью и параметрами массивов, как переменных. Студенты должны четко представлять, что в Си возможно использование только одномерных массивов, а многомерные массивы – всего лишь форма представления данных. Возможность генерации, вывода и т.п. элементов массива необходимо оформлять в виде функций (с параметрами) пользователя. На данном этапе изучаются простейшие алгоритмы сортировки одномерного массива: простыми обменами («пузырек»), простым выбором, простыми включениями («вставка и сдвиг»).

Тема «Ввод-вывод символьных данных. Внутренние коды и упорядоченность символов. Строковые константы» требует особого внимания с целью отработки навыков задания значений строк (переменных и постоянных), изучения стандартных функций по работе со строками, встроенных в модули `stdio.h`, `string.h`, `stdlib.h`.

При изучении указателей следует четко разделять работу с указателями-переменными и указателями-константами. Обращать внимание на нулевой указатель, способы инициализации значений указателя, операцию получения адреса, преобразования типов, аддитивные операции. Соотношение имени массива, указателя типа базового для элементов массива, применение операции `&` для имени массива. Навыки использования ближних, дальних, нормализованных и сегментных указателей не отрабатываются. При изучении указателей надо различать особенности использования указателей для обозначения типа возвращаемого функцией значения, имени функции, как элемента сигнатуры функции.

Приводятся типовые примеры использования функций с переменным количеством параметров (иногда тип параметров также неизвестен). Функции выделения и освобождения памяти (`calloc`, `malloc`, `free` или `new`, `delete`) позволяют работать с динамической памятью, а доступ к элементам массива производится через указатели по правилам адресной арифметики. При изучении файлов необходимо отработать навык использования стандартных функций для открытия и закрытия потока и файла, функций по работе с файлами на диске. Особое внимание следует уделять форматированному выводу информации различных типов данных. Основными считать функции, описанные в модуле `stdio.h`, `stdlib.h` и т.д.

Тема «Препроцессорные средства обработки программ. Общие принципы механизма обработки исключений» требует формирования представления о стадиях препроцессорной обработки, видах и назначениях препроцессорных команд, способах программной обработки исключений. Особое внимание следует обратить на возможность определения команд или их последовательностей через директиву `#define`; заголовочные файлы и модули пользователя. Возможности обработки исключений должны быть даны в ознакомительном порядке, а дальнейшее их использование может быть определено личными симпатиями студентов.

При изучении темы «Рекурсия как метод решения задач» важно уметь описывать и осуществлять реализацию каждого этапа рекурсивной триады. Выделяем умение оптимальной параметризации рекурсивных функций. Проблемное обучение реализуется на этапе осмысления явления рекурсивного взрыва. Разрешением проблемы является введение понятия рекурсии с динамической базой. Примерами анализа и тестирования данного вида рекурсии следует считать рекурсивные функции вычисления чисел Фибоначчи и биномиальных коэффициентов.

Тема «Структуры данных» рассматривается как с формализованной точки зрения на классических примерах, так и с прикладной в части построения структур данных на основе математических объектов (комплексные числа) и чистых прикладных задач обработки данных в прототипах СУБД. Студенты должны четко представлять, что в Си возможно использование массивов с использованием структуры как базового типа данных и их (структур) влияние на дальнейшее развитие сущностей и объектов программирования.

Тема «Средства обобщенного программирования» рассматривается с точки зрения развития профессиональных навыков программиста в части работы с шаблонами и перегрузкой функций как внутри процедурного, так и с перспективой использования данного инструментария в объектно-ориентированном программировании.

Тема «Ввод-вывод и исключения» требует особого внимания с целью приобретения навыков обработки строк (переменных и постоянных), как элементов базового класса `string`. Параллельно с этим рассматриваются класс ввода-вывода `ios` и его представители `cin` `cout`. Особняком стоит вопрос о формировании навыков работы с потоками как с классами ввода-вывода. Исключения тоже представляют собой класс, а значит, – имеют собственные методы доступа к данным.

Преподавание дисциплины должно включать в себя следующие образовательные технологии:

- 1) Организация лекций с использованием презентаций, выполненных с использованием мультимедийных технологий;
 - 2) Использование в ходе лабораторных работ дидактических материалов в виде: опорных конспектов по теоретической составляющей занятий, файлов с примерами программ и т.п.
 - 3) Использование ресурсов LMS MOODLE с целью организации процесса систематизации, приобретения и контроля знаний;
 - 4) Организация выполнения индивидуальных заданий с целью усиления приобретения навыков коллективного и самостоятельного построения моделей, разработки алгоритмов и написания программ;
- Организация презентации индивидуальных и групповых заданий с целью формирования навыка владения профессиональной терминологией, грамотного выстраивания цепочки высказываний для доказательства обоснованности построенных моделей предметной области и правильности отражения реальных свойств объектов.