

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«УДМУРТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
ФИЛИАЛ ФГБОУ ВО «УДГУ» В Г. ВОТКИНСКЕ
СРЕДНЕЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАНИЕ

«УТВЕРЖДАЮ»

Зам. директора по УМР



Е.Н. Бралгина
«23» марта 2023г.



РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

ОП.05 Алгоритмизация и программирование

09.02.07 «Информационные системы и программирование»

Квалификация выпускника

Специалист по информационным системам

Утверждена на заседании кафедры «Информационных и инженерных технологий»	Протокол №7 от 14.03.23		Заведующий кафедрой О.В. Мамрыкин
Утверждена на заседании научно-методического совета	Протокол №3 от 21.03.23		Председатель Е.Н. Бралгина

Воткинск 2023г.

1. ПАСПОРТ РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ **«ОП.05 Алгоритмизация и программирование»**

1.1. Область применения рабочей программы

Рабочая программа дисциплины является частью основной образовательной программы в соответствии с ФГОС для специальности: 09.02.07 Информационные системы и программирование, для обучающихся очной формы обучения

1.2. Место дисциплины в структуре основной профессиональной образовательной программы: дисциплина входит в профессиональный цикл, как общепрофессиональная дисциплина.

1.3. Цель и планируемые результаты освоения дисциплины:

Целью изучения дисциплины «Алгоритмизация и программирование» является подготовка выпускников к деятельности, связанной с разработкой программного обеспечения для решения профессиональных задач.

Задачи:

- ознакомление с теоретическими основами программирования;
- изучение основ алгоритмизации;
- изучение средств описания данных и средств описания действий языков программирования;
- овладение навыками программирования;
- освоение современных сред создания программных продуктов.

В результате освоения дисциплины студент должен **уметь:**

- Разрабатывать алгоритмы для конкретных задач.
- Использовать программы для графического отображения алгоритмов.
- Определять сложность работы алгоритмов.
- Работать в среде программирования.
- Реализовывать построенные алгоритмы в виде программ на конкретном языке программирования.
- Оформлять код программы в соответствии со стандартом кодирования.
- Выполнять проверку, отладку кода программы.

В результате освоения дисциплины студент должен **знать:**

- Понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции.
- Эволюцию языков программирования, их классификацию, понятие системы программирования.
- Основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти.
- Подпрограммы, составление библиотек подпрограмм
- Объектно-ориентированную модель программирования, основные принципы объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляции и полиморфизма, наследования и переопределения

Дисциплина способствует формированию следующих общих и профессиональных компетенций:

Код	Наименование компетенций
ОК 01.	Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам;
ОК 02.	Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности;
ОК 04.	Эффективно взаимодействовать и работать в коллективе и команде;
ОК 05.	Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;
ОК 09.	Пользоваться профессиональной документацией на государственном и иностранном языках.
ПК 2.2	Выполнять интеграцию модулей в программное обеспечение.
ПК 2.3	Выполнять отладку программного модуля с использованием специализированных программных средств.
ПК 2.4	Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.
ПК 2.5	Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.
ПК 3.1	Осуществлять ревьюирование программного кода в соответствии с технической документацией.
ПК 3.2	Выполнять процесс измерения характеристик компонент программного продукта для определения соответствия заданным критериям.
ПК 3.4	Проводить сравнительный анализ программных продуктов и средств разработки, с целью выявления наилучшего решения согласно критериям, определенным техническим заданием.
ПК 5.3	Разрабатывать подсистемы безопасности информационной системы в соответствии с техническим заданием.
ПК 5.4	Производить разработку модулей информационной системы в соответствии с техническим заданием.

2. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

Количество часов на освоение программы дисциплины:

максимальной учебной нагрузки студента 252 часа, в том числе: обязательной учебной нагрузки обучающегося во взаимодействии с преподавателем 234 час; самостоятельной работы студента 18 час.

В результате освоения *вариативной части* дисциплины студент должен **уметь**:

- Разрабатывать алгоритмы для конкретных задач.
- Использовать программы для графического отображения алгоритмов.
- Определять сложность работы алгоритмов.
- Работать в среде программирования.
- Реализовывать построенные алгоритмы в виде программ на конкретном языке программирования.
- Оформлять код программы в соответствии со стандартом кодирования.
- Выполнять проверку, отладку кода программы.

В результате освоения *вариативной части* дисциплины студент должен **знать**:

- Понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции.

- Эволюцию языков программирования, их классификацию, понятие системы программирования.
- Основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти.
- Подпрограммы, составление библиотек подпрограмм.
- Объектно-ориентированную модель программирования, основные принципы объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляции и полиморфизма, наследования и переопределения

2.1. Объем учебной дисциплины и виды учебной работы

Очная форма обучения

<i>Вид учебной работы</i>	<i>Объем часов</i>
Общий объем образовательной программы (всего)	252
Нагрузка во взаимодействии с преподавателем (всего)	234
в том числе:	
лекции (уроки)	90
практические занятия	
лабораторные занятия	144
консультация	
Самостоятельная работа обучающегося (всего)	18
Промежуточная аттестация в форме дифференцированного зачета в 3 семестре, экзамена в 4 семестре.	

2.2. Тематический план и содержание учебной дисциплины

Наименование тем	Содержание учебного материала и формы организации деятельности обучающихся	Лек.	Лаб.	Сам.	Коды компетенций, формированию которых способствует элемент программы
Тема 1	<p>Что понимается под термином язык программирования? Что называется исходным кодом программы? Что такое интерпретатор и компилятор?</p> <p>Что называется алгоритмом? Что такое алгоритмизация? Основные требования к составлению алгоритмов. Основные этапы разработки программы</p> <p>Основные алгоритмические структуры (линейная последовательность действий, ветвление, циклы). Их назначение. Примеры применения.</p> <p>Графическое представление алгоритмов. Блок-схемы. Назначение блок-схем. Достоинства и недостатки представления алгоритмов в виде блок-схем.</p> <p>Лабораторные занятия: Знакомство со средой программирования Microsoft VisualStudio. Создание проекта. Свойства проекта. Текстовый редактор кода.</p>	6			ПК 3.1,ПК 3.2,ПК 3.4, ПК 2.2,ПК 2.3,ПК 2.4,ПК 2.5,ПК 5.3,ПК 5.4,ОК 09,ОК 05,ОК 04,ОК 02,ОК 01,
			5		

	<p>Обозреватель объектов и обозреватель решений. Структура проекта, файлы проекта.</p> <p>Самостоятельные работы. Подготовка к теоретическому опросу, проработка конспектов лекций, учебной и научно-исследовательской литературы. Поиск информации в Интернет – ресурсах по теме задания.</p> <p>Построить алгоритмы различной конструкции по индивидуальным заданиям.</p> <p>Проанализировать примеры алгоритмов (определение НОД 2-х чисел, раскладка шаров по цвету, выбор большего числа).</p> <p>Составление конспекта на тему «История развития термина алгоритма». Изучить приоритет выполнения действий в выражениях, подготовить конспект.</p>			1	
Тема 2	<p>Алфавит языка C++. Переменные. Правила составления имен переменных. Основные типы данных языка C++, их назначение. Тип данных auto. Особенности использования.</p> <p>Основные сведения об операциях в языке C++. Операция sizeof(). Приоритет выполнения операций. Выражения. Правила составления выражений.</p> <p>Описание переменных. Директива #define. Модификатор const. Оператор присваивания. Тернарная операция «?».</p> <p>Арифметические операции. Операции отношения. Логические операции. Битовые операции. Арифметические сдвиги. Бинарные и унарные операции.</p> <p>Лабораторные работы. Написание простой консольной программы на языке c++. Знакомство со структурой программы. Компиляция программы. Русификация вывода в консоль.</p> <p>Самостоятельные работы. Составление выражений. Составление таблиц истинности логических операций. Изменение приоритета операций. Составление алгоритмов простых программ (решение линейного уравнения, квадратного уравнения. Вычисление значений кусочно-линейных функций).</p>	8		5	ПК 3.1,ПК 3.2,ПК 3.4 ,ПК 2.2,ПК 2.3,ПК 2.4,ПК 2.5,ПК 5.3,ПК 5.4,ОК 09,ОК 05,ОК 04,ОК 02,ОК 01,
Тема 3	<p>Общие сведения об операторах языка C++. Простой оператор. Составной оператор. Область видимости переменных. Метки в C++. Оператор безусловного перехода goto.</p> <p>Общая структура программы на языке C++. Назначение разделов программы. Функция main. Пример простой программы.</p> <p>Условный оператор. Назначение условного оператора. Структура условного оператора. Разновидности условного оператора.</p> <p>Оператор выбора switch. Назначение. Отличие от условного оператора. Примеры использования. «Подводные камни» и ошибки при использовании оператора выбора.</p>	8		1	ПК 3.1,ПК 3.2,ПК 3.4ПК 2.2,ПК 2.3,ПК 2.4,ПК 2.5,ПК 5.3,ПК 5.4,ОК 09,ОК 05,ОК 04,ОК 02,ОК 01,
	Лабораторные работы. Реализация программы вычисления линейного уравнения. Реализация программы решения квадратного уравнения. Реализация вычисления кусочно-линейной			5	

	функции. Реализация простого консольного калькулятора. Тестирование и пошаговая отладка программы. Отслеживание значений переменных на каждом шаге. Точки останова.				
	Самостоятельные работы. Пошаговое выполнение программ. Построение сложных логических выражений в условных операторах.			1	
Тема 4	Циклы в C++. Назначение циклов. Типы циклов. Цикл с предусловием while. Операторы break и continue.	8			ПК 3.1,ПК 3.2,ПК 3.4, ПК 2.2,ПК 2.3,ПК 2.4,ПК 2.5,ПК 5.3,ПК 5.4,ОК 09,ОК 05,ОК 04,ОК 02,ОК 01,
	Циклы в C++. Назначение циклов. Типы циклов. Цикл с постусловием do ... while.				
	Циклы в C++. Назначение циклов. Типы циклов. Цикл с параметром for.				
	Лабораторные работы. Вычисление суммы ряда. Поиск минимума/максимума. Вычисление факториала, арифметической и геометрической прогрессии.		5		
	Самостоятельные работы. Вычисление функций, рядов в соответствии с заданиями. Анализ работы циклических алгоритмов. Особенности использования циклов, основные подводные камни использования циклов. Искусственный цикл с использованием условного оператора и оператора перехода.			2	
Тема 5	Структурированные типы данных. Массивы. Одномерные и многомерные массивы. Как в памяти располагается многомерный массив? Правила формирования индексов элементов массивов. Выход индекса за границы массивов, последствия этого и как избежать?	6			ПК 3.1,ПК 3.2,ПК 3.4,ПК 2.2,ПК 2.3,ПК 2.4,ПК 2.5,ПК 5.3,ПК 5.4,ОК 09,ОК 05,ОК 04,ОК 02,ОК 01,
	Структурированные типы данных. Цепочки (массивы) символов (строки). Представление строки в памяти. Операции со строками. Присваивание строк, сравнение строк, копирование строк и т.д. Управляющие символы.				
	Структурированные типы данных. Структуры. Описание структур. Назначение структур. Примеры структур.				
	Лабораторные работы: Создание и инициализация массива. Использование циклов при обработке массивов. Поиск минимума и максимума в массиве. Расчет определителя матрицы. Транспонирование матрицы. Поиск суммы по строкам и столбцам матрицы. Поиск среднего арифметического значения элементов матрицы. Создание приложения по обработке экспертных данных методом парных сравнений.			8	
	Самостоятельные работы. Обзор алгоритмов обработки данных в массивах. Стандартные функции по работе с массивами. Копирование массивов. Подготовка с проверочным работам.				
Тема 6	Функции в C++. Назначение функций. Описание функций, реализация функций. Имя функции. Тип возвращаемого результата. Параметры функции. Возврат результата из функции.	6			ПК 3.1,ПК 3.2,ПК 3.4, ПК 2.2,ПК 2.3,ПК 2.4,ПК 2.5,ПК 5.3,ПК 5.4,ОК 09,ОК
	Прототипы функций. Назначение. Параметры по умолчанию и правила их применения.				

	<p>Перегрузка функций. Назначение перегрузки. Правила перегрузки.</p> <p>Передача аргументов в функцию. Передача по значению. Передача по ссылке. В чем разница в механизме и в описании аргументов, передаваемых по ссылке и по значению? Передача массивов в функции.</p> <p>Лабораторные работы. Создание пользовательских функций по обработке данных в массивах (сумма, среднее, минимум, максимум, количество элементов с заданными условиями) с несколькими перегрузками в зависимости от типа обрабатываемых данных. Создание пользовательских функций управления консолью, курсором, цветом символов. Работа со стандартными функциями библиотек.</p> <p>Самостоятельная работа: знакомство со стандартными функциями библиотек с++. Написание реферата по обзору часто используемых функций.</p>		8	2	05,ОК 04,ОК 02,ОК 01,
Тема 7	<p>Динамическая память (куча). Указатели в С++. Назначение указателей. Определение указателей. Типы указателей. Операция разадресации. Адресная арифметика. Специальное значение NULL (nullptr).</p> <p>Операции с динамической памятью. Динамические переменные. Операции выделения и освобождения памяти (new и delete). Алгоритм работы с динамической памятью.</p> <p>Динамические массивы. Одномерные динамические массивы, Двухмерные динамические массивы. Особенности и основные приемы работы.</p> <p>Лабораторные работы: Решение задачи поиска корней квадратного уравнения, используя указатели на переменные. Размещение в динамической памяти большого массива и работа с ним (реализация функций поиска: сумма, среднее, минимум, максимум, количество элементов с заданными условиями). Передача указателя массива в функцию. Возврат указателя на массив из функции.</p> <p>Самостоятельная работа. Разобрать типовые ошибки работы с указателями и их последствия. Разработать программу (на выбор) по работе с динамическими переменными. Заполнение динамического массива расчетными значениями из ряда Тейлора.</p>	6	8	1	<p>ПК 3.1,ПК 3.2,ПК 3.4, ПК 2.2,ПК 2.3,ПК 2.4,ПК</p> <p>2.5,ПК 5.3,ПК 5.4,ОК 09,ОК 05,ОК 04,ОК 02,ОК 01,</p>
Тема 8	<p>Ввод-вывод в консоль в С++. Стандартные потоки ввода-вывода cin и cout. Правила использования. Основные операции ввода вывода. Спецификаторы и модификаторы вывода.</p> <p>Обработка событий клавиатуры в консольном приложении. Получение кода нажатой клавиши. Получение расширенного кода клавиши (например клавиши управления курсором, функциональные клавиши). Как определить была ли нажата какая –либо клавиша на клавиатуре, или нет?</p> <p>Потоковый ввод-вывод в файлы. Потоки</p>	6			<p>ПК 3.1,ПК 3.2,ПК 3.4, ПК 2.2,ПК 2.3,ПК 2.4,ПК</p> <p>2.5,ПК 5.3,ПК 5.4,ОК 09,ОК 05,ОК 04,ОК 02,ОК 01,</p>

	<p>ifstream, ofstream,fstream. Их назначение и отличие. Типовая процедура работы с файлом (чтение, запись), последовательность действий.</p> <p>Основные методы для работы с файловыми потоками. Методы open(), close(), eof(), seekg(), tellg, get(), getline(), read(), write(). Режимы доступа к файлу.</p> <p>Типы файлов. Текстовые файлы, бинарные файлы. Отличия в работе с этими файлами. Методы чтения и записи данных в бинарный файл.</p> <p>Лабораторные работы: Форматированный вывод табличных значений в консоль. Обработка событий клавиатуры с ожиданием нажатия клавиш и без такового. Организация консольного пользовательского интерфейса (меню), обработка действий пользователя при работе с меню. Работа с файловыми потоками ввода-вывода. Чтение текстовых и бинарных файлов. Запись текстовых и бинарных файлов. Перенаправление ввода-вывода из консоли в файл. Написание программы «Записная книжка» с произвольным доступом к записям базы данных. Чтение сведений о характеристиках изображения, хранимых в файлах Windows Bitmap. Вывод изображения, хранимого в монохромном файле Windows Bitmap на экран. Написание постой консольной игры «Лабиринт» - игрок собирает монеты на уровне и переходит на следующий уровень. Карты уровня хранятся в текстовых файлах.</p> <p>Самостоятельная работа. Сравнение файлового ввода-вывода и потокового ввода-вывода. Запись и чтение матриц в файлы. Сохранение пользовательских данных в файлы и чтение пользовательских данных из файлов. Обзор популярных форматов файлов хранения данных. Обзор простых консольных игр и методов их реализации на языке C++</p>		12		1
Тема 9	<p>Понятие исключения (exception) в программе. Причины возникновения исключений. Обработка исключений. Операторы try...catch. Оператор throw. Стандартные классы исключений.</p> <p>Лабораторные работы. Создание функции вычисления корней квадратного уравнения, выбрасывающей исключение при невозможности их вычисления и обработка данного исключения. Создание функции поиска заданного элемента в массиве, выбрасывающей исключение out_of_range при задании неверного индекса элемента, с которого начать поиск. Создание библиотеки пользовательских функций с выбрасыванием и обработкой исключений.</p> <p>Самостоятельная работа. Программные и аппаратные исключения в c++. Сравнение способов обработки этих исключений. Анализ – когда и в каких случаях целесообразней из функции выбросить исключение, а когда вернуть специальное значение (например NULL или -1). Сравнение данных способов по</p>	2			2
			6		ПК 3.1,ПК 3.2,ПК 3.4, ПК 2.2,ПК 2.3,ПК 2.4,ПК 2.5,ПК 5.3,ПК 5.4,ОК 09,ОК 05,ОК 04,ОК 02,ОК 01,

	быстродействию. Стандартные классы исключений.				
Тема 10	Алгоритмы обработки данных. Поиск данных в массиве. Поиск данных в несортированном массиве. Поиск данных в сортированном массиве.	8	14	1	ПК 3.1,ПК 3.2,ПК 3.4, ПК 2.2,ПК 2.3,ПК 2.4,ПК 2.5,ПК 5.3,ПК 5.4,ОК 09,ОК 05,ОК 04,ОК 02,ОК 01,
	Алгоритмы обработки данных. Сортировка данных. Назначение сортировки. Алгоритмы сортировки. Пузырьковая сортировка.				
	Алгоритмы обработки данных. Сортировка данных. Назначение сортировки. Алгоритмы сортировки. Выборочная сортировка.				
	Алгоритмы обработки данных. Сортировка данных. Назначение сортировки. Алгоритмы сортировки. Быстрая сортировка (quick sort).				
	Алгоритмы обработки данных. Парсинг строк. Для чего это нужно? Особенности парсинга. Пример парсинга строки.				
	Деревья. Двоичные деревья. Задачи на деревьях. Иерархические структуры. Графы.				
	Понятие рекурсии. Рекурсивные алгоритмы. Примеры рекурсивных алгоритмов. Достоинства и недостатки рекурсивных алгоритмов. Рекурсивные алгоритмы поиска, сортировки и обхода дерева.				
	Лабораторные работы. Разработка программ последовательного и бинарного поиска. Сравнение временных характеристик таких алгоритмов. Особенности при реализации бинарного поиска. Понятие левой и правой границ поиска. Реализация алгоритмов сортировки данных (пузырьковая сортировка, выборочная сортировка, сортировка Хоара (быстрая сортировка). Разработка алгоритма парсинга строк (из заданной текстовой строки с координатами точек, разделенных запятой, выделить пары координат X,Y и вернуть двумерный массив таких координат. Из заданного текстового файла выделить все слова и вернуть массив строк со словами без повторов). Провести частотный анализ повторяемости слов в тексте. Реализация рекурсивного алгоритма бинарного поиска. Реализация рекурсивного алгоритма сортировки Хоара. Реализация рекурсивного алгоритма подбора пароля из заданного алфавита и заданной длины.				
Самостоятельные работы. Разобрать алгоритмы сортировки слиянием, сортировки пирамидой. Построение двоичного дерева поиска. Построение алгоритма поиска в бинарном дереве. Обход дерева. Написание рефератов по одному из алгоритмов.					
Тема 11	Динамические структуры данных. Списки. Линейные списки. Циклические списки. Однонаправленные и двунаправленные списки. Назначение и область применения списков. Достоинства и недостатки списков.	8			ПК 3.1,ПК 3.2,ПК 3.4, ПК 2.2,ПК 2.3,ПК 2.4,ПК 2.5,ПК 5.3,ПК 5.4,ОК 09,ОК 05,ОК 04,ОК
	Динамические структуры данных. Стеки. Особенности организации стеков. Область применения стеков. Достоинства и недостатки				

	стеков.				02,ОК 01,
	Динамические структуры данных. Очереди. Особенности организации очереди. Область применения очереди. Достоинства и недостатки очереди.				
	Лабораторные работы: Организация базы данных товаров в виде линейного однонаправленного списка. Реализация функций по добавлению, удалению, вставке, поиску элементов в списке. Организация стека в виде связанного списка. Реализация алгоритма компьютерной графики – заливка внутренне- и гранично-определенной области с затравкой на основе стека. Моделирование системы массового обслуживания в многофункциональном центре в виде очереди (связанный список). Визуализация движения очереди на экране. Расчет показателей системы массового обслуживания.		20		
	Самостоятельные работы: Разработка комплексной программы на основе связанных списков (по вариантам).			1	
Тема 12	Объектно-ориентированное программирование (ООП). История развития ООП. Базовые понятия ООП: класс, его свойства и методы. Различие между классом и объектом.	8			ПК 3.1,ПК 3.2,ПК 3.4, ПК 2.2,ПК 2.3,ПК 2.4,ПК
	Основные принципы ООП: инкапсуляция, наследование, полиморфизм. Примеры создания классов. Стандартные классы, используемые в C++.				2.5,ПК 5.3,ПК 5.4,ОК 09,ОК 05,ОК 04,ОК 02,ОК 01,
	Пример создания иерархии классов на примере графических объектов (точка, окружность, прямоугольник). Управление объектами. Показательный пример преимущества использования классов в программировании.				
	Шаблоны классов. Пример создания пользовательского шаблона классов.				
	Лабораторные работы: Создание иерархии классов на примере графических примитивов (точка, окружность, прямоугольник). Управление объектами. Движение по экрану. Рефакторинг кода игровой программы «Лабиринт» с использованием объектно-ориентированного подхода.		20		
	Самостоятельные работы: Обзор возможностей объектно-ориентированного программирования в разных языках (C++, C#, Python, Visual Basic, Object Pascal, Java и т.п.)			1	
Тема 13	Стандартные контейнерные классы. Строки (string), достоинства и недостатки по сравнению с массивами символов (char[]). Описание строк. Основные операции и методы работы со строками.	6			ПК 3.1,ПК 3.2,ПК 3.4,ПК 2.2,ПК 2.3,ПК 2.4,ПК
	Стандартные контейнерные классы – array, list, vector. Создание и использование данных контейнеров. Понятие итератора. Назначение итераторов. Доступ к элементам, добавление, вставка, удаление и изменение элементов.				2.5,ПК 5.3,ПК 5.4,ОК 09,ОК 05,ОК 04,ОК 02,ОК 01,
	Ассоциативный контейнер map. Назначение. Что такое ключ, что такое значение. Добавление и вставка элементов в контейнер. Удаление элементов. Доступ к элементу по ключу.				

	Ассоциативный контейнер multimap. Назначение. Отличие от map. Что такое ключ, что такое значение. Добавление и вставка элементов в контейнер. Удаление элементов. Доступ к элементу по ключу.				
	Лабораторные работы: Организация базы данных товаров с использованием стандартных контейнеров (vector, array, list). Сравнительная характеристика алгоритмов, использующих массив символов и класс string. Создание простой базы данных студентов и изучаемых дисциплин с использованием контейнеров map и multimap. Парсинг предложений с использованием контейнера map. Задача криптоанализа – изучить частоту появления символов и их комбинаций русского и английского текстов. Построить гистограмму частот появления символов в консоли.		20		
	Самостоятельная работа: Стандартная библиотека шаблонов классов STL. Основные и компоненты и алгоритмы.			1	
Тема 14	Основы создания многопоточных приложений. Понятие процесса и потока. Что такое поток (thread) в плане многозадачности? Как создать поток в своей программе? Как запустить фрагмент кода в программе в отдельном потоке? Как дождаться завершения потока? Когда использования многопоточности оправдано в программе? Синхронизация потоков. Методы и способы синхронизации.	4			ПК 3.1, ПК 3.2, ПК 3.4, ПК 2.2, ПК 2.3, ПК 2.4, ПК
	Лабораторные работы: Создание простого многопоточного приложения, выводящего на консоль результат вычисления некоторой функции (факториал, числа Фибоначи и др.), каждая из которых реализована в собственном потоке. Сравнительная характеристика 3-х алгоритмов сортировки, выполняемых в разных потоках с визуализацией результатов. Организация фоновой обработки данных пользователя с использованием многопоточности. Моделирование системы массового обслуживания с использованием 2-х потоков с синхронизацией. Анализ эффективности распараллеливания алгоритма подбора пароля на многоядерных процессорах.		8		2.5, ПК 5.3, ПК 5.4, ОК 09, ОК 05, ОК 04, ОК 02, ОК 01,
	Самостоятельные работы: Обзор организации многопоточности в разных языках программирования (C++, C#, Python, Visual Basic, Object Pascal, Java и т.п.)			1	
Всего		90	144	18	252

3. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ

3.1. Требования к минимальному материально-техническому обеспечению

Лаборатория Программирования и баз данных;	Доска ученическая 5-ти секционная Комплект учебной мебели, набор демонстрационного оборудования
--	--

	<p>(проектор, экран), учебно-наглядные пособия (презентации по дисциплине), 16 компьютеров с выходом в сеть Интернет, имеющие неограниченный доступ к электронно – библиотечным системам и электронной информационной образовательной среде филиала</p> <p>Microsoft Office, Microsoft Windows, договор 0313100004015000052-0006194-01/1858 от 30.11.2015, Microsoft Visio договор 0005111053-C-M088078 от 31.01.2011, платформа 1С:Предприятие (версия для обучения, бесплатное ПО) виртуальная машина Virtual Box (бесплатное ПО) операционная система семейства Windows (для установки на вирт.машину) с пакетом встроенного ПО, WAMP-сервер «Denwer» или аналогичный (свободное ПО), пакет управления СУБД MySQL Workbench или аналогичный (свободное ПО), архиватор 7-Zip (свободное ПО), Visual C++ Express Edition, Visual Studio CE (бесплатное ПО)</p>
--	---

3.2. Учебно-методическое и информационное обеспечение дисциплины (модуля)

3.2.1. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины (модуля)

Основная учебная литература:

1. Родыгин, А. В. Информационные технологии. Алгоритмизация и программирование : учебное пособие / А. В. Родыгин. — Новосибирск : Новосибирский государственный технический университет, 2017. — 92 с. — ISBN 978-5-7782-3300-3. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/91212.html>
2. Трофимов, В. В. Основы алгоритмизации и программирования : учебник для среднего профессионального образования / В. В. Трофимов, Т. А. Павловская ; под редакцией В. В. Трофимова. — Москва : Издательство Юрайт, 2021. — 137 с. — (Профессиональное образование). — ISBN 978-5-534-07321-8. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/473347>
3. Тюльпинова, Н. В. Алгоритмизация и программирование : учебное пособие / Н. В. Тюльпинова. — Саратов : Вузовское образование, 2019. — 200 с. — ISBN 978-5-4487-0470-3. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/80539.html>

Дополнительная учебная литература:

1. Букунов, С. В. Основы программирования на языке C++ : учебное пособие / С. В. Букунов. — Санкт-Петербург : Санкт-Петербургский государственный архитектурно-строительный университет, ЭБС АСВ, 2015. — 201 с. — ISBN 978-5-9227-0619-3. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <http://www.iprbookshop.ru/63631.html>
2. Вязовик, Н. А. Программирование на Java : учебное пособие для СПО / Н. А. Вязовик. — Саратов : Профобразование, 2019. — 604 с. — ISBN 978-5-4488-0365-9. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <http://www.iprbookshop.ru/86206.html>
3. Забержинский, Б. Э. Программирование. Введение в разработку на C# : учебное пособие / Б. Э. Забержинский, А. Г. Золин. — Самара : Самарский государственный технический университет, ЭБС АСВ, 2017. — 120 с. — ISBN 2227-8397. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <http://www.iprbookshop.ru/90876.html>
4. Иванов, В. Б. Прикладное программирование на C/C++: с нуля до мультимедийных и сетевых приложений / В. Б. Иванов. — Москва : СОЛОН-Пресс, 2018. — 240 с. — ISBN 978-5-91359-308-5. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <http://www.iprbookshop.ru/90397.html>

5. Казанский, А. А. Программирование на Visual C# 2013 : учебное пособие для среднего профессионального образования / А. А. Казанский. — Москва : Издательство Юрайт, 2020. — 191 с. — (Профессиональное образование). — ISBN 978-5-534-02721-1. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <http://www.biblio-online.ru/bcode/452454>
6. Канцедал, С.А. Алгоритмизация и программирование : учеб. пос. для СПО по спец. "Информатика и вычислит. техника" / С.А. Канцедал. - Москва : Форум : ИНФРА-М, 2019 (2013)
7. Кудрина, Е. В. Основы алгоритмизации и программирования на языке C# : учебное пособие для среднего профессионального образования / Е. В. Кудрина, М. В. Огнева. — Москва : Издательство Юрайт, 2019. — 322 с. — (Профессиональное образование). — ISBN 978-5-534-10772-2. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://www.biblio-online.ru/bcode/431505>
8. Малиновская, Е. А. Языки программирования. Часть 1 : лабораторный практикум / Е. А. Малиновская, Р. А. Рыскаленко. — Ставрополь : Северо-Кавказский федеральный университет, 2016. — 103 с. — ISBN 2227-8397. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <http://www.iprbookshop.ru/69449.html>
9. Подбельский, В. В. Программирование. Базовый курс C# : учебник для среднего профессионального образования / В. В. Подбельский. — Москва : Издательство Юрайт, 2020. — 369 с. — (Профессиональное образование). — ISBN 978-5-534-11467-6. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://www.biblio-online.ru/bcode/456697>

3.2.2. Перечень ресурсов информационно-телекоммуникационной сети «Интернет» (далее - сеть «Интернет»), необходимых для освоения дисциплины (модуля)

Электронно-библиотечные системы (ЭБС), обеспечивающие доступ для каждого обучающегося из любой точки, в которой имеется доступ к сети Интернет:

ЭБС "Юрайт";

ЭБС "Лань";

ЭБС IPRbooks;

УдНОЭБ (Удмуртская научно-образовательная электронная библиотека),

обеспечивающая возможность индивидуального доступа каждого обучающегося из любой точки, в которой имеется доступ к сети Интернет (<http://lib.udsu.ru/>).

3.2. 3. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения и информационных справочных систем (при необходимости)

Наименование программного обеспечения
Microsoft Office, Microsoft Windows, договор 0313100004015000052-0006194-01/1858 от 30.11.2015, Microsoft Visio договор 0005111053-С-М088078 от 31.01.2011, платформа 1С:Предприятие (версия для обучения, бесплатное ПО) виртуальная машина Virtual Box (бесплатное ПО) операционная система семейства Windows (для установки на вирт.машину) с пакетом встроенного ПО, WAMP-сервер «Denwer» или аналогичный (свободное ПО), пакет управления СУБД MySQL Workbench или аналогичный (свободное ПО), архиватор 7-Zip (свободное ПО), Visual C++ Express Edition, Visual Studio CE (бесплатное ПО)

4. ФОНДЫ ОЦЕНОЧНЫХ СРЕДСТВ

4.1. Область применения рабочей программы

Рабочая программа дисциплины является частью основной образовательной программы в соответствии с ФГОС для специальности: 09.02.07 Информационные системы и программирование, для обучающихся очной формы обучения

4.2. Объекты оценивания – результаты освоения дисциплины

ФОС позволяет оценить следующие результаты освоения дисциплины в соответствии с ФГОС специальности 09.02.07 Информационные системы и программирование и рабочей программой дисциплины ОП.05 Языки программирования:

<i>Результаты обучения</i>	<i>Критерии оценки</i>	<i>Формы и методы оценки</i>
<i>Перечень знаний, осваиваемых в рамках дисциплины:</i>		
Понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции. Эволюцию языков программирования, их классификацию, понятие системы программирования. Основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти. Подпрограммы, составление библиотек подпрограмм. Объектно-ориентированную модель программирования, основные принципы объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляции и полиморфизма, наследования и переопределения.	«Отлично» - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко. «Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками. «Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных	Оценка в рамках текущего контроля результатов выполнения индивидуальных контрольных заданий, результатов выполнения практических работ, устный индивидуальный опрос. Письменный опрос в форме тестирования
<i>Перечень умений, осваиваемых в рамках дисциплины:</i>		

<p>Разрабатывать алгоритмы для конкретных задач. Использовать программы для графического отображения алгоритмов. Определять сложность работы алгоритмов. Работать в среде программирования. Реализовывать построенные алгоритмы в виде программ на конкретном языке программирования. Оформлять код программы в соответствии со стандартом кодирования. Выполнять проверку, отладку кода программы.</p>	<p>программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки. «Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.</p>	<p>Экспертное наблюдение и оценивание выполнения практических работ. Текущий контроль в форме защиты практических работ</p>
---	---	--

4.3. **Формы контроля и оценки результатов освоения дисциплины**

Контроль и оценка результатов освоения – это выявление, измерение и оценивание результатов освоения дисциплины.

В соответствии с учебным планом специальности 09.02.07 Информационные системы и программирование, рабочей программой дисциплины предусматривается текущий и промежуточный контроль результатов освоения.

Формы текущего контроля

Текущий контроль успеваемости представляет собой проверку усвоения учебного материала, регулярно осуществляемую на протяжении курса обучения.

Текущий контроль результатов освоения дисциплины в соответствии с рабочей программой и тематическим планом происходит при использовании следующих обязательных форм контроля:

- *выполнение и защита практических заданий, решение задач*
- *проверка выполнения самостоятельной работы студентов,*
- *проверка выполнения контрольных работ.*

Во время проведения учебных занятий дополнительно используются следующие формы текущего контроля – *устный опрос, тестирование по темам отдельных занятий, групповая дискуссия.*

Выполнение и защита лабораторных заданий, решение задач. Лабораторные занятия проводятся с целью усвоения и закрепления результатов освоения дисциплины. В ходе занятий обучающиеся учатся *использовать изученные знания и применять различные методы решения задач, анализировать полученные результаты и делать выводы, опираясь на теоретические знания.*

Список лабораторных занятий:

Приведен в таблице раздела 2.2.

Проверка выполнения самостоятельной работы.

Самостоятельная работа направлена на самостоятельное освоение и закрепление результатов обучения.

Самостоятельная подготовка обучающихся по дисциплине предполагает

следующие виды и формы работы:

- подготовка к теоретическому опросу, проработка конспектов лекций, учебной и научно-исследовательской литературы
- поиск информации в Интернет – ресурсах по теме задания;
- выполнение рефератов, докладов по теме задания;
- выполнение расчетно-графических работ, решение типовых задач;
- составление опорных конспектов;
- подготовка к тестированию;
- составление отчета решения исследовательских задач;
- составление сопровождающей документации;
- разработка программ, решение задач;
- разработка алгоритмов;
- выполнение индивидуального проектного задания;
- тестирование и отладка разработанного приложения.

Проверка выполнения контрольных работ. Контрольная работа (тест) проводится с целью проверки результатов обучения и последующего анализа типичных ошибок и затруднений обучающихся в конце изучения темы или раздела.

Варианты некоторых контрольных работ:

Контрольная работа №1

Вариант 1

1. Напишите консольное приложение на языке C++, которое запрашивает у пользователя его имя, а потом выводит на экран приветствие. Например пользователь за запрос программы ввел имя **Маша**. Программа должна напечатать: **Привет, Маша!**

Образец:

Введите ваше имя: Маша Привет, Маша!

2. Какие известные вам типы данных языка C++ позволяют обеспечить хранение и обработку целых чисел в диапазоне от -500 до + 32000
3. Перечислите основные элементы блок-схем алгоритмов и опишите их назначение
4. Напишите на языке C++ функцию, возвращающую сумму чисел в диапазоне от 50 до 100.
5. Каким будет значение переменной **X** при выполнении следующего фрагмента программы:

```
int A=10;  
int B=20;  
int X=A>B?A+B:A-B;
```
6. Вычислить: $-2+(1+2*(3-1))/2*3$
7. Какие из чисел являются целыми? (5 12 7.0 -0.0 -67.9 1E3 45)
8. Имеется две переменные символьного типа a и b, содержащие произвольные символы. Запишите фрагмент программы, позволяющий вывести эти символы на экран в алфавитном порядке.
9. Нужно вывести на экран следующее:
WEC = 56.2 кг
значение веса храниться в переменной X. Запишите оператор, позволяющий выполнить указанное действие.
10. Чему будет равно значение переменной x и y после выполнения следующего фрагмента программы:

```
float x = -6.7;  
float y= x/2;  
if (x>=-10) {x = x + 1; y = x + 2;}  
else  
    x = x-1;  
    y = y-1;
```


Вариант 2

1. Напишите консольное приложение на языке C++, которое запрашивает у пользователя его имя, а потом выводит на экран приветствие. Например пользователь за запрос программы ввел имя **Маша**. Программа должна напечатать: **Маша, привет!**

Образец:

```
Введите ваше имя: Маша
Маша, привет!
```

2. Какие известные вам типы данных языка C++ позволяют обеспечить хранение и обработку целых чисел в диапазоне от -50000 до + 312000
3. Приведите фрагмент блок-схемы алгоритма, реализующий ветвление в алгоритме.
4. Напишите на языке C++ функцию, возвращающую сумму чисел в диапазоне от -50 до 100 с использованием цикла с предусловием while.
5. Каким будет значение переменной **X** при выполнении следующего фрагмента программы:

```
int A=20;
int B=10;
int X=A>B?A+B:A-B;
```
6. Вычислить: $-2+(1*2-(3-1))/(2*3)$
7. Какие из чисел являются числами с плавающей точкой? (5 12 7.0 -0.0 -67.9 1E3 45)
8. Имеется две переменные целого типа a и b, содержащие произвольные числа. Запишите фрагмент программы, позволяющий вывести эти числа на экран в порядке убывания.
9. Нужно вывести на экран следующее:
Маше 19 лет
значение возраста (19) храниться в переменной X. Запишите оператор, позволяющий выполнить указанное действие.
10. Чему будет равно значение переменной x и y после выполнения следующего фрагмента программы:

```
float x = -6.7;
float y= x/2;
if (x>=10) {x = x + 1; y = x + 2;}
else
    x = x-1;
    y = y-1;
```

Вариант 3

1. Напишите консольное приложение на языке C++, которое запрашивает у пользователя его возраст, а потом выводит на экран приветствие. Например пользователь за запрос программы ввел возраст 17. Программа должна напечатать: **Ура! Тебе уже 17 лет!**

Образец:

```
Введите ваш возраст: 17
Ура! Тебе уже 17 лет!
```

2. Какие известные вам типы данных языка C++ позволяют обеспечить хранение и обработку чисел в диапазоне от -50000.0 до + 3120.00
3. Приведите фрагмент блок-схемы алгоритма, реализующий в алгоритме цикл.
4. Напишите на языке C++ функцию, возвращающую произведение чисел в заданном диапазоне от a до b. Значения параметров a и b должны передаваться в функцию.
5. Каким будет значение переменной **X** при выполнении следующего фрагмента программы:

```
int A=20;
int B=10;
int X=A>(B+10)?A*B:A--;
```

6. Запишите на языке C++ выражение $\frac{1}{\left(\frac{f}{k-1} - \frac{f^3}{5+4k}\right)} \cdot k$

7. Какие из значений являются символьными константами, а какие строковыми? ('5' "12" '7' "0.0 - 0.0" -67.9 1E3 "45")
8. Имеется две переменные целого типа a и b, содержащие произвольные числа. Запишите фрагмент программы, позволяющий вывести эти числа на экран, а также произведение этих чисел.
9. Нужно вывести на экран следующее:
До пункта назначения Воткинск осталось 19 км
Название пункта хранится в строке с именем pr, а расстояние в переменной r. Запишите оператор, позволяющий выполнить указанное действие.
10. Чему будет равно значение переменной x и y после выполнения следующего фрагмента программы:


```
float x = 1;
float y = 2;
if (x!=y) {x = y; y =x;}
        else
            x = x-1;
            y = y-1;
```

Вариант 4

1. Напишите консольное приложение на языке C++, которое запрашивает у пользователя его возраст, а потом выводит на экран приветствие. Например пользователь за запрос программы ввел возраст 17. Программа должна напечатать: **Ура! Тебе скоро 18 лет!**

Образец:

Введите ваш возраст: 17
Ура! Тебе скоро 18 лет!

2. Перечислите все переменные целого типа в указанном фрагменте программы:


```
short i; unsigned long d; float r; char t[20];
int f,j; bool b; int* e;
```
3. Приведите фрагмент блок-схемы алгоритма, реализующий в алгоритме линейную последовательность действий.
4. Напишите на языке C++ функцию, возвращающую сумму чисел в заданном диапазоне от a до b. Значения параметров a и b должны передаваться в функцию.
5. Каким будет значение переменной X при выполнении следующего фрагмента программы:


```
int A=20;
int B=10;
int X=A>B?A-B:A--;
```
6. Запишите на языке C++ выражение $(a + b) \cdot \frac{1}{a - b} + 3c$
7. Какие из значений являются символьными константами, а какие строковыми? ('5' "TRUE" '7' "0.0" '- ' "0.0" -67.9 1E3 "45")
8. Имеется две переменные целого типа a и b, содержащие произвольные числа. Запишите фрагмент программы, позволяющий вывести эти числа на экран, а также произведение этих чисел.
9. Нужно вывести на экран следующее:
ВАШ СЧЕТ: 200
Значение счета (200) содержится в переменной X. Запишите оператор, позволяющий выполнить указанное действие.
10. Чему будет равно значение переменной x и y после выполнения следующего фрагмента программы:


```
float x = 1; float y = 2;
if (x=y) {x = y; y =x;}
        else
            x = x-1;
            y = y-1;
```

Вариант 5

1. Напишите консольное приложение на языке C++, которое запрашивает у пользователя два числа, а потом выводит на экран их разность. Например пользователь за запрос программы ввел числа 5 и 7. Программа должна напечатать: **5-7=-2**
Образец:

```
Введите два числа а и b: 5 7
5-7=-2
```

2. Перечислите все переменные вещественного типа в указанном фрагменте программы:
`short i; unsigned long d; float r; char t[20];
int f, j; bool b; int* e;`
3. Приведите блок-схему алгоритма для задачи, указанной в вопросе 1,
4. Напишите на языке C++ функцию, возвращающую сумму чисел в заданном диапазоне от а до b. Значения параметров а и b должны передаваться в функцию. Для реализации алгоритма использовать цикл `do.. while`
5. Каким будет значение переменной **X** при выполнении следующего фрагмента программы:
`int A=20;
int B=10;
int X=A>B?A++:A--;`
6. Запишите на языке C++ выражение $a + b \cdot \frac{1}{a-b} + 3c$
7. Какие из значений являются логическими константами (5 "true" false '7' "0.0" '- ' "0.0" -67.9 TRUE 1E3 "45")
8. Имеется две переменные целого типа а и b, содержащие произвольные числа. Запишите фрагмент программы, позволяющий вывести эти числа на экран, а также сумму квадратов этих чисел.
9. Нужно вывести на экран следующее:
ВАШ СЧЕТ: 200 очков
Значение счета (200) содержится в переменной X. Запишите оператор, позволяющий выполнить указанное действие.
10. Чему будет равно значение переменной x и y после выполнения следующего фрагмента программы:
`float x = 1; float y= 2;
if (x==y) {x = y; y =x;}
else
 x = x-1;
 y = y-1;`

Вариант 6

1. Напишите консольное приложение на языке C++, которое запрашивает у пользователя два числа, а потом выводит на экран их сумму. Например пользователь за запрос программы ввел числа 5 и 7. Программа должна напечатать: **5+7=12**
Образец:

```
Введите два числа а и b: 5 7
5+7=12
```

2. Какие известные вам типы данных языка C++ позволяют обеспечить хранение и обработку целых чисел в диапазоне от 500 до + 62000
3. Приведите фрагмент блок-схемы алгоритма, реализующий ветвление в алгоритме.
4. Напишите на языке C++ функцию, возвращающую произведение чисел в заданном диапазоне от а до b. Значения параметров а и b должны передаваться в функцию.
5. Каким будет значение переменной **X** при выполнении следующего фрагмента программы:
`int A=20;
int B=10;
int X=A>B?A++:A--;`

6. Запишите на языке C++ выражение $\frac{1}{\left(\frac{f}{k-1} - \frac{f^3}{5+4k}\right)} \cdot k$
7. Какие из значений являются символьными константами, а какие строковыми? ('5' "12" '7' "0.0 - 0.0" -67.9 1E3 "45")
8. Имеется две переменные целого типа a и b, содержащие произвольные числа. Запишите фрагмент программы, позволяющий вывести эти числа на экран, а также сумму квадратов этих чисел.
9. Нужно вывести на экран следующее:
ВАШ СЧЕТ: 200
Значение счета (200) содержится в переменной X. Запишите оператор, позволяющий выполнить указанное действие.
10. Чему будет равно значение переменной x и y после выполнения следующего фрагмента программы:
float x = 1; float y = 2;
if (x==y) {x = y; y =x;}
else
x = x+1;
y = y+1;

Вариант 7

1. Напишите консольное приложение на языке C++, которое запрашивает у пользователя два числа, а потом выводит на экран их произведение. Например пользователь за запрос программы ввел числа 5 и 7. Программа должна напечатать: **5*7=35**

Образец:

Введите два числа a и b: 5 7
5*7=35

2. Какие известные вам типы данных языка C++ позволяют обеспечить хранение и обработку целых чисел в диапазоне от 0 до 255?
3. Приведите фрагмент блок-схемы алгоритма, реализующий цикл в алгоритме.
4. Напишите на языке C++ функцию, возвращающую произведение чисел в заданном диапазоне от a до b с шагом 2. Значения параметров a и b должны передаваться в функцию.
5. Каким будет значение переменной X при выполнении следующего фрагмента программы:
int A=0;
int B=0;
int X=A>B?A++:A--;

6. Запишите на языке C++ выражение $\frac{f^3}{\left(\frac{f}{k-1} - \frac{f^3}{5+4k}\right)} \cdot k$

7. Какие из значений являются целыми константами (5 5.2 -1.7E-5 43 0123 -1.1 0x95)
8. Имеется две переменные целого типа a и b, содержащие произвольные числа. Запишите фрагмент программы, позволяющий вывести эти числа на экран, а также разность квадратов этих чисел.
9. Нужно вывести на экран следующее:
ВАШ СЧЕТ:
200
200 должно быть напечатано с новой строки. Значение счета (200) содержится в переменной X. Запишите оператор, позволяющий выполнить указанное действие.
10. Чему будет равно значение переменной x и y после выполнения следующего фрагмента программы:
int x = 1; int y = 2;
if (x==y) {x = ++y; y = ++x;}
else
x++;
++y;

Контрольная работа №2

Вариант 1

1. Опишите указатель на переменную типа `int`. Разместите в динамической памяти такую переменную и присвойте ей произвольное значение. Освободите память, занимаемую данной переменной.
2. Пусть имеется следующее описание: `double F`; Что вернет следующее выражение: `&F` ?
3. Вам необходимо разместить в памяти динамический одномерный массив из `n` элементов типа `int`, заполнить его возрастающими значениями от 0 до `n-1`. Напишите фрагмент программы, которая выполняет данные действия. Значение `n` необходимо запросить от пользователя.
4. Что будет напечатано на экране после выполнения следующего фрагмента программы?

```
int main() { int *e; int c = 543; e=&c; (*e)++; cout << c; }
```

5. Опишите структуру на `c++`, позволяющую хранить следующую информацию о печатающем устройстве: модель устройства (до 20 символов), год выпуска (целое число), серийный номер (до 10 символов), скорость печати листов/мин (целое число), наличие встроенного `wifi` модуля (логическое значение).
6. Напишите фрагмент программы, которая открывает текстовый файл, считывает из него два целых числа и выводит на экран их произведение.
7. Пусть заголовок растрового файла располагается в самом начале файла и описан следующей структурой:

```
struct GF {  
    int width; // ширина изображения  
    int height; // высота изображения  
    char colors; // кол-во цветов  
    bool compression; // true – изображение сжато, false – не сжато.  
};
```

Приведите фрагмент программы, считывающей из файла эту информацию и выводящей на экран характеристики этого изображения.

8. Какой из потоков позволяет считывать вводимые данные с клавиатуры?
а) `cin` б) `cout` в) `ifstream` г) `ofstream` д) `fstream`
9. Прототип функции описан следующим образом:
`void myfunc (int a, char b, int c=0, double d =5.0);`
какие из вариантов вызова функции являются правильными?
а) `int x=myfunc(12,'d');`
б) `myfunc(2,'t');`
в) `myfunc(2,'t',5,5);`
г) `myfunc(2);`
10. Напишите фрагмент программы, выводящей на экран количество символов «А», находящихся в текстовом файле.

Вариант 2

1. Опишите указатель на переменную типа `double`. Разместите в динамической памяти такую переменную и присвойте ей значение, хранящейся в другой переменной. Освободите память, занимаемую данной переменной.
2. Пусть имеется следующее описание: `double *F`; Что содержится в переменной `F`? Что содержится в `*F`?
3. Вам необходимо разместить в памяти динамический одномерный массив из `n` элементов типа `short`, заполнить убывающими значениями от `n` до 1. Напишите фрагмент программы, которая выполняет данные действия. Значение `n` необходимо запросить от пользователя.
4. Что будет напечатано на экране после выполнения следующего фрагмента программы?

```
int main() { int *e; int c = 500; e=&c; c++; cout << *e; }
```

5. Опишите структуру на `c++`, позволяющую хранить следующую информацию о студенте: ФИО (до 40 символов), год рождения (целое число), номер зачетной книжки (до 10 символов), Пол (символьное значение М или Ж), статус студента (отчислен или не отчислен) (логическое значение).
6. Напишите фрагмент программы, которая открывает текстовый файл, записывает в него два целых числа и выводит на экран их разность.
7. Пусть заголовок растрового файла располагается в самом начале файла и описан следующей структурой:

```
struct F1 {  
    int w; // ширина изображения
```

```
int h; // высота изображения
char BitsPerPixel; // кол-во бит, необходимых для представления цвета точки
bool compression; // true – изображение сжато, false – не сжато.
};
```

Приведите фрагмент программы, считывающей из файла эту информацию и выводящей на число байт, необходимых для хранения растровых данных такого изображения. BitsPerPixel может принимать значения 1, 4, 8, 16, 24, 32

8. Какой из потоков позволяет записывать данные в файл?
а) cin б) cout в) ifstream г) ofstream д) fstream
9. Прототип функции описан следующим образом:
int myfunc (int a=0, char b='0', int c=0, double d =5.0);
какие из вариантов вызова функции являются правильными?
а) int x=myfunc(12,'d');
б) int y = myfunc(2,'t');
в) int z = myfunc(2,'t',5,5);
г) int t = myfunc();
10. Напишите фрагмент программы, выводящей на экран количество символов пробела, находящихся в текстовом файле.

Вариант 3

1. Опишите указатель на символьную переменную. Разместите в динамической памяти такую переменную и присвойте ей произвольное значение. Опишите другой указатель на символьную переменную и сделайте так, чтобы оба эти указателя ссылались на одну и ту же выделенную ячейку памяти.
2. Пусть имеется следующее описание: double *F, *P; Выделите память под динамические переменные, инициализируйте их начальными значениями и напечатайте на экране сумму этих значений.
3. Вам необходимо разместить в памяти динамический одномерный массив из n элементов типа char, заполнить убывающими значениями от n до 1 (n<=255). Напишите фрагмент программы, которая выполняет данные действия. Значение n необходимо запросить от пользователя.
4. Что будет напечатано на экране после выполнения следующего фрагмента программы?

```
int main() { int *e; int c = 500; e=&c; c++; cout << *e-1; }
```

5. Опишите структуру на с++, позволяющую хранить следующую информацию об автомобиле: модель (до 20 символов), год выпуска (целое число), гос. номер (до 10 символов), цвет (до 10 символов), мощность двигателя (вещественное число).
6. Напишите фрагмент программы, которая открывает текстовый файл, и записывает в него массив вещественных чисел. Размер массива определите самостоятельно.
7. Какие способы открытия потока для чтения данных из файлы вы знаете? Приведите пример открытия файла для чтения в двоичном формате. Какие функции чтения данных из этого файла вы знаете? Приведите пример.
8. Какой из потоков позволяет вывести данные на экран?
а) cin б) cout в) ifstream г) ofstream д) fstream
9. Приведите общий алгоритм действий, необходимых для работы с файлом.
10. Напишите фрагмент программы, выводящей на экран количество слов, находящихся в текстовом файле. Предполагается, что знаки препинания отсутствуют.

Вариант 4

1. Что такое указатель? Приведите примеры указателей. Каков алгоритм работы с указателями?
2. Пусть имеется следующее описание: char *i, *j; Выделите память под динамические переменные, инициализируйте их начальными значениями и напечатайте на экране эти значения.
3. Вам необходимо разместить в памяти динамический одномерный массив из n элементов типа float, заполнить убывающими значениями от c шагом 0,1. Напишите фрагмент программы, которая выполняет данные действия. Значение n необходимо запросить от пользователя.
4. Что будет напечатано на экране после выполнения следующего фрагмента программы?

```
int main() { int *e; int c = 500; e=&c; c++; cout << *e<<c; }
```

5. Опишите структуру на с++, позволяющую хранить следующую информацию о городе: Название (до 20 символов), год основания (целое число), количество жителей (целое число), является ли город столицей субъекта России (логическое значение).
6. Напишите фрагмент программы, которая открывает текстовый файл, и считывает из него в массив 10 вещественных чисел и выводит их на экран.
7. Какие способы открытия потока для записи данных в файл вы знаете? Приведите пример открытия файла для чтения в текстовом формате. Какие функции чтения данных из этого файла вы знаете? Приведите пример.
8. Какой из потоков позволяет прочитать данные из файла?
а) cin б) cout в) ifstream г) ofstream д)fstream
9. Приведите общий алгоритм действий, необходимых чтения данных из файла.
10. Напишите фрагмент программы, выводящей на экран слова, находящиеся в текстовом файле. Предполагается, что знаки препинания отсутствуют.

Контрольная работа №3

Вариант 1

1. В программировании связанный список это:
 - a. Список динамических элементов, связанных определенной последовательностью их программной обработки и отображения на экране
 - b. динамическая структура данных, которая представляет собой цепочку записей-узлов, в которой каждая запись содержит основные данные и ссылку на следующую запись в цепочке.
 - c. динамическая структура данных, которая представляет собой цепочку записей-узлов, в которой каждая запись может содержать ссылку на следующую запись в цепочке
 - d. Список динамических элементов, способных хранить как числовые, так и строковые данные, и дающий простой механизм их сохранения в файле и чтения из файла.
2. Связные списки бывают:
 - a. односвязные линейные списки;
 - b. односвязные циклические списки;
 - c. односвязные нелинейные списки;
 - d. двусвязные линейные списки;
 - e. двусвязные циклические списки.
 - f. двусвязные нелинейные списки.
 - g. трехсвязные линейные списки
 - h. трехсвязные циклические списки
 - i. трехсвязные нелинейные списки
3. В чем преимущества связанных списков по сравнению с обычными динамическими массивами?
 - a. для хранения данных в связанном списке требуется меньший объем памяти
 - b. доступ к произвольному элементу осуществляется быстрее
 - c. в связанном списке можно легко добавлять новые элементы и удалять ненужные практически без ограничений
 - d. в отличие от массива, можно получить доступ к любому элементу в связанном списке по его индексу.
 - e. преимуществ нет никаких
4. В чем преимущество массива по сравнению со связанным списком?
 - a. для хранения данных в массиве при прочих равных условиях требуется меньший объем памяти
 - b. в массиве можно легко добавлять новые элементы и удалять ненужные практически без ограничений
 - c. доступ к произвольному элементу массива осуществляется быстрее, чем к элементу в связанном списке
 - d. в отличие от связанного списка, в массиве можно получить доступ к любому элементу по его индексу

- e. преимуществ нет никаких
5. Необходимо хранить следующие данные о кабинетах: Номер, Площадь, Количество окон, Количество розеток, Количество радиаторов отопления. Опишите структуру элемента связанного однонаправленного линейного списка, позволяющего хранить данные о произвольном количестве таких кабинетов.
6. Объявите указатель на начало списка из задания №5.
7. В чем основное отличие структур данных «стек» и «очередь»?
- стек является частным случаем связанного списка, а очередь – нет.
 - очередь является частным случаем связанного списка, а стек – нет.
 - очередь использует для доступа к элементам принцип FIFO, добавление элемента производится в конец очереди, а извлечение – из начала очереди
 - стек использует для доступа к элементам принцип FIFO, добавление элемента производится в конец стека, а извлечение – из начала стека
 - очередь использует для доступа к элементам принцип LIFO, добавление и извлечение элемента возможно только в вершине очереди.
 - стек использует для доступа к элементам принцип LIFO, добавление и извлечение элемента возможно только в вершине стека
8. Сколько элементов может содержать стек?
- 10
 - 100
 - 1000
 - 10000
 - все варианты верны
 - все варианты не верны
9. Пусть линейный связанный список описан следующим образом:
- ```
struct sList
{
 int x;
 double y;
 bool z;
 sList next;
};
```
- укажите хранимые данные, укажите где указатель на следующий элемент.
  - объявите переменную F, указывающую на начало такого списка
  - создайте новый элемент такого списка, выделите под него память и присвойте полям с данными какие-либо значения. Поместите его в начало списка.
  - создайте еще один элемент этого списка и сделайте его первым в списке. Присвойте произвольные значения полям данных этого элемента.
  - Что означает выражение  $F \rightarrow next$  ?
  - Чему равно значение  $F \rightarrow next \rightarrow next$  ?
  - Чему равно значение  $F \rightarrow y$  ?
  - Чему равно значение  $F \rightarrow next \rightarrow x$  ?
10. Запишите фрагмент программы, позволяющей вывести на экран все элементы списка, структура которого описана в задании 9, поле z которого равно true. Предполагается, что список может содержать произвольное количество элементов.

## Контрольная работа №4

### 1 вариант

1) К основному типу данных языка C++ не относят:

- a) char   б) int  
в) bool   г) string

2) Операция вида «+=» называется:

- a) простым присваиванием  
б) составным присваиванием  
в) логическим сложением



- г) логическим равенством
- 3) Для управления потоковым вводом-выводом используется заголовочный файл:  
а) `stdio` б) `conio`  
в) `iostream` г) `math`
- 4) Именованный участок памяти, в котором хранится значение определенного типа, называется:  
а) идентификатором  
б) константой  
в) файлом  
г) переменной
- 5) Квадрат числа  $x^2$  записывается в C++ как функция:  
а)  $x^2$  б) `sqr(x)`  
в) `sqrt(2)` г) `pow(x,2)`
- 6) Стандартная функция ввода в C++ имеет вид: а) `printf`  
б) `scanf` в) `inputf` г) `writef`
- 7) Символ строки форматов для перевода строки при выводе:  
а) `\b` б) `\n`  
в) `\t` г) `\r`
- 8) Виртуальный канал связи для передачи данных называется: а) функцией  
б) переменной в) потоком  
г) компиляцией
- 9) Условный оператор в C++ имеет следующую структуру: а) `if (условие) оператор_1; else оператор_2;`  
б) `if условие оператор_1 else оператор_2;`  
в) `if (условие) then оператор_1 else оператор_2;` г) `if условие do оператор_1 elseif оператор_2;`
- 10) Оператор цикла с постусловием:  
а) `for (начальные_присваивания; выражение; приращение) оператор;` б) `while (выражение) оператор;`  
в) `do оператор while (выражение);` г) `goto метка;`

## 2 вариант

- 1) Какой спецификатор данных не определен в C++:  
а) `double` б) `short`  
в) `long` г) `signed`
- 2) Операция вида `a=b=c` называется:  
а) составным присваиванием  
б) множественным присваиванием  
в) простым преобразованием  
г) логическим равенством
- 3) Для управления стандартным вводом-выводом используется заголовочный файл:

- a) stdio
- б) conio
- в) iostream
- г) math

4) Величина, которая не изменяет своего значения в процессе выполнения программы, называется:

- a) идентификатором
- б) константой
- в) файлом
- г) переменной

5) Модуль вещественного числа в C++ обозначается:

- a) abs    б) abc
- в) fabs   г) modul

6) Стандартная функция вывода в C++ имеет вид: а) printf

- б) scanf в) inputf г) writef

7) Символ строки форматов для табуляции при выводе:

- a) \b    б) \n
- в) \t    г) \r

8) Поточковые операции для ввода-вывода имеют вид:

- a) << >> б) { }
- в) ()   г) % &

9) Оператор множественного выбора в C++ имеет вид: а) select

- б) case
- в) include   г) switch

10) Оператор цикла с предусловием:

- a) for (начальные\_присваивания; выражение; приращение) оператор; б) while (выражение) оператор;
- в) do оператор while (выражение); г) goto метка

#### **а. Форма промежуточной аттестации**

Промежуточная аттестация по дисциплине– *дифференцированный зачет в 5 семестре, экзамен в 6 семестре.*

*Обучающиеся допускаются к сдаче дифференцированного зачета и экзамена при выполнении всех видов самостоятельной работы, практических и контрольных работ, предусмотренных рабочей программой дисциплины.*

#### **Перечень вопросов для проведения промежуточной аттестации:**

Задания к промежуточной аттестации

Теоретическая часть.

1. Что понимается под термином язык программирования? Что называется исходным кодом программы? Что такое интерпретатор и компилятор?

2. Что называется алгоритмом? Что такое алгоритмизация? Основные требования к составлению алгоритмов. Основные этапы разработки программы
3. Основные алгоритмические структуры (линейная последовательность действий, ветвление, циклы). Их назначение. Примеры применения.
4. Графическое представление алгоритмов. Блок-схемы. Назначение блок-схем. Достоинства и недостатки представления алгоритмов в виде блок-схем.
5. Алфавит языка C++. Переменные. Правила составления имен переменных. Основные типы данных языка C++, их назначение. Тип данных auto. Особенности использования.
6. Основные сведения об операциях в языке C++. Операция sizeof(). Приоритет выполнения операций. Выражения. Правила составления выражений.
7. Описание переменных. Директива #define. Модификатор const. Оператор присваивания. Тернарная операция «?».
8. Арифметические операции. Операции отношения. Логические операции. Битовые операции. Арифметические сдвиги. Бинарные и унарные операции.
9. Общие сведения об операторах языка C++. Простой оператор. Составной оператор. Область видимости переменных. Метки в C++. Оператор безусловного перехода goto.
10. Общая структура программы на языке C++. Назначение разделов программы. Функция main. Пример простой программы.
11. Условный оператор. Назначение условного оператора. Структура условного оператора. Разновидности условного оператора.
12. Оператор выбора switch. Назначение. Отличие от условного оператора. Примеры использования. «Подводные камни» и ошибки при использовании оператора выбора.
13. Циклы в C++. Назначение циклов. Типы циклов. Цикл с предусловием while. Операторы break и continue.
14. Циклы в C++. Назначение циклов. Типы циклов. Цикл с постусловием do ... while.
15. Циклы в C++. Назначение циклов. Типы циклов. Цикл с параметром for.
16. Структурированные типы данных. Массивы. Одномерные и многомерные массивы. Как в памяти располагается многомерный массив? Правила формирования индексов элементов массивов. Выход индекса за границы массивов, последствия этого и как избежать?
17. Структурированные типы данных. Цепочки (массивы) символов (строки). Представление строки в памяти. Операции со строками. Присваивание строк, сравнение строк, копирование строк и т.д. Управляющие символы.
18. Структурированные типы данных. Структуры. Описание структур. Назначение структур. Примеры структур.
19. Функции в C++. Назначение функций. Описание функций, реализация функций. Имя функции. Тип возвращаемого результата. Параметры функции. Возврат результата из функции.
20. Прототипы функций. Назначение. Параметры по умолчанию и правила их применения. Перегрузка функций. Назначение перегрузки. Правила перегрузки.
21. Передача аргументов в функцию. Передача по значению. Передача по ссылке. В чем разница в механизме и в описании аргументов, передаваемых по ссылке и по значению? Передача массивов в функции.

22. Динамическая память (куча). Указатели в C++. Назначение указателей. Определение указателей. Типы указателей. Операция разадресации. Адресная арифметика. Специальное значение NULL.
23. Операции с динамической памятью. Динамические переменные. Операции выделения и освобождения памяти (new и delete). Алгоритм работы с динамической памятью.
24. Динамические массивы. Одномерные динамические массивы, Двухмерные динамические массивы. Особенности и основные приемы работы.
25. Ввод-вывод в консоль в C++. Стандартные потоки ввода-вывода cin и cout. Правила использования. Основные операции ввода вывода.
26. Обработка событий клавиатуры в консольном приложении. Получение кода нажатой клавиши. Получение расширенного кода клавиши (например клавиши управления курсором, функциональные клавиши). Как определить была ли нажата какая-либо клавиша на клавиатуре, или нет?
27. Поточный ввод-вывод в файлы. Потоки ifstream, ofstream, fstream. Их назначение и отличие. Типовая процедура работы с файлом (чтение, запись), последовательность действий.
28. Основные методы для работы с файловыми потоками. Методы open(), close(), eof(), seekg(), tellg, get(), getline(), read(), write(). Режимы доступа к файлу.
29. Типы файлов. Текстовые файлы, бинарные файлы. Отличия в работе с этими файлами. Методы чтения и записи данных в бинарный файл.
30. Понятие исключения (exception) в программе. Причины возникновения исключений. Обработка исключений. Операторы try...catch. Оператор throw.
31. Алгоритмы обработки данных. Поиск данных в массиве. Поиск данных в несортированном массиве. Поиск данных в сортированном массиве.
32. Алгоритмы обработки данных. Сортировка данных. Назначение сортировки. Алгоритмы сортировки. Пузырьковая сортировка.
33. Алгоритмы обработки данных. Сортировка данных. Назначение сортировки. Алгоритмы сортировки. Выборочная сортировка.
34. Алгоритмы обработки данных. Сортировка данных. Назначение сортировки. Алгоритмы сортировки. Быстрая сортировка (quick sort).
35. Алгоритмы обработки данных. Парсинг строк. Для чего это нужно? Особенности парсинга. Пример парсинга строки.
36. Понятие рекурсии. Рекурсивные алгоритмы. Примеры рекурсивных алгоритмов. Достоинства и недостатки рекурсивных алгоритмов.
37. Динамические структуры данных. Списки. Линейные списки. Циклические списки. Однонаправленные и двунаправленные списки. Назначение и область применения списков. Достоинства и недостатки списков.
38. Динамические структуры данных. Стеки. Особенности организации стеков. Область применения стеков. Достоинства и недостатки стеков.
39. Динамические структуры данных. Очереди. Особенности организации очереди. Область применения очереди. Достоинства и недостатки очереди.
40. Понятие ООП. Класс. Объект. Инкапсуляция. Наследование. Полиморфизм. Приведите пример класса и иерархии классов.

41. Стандартная контейнерные классы. Строки (string), достоинства и недостатки по сравнению с массивами символов (char[]). Описание строк. Основные операции и методы работы со строками.
42. Стандартная контейнерные классы – array, list, vector. Создание и использование данных контейнеров. Понятие итератора. Назначение итераторов. Доступ к элементам, добавление, вставка, удаление и изменение элементов.
43. Ассоциативный контейнер map. Назначение. Что такое ключ, что такое значение. Добавление и вставка элементов в контейнер. Удаление элементов. Доступ к элементу по ключу.
44. Ассоциативный контейнер multimap. Назначение. Отличие от map. Что такое ключ, что такое значение. Добавление и вставка элементов в контейнер. Удаление элементов. Доступ к элементу по ключу.
45. Основы создания многопоточных приложений. Понятие процесса и потока. Что такое поток (thread) в плане многозадачности? Как создать поток в своей программе? Как запустить фрагмент кода в программе в отдельном потоке? Как дождаться завершения потока? Когда использования многопоточности оправдано в программе?

#### Практическая часть

11. Напишите консольное приложение на языке C++, которое запрашивает у пользователя его имя, а потом выводит на экран приветствие. Например пользователь за запрос программы ввел имя **Маша**. Программа должна напечатать:  
**Привет, Маша!**  
Образец:

|                                         |
|-----------------------------------------|
| Введите ваше имя: Маша<br>Привет, Маша! |
|-----------------------------------------|

12. Напишите на языке C++ функцию, возвращающую сумму чисел в диапазоне от 50 до 100.
13. Имеется две переменные символьного типа a и b, содержащие произвольные символы. Запишите фрагмент программы, позволяющий вывести эти символы на экран в алфавитном порядке.
14. Напишите на языке C++ функцию, возвращающую сумму чисел в диапазоне от -50 до 100 с использованием цикла с предусловием while.
15. Каким будет значение переменной X при выполнении следующего фрагмента программы:  

```
int A=20;
int B=10;
int X=A>B?A+B:A-B;
```
16. Имеется две переменные целого типа a и b, содержащие произвольные числа. Запишите фрагмент программы, позволяющий вывести эти числа на экран в порядке убывания.
17. Напишите консольное приложение на языке C++, которое запрашивает у пользователя его возраст, а потом выводит на экран приветствие. Например пользователь за запрос программы ввел возраст 17. Программа должна напечатать:  
**Ура! Тебе уже 17 лет!**

Образец:

```
Введите ваш возраст: 17
Ура! Тебе уже 17 лет!
```

18. Напишите на языке C++ функцию, возвращающую произведение чисел в заданном диапазоне от a до b. Значения параметров a и b должны передаваться в функцию.
19. Каким будет значение переменной X при выполнении следующего фрагмента программы:
- ```
int A=20;
int B=10;
```
20. `int X=A>(B+10)?A*B:A--;`

21. Напишите на языке C++ функцию, возвращающую сумму чисел в заданном диапазоне от a до b. Значения параметров a и b должны передаваться в функцию.
22. Напишите консольное приложение на языке C++, которое запрашивает у пользователя два числа, а потом выводит на экран их сумму. Например пользователь за запрос программы ввел числа 5 и 7. Программа должна напечатать:

5+7=12

Образец:

```
Введите два числа a и b: 5 7
5+7=12
```

23. Напишите консольное приложение на языке C++, которое запрашивает у пользователя два числа, а потом выводит на экран их произведение. Например пользователь за запрос программы ввел числа 5 и 7. Программа должна напечатать:
- 5*7=35**

Образец:

```
Введите два числа a и b: 5 7
5*7=35
```

24. Опишите указатель на переменную типа int. Разместите в динамической памяти такую переменную и присвойте ей произвольное значение. Освободите память, занимаемую данной переменной.
25. Вам необходимо разместить в памяти динамический одномерный массив из n элементов типа int, заполнить его возрастающими значениями от 0 до n-1. Напишите фрагмент программы, которая выполняет данные действия. Значение n необходимо запросить от пользователя.
26. Опишите структуру на c++, позволяющую хранить следующую информацию о печатающем устройстве: модель устройства (до 20 символов), год выпуска (целое число), серийный номер (до 10 символов), скорость печати листов/мин (целое число), наличие встроенного wifi модуля (логическое значение).
27. Напишите фрагмент программы, которая открывает текстовый файл, считывает из него два целых числа и выводит на экран их произведение.
28. Пусть заголовок растрового файла располагается в самом начале файла и описан следующей структурой:
- ```
struct GF {
 int width; // ширина изображения
 int height; // высота изображения
 char colors; // кол-во цветов
 bool compression; // true – изображение сжато, false – не сжато.
};
```

Приведите фрагмент программы, считывающей из файла эту информацию и выводящей на экран характеристики этого изображения.

29. Напишите фрагмент программы, выводящей на экран количество символов «А», находящихся в текстовом файле.
30. Опишите указатель на переменную типа double. Разместите в динамической памяти такую переменную и присвойте ей значение, хранящейся в другой переменной. Освободите память, занимаемую данной переменной.
31. Вам необходимо разместить в памяти динамический одномерный массив из n элементов типа short, заполнить убывающими значениями от n до 1. Напишите фрагмент программы, которая выполняет данные действия. Значение n необходимо запросить от пользователя.
32. Опишите структуру на c++, позволяющую хранить следующую информацию о студенте: ФИО (до 40 символов), год рождения (целое число), номер зачетной книжки (до 10 символов), Пол (символьное значение М или Ж), статус студента (отчислен или не отчислен) (логическое значение).
33. Напишите фрагмент программы, которая открывает текстовый файл, записывает в него два целых числа и выводит на экран их разность.
34. Пусть заголовок растрового файла располагается в самом начале файла и описан следующей структурой:

```
struct F1 {
 int w; // ширина изображения
 int h; // высота изображения
 char BitsPerPixel; // кол-во бит, необходимых для представления цвета
 точки
 bool compression; // true – изображение сжато, false – не сжато.
};
```
- Приведите фрагмент программы, считывающей из файла эту информацию и выводящей на число байт, необходимых для хранения растровых данных такого изображения. BitsPerPixel может принимать значения 1, 4, 8, 16, 24, 32
35. Опишите указатель на символьную переменную. Разместите в динамической памяти такую переменную и присвойте ей произвольное значение. Опишите другой указатель на символьную переменную и сделайте так, чтобы оба эти указателя ссылались на одну и ту же выделенную ячейку памяти.
36. Пусть имеется следующее описание: double \*F, \*P; Выделите память под динамические переменные, инициализируйте их начальными значениями и напечатайте на экране сумму этих значений.
37. Вам необходимо разместить в памяти динамический одномерный массив из n элементов типа char, заполнить убывающими значениями от n до 1 (n<=255). Напишите фрагмент программы, которая выполняет данные действия. Значение n необходимо запросить от пользователя.
38. Опишите структуру на c++, позволяющую хранить следующую информацию об автомобиле: модель (до 20 символов), год выпуска (целое число), гос. Пусть имеется следующее описание: char \*i, \*j; Выделите память под динамические переменные, инициализируйте их начальными значениями и напечатайте на экране эти значения.
39. Вам необходимо разместить в памяти динамический одномерный массив из n элементов типа float, заполнить убывающими значениями от с шагом 0,1. Напишите фрагмент программы, которая выполняет данные действия. Значение n необходимо запросить от пользователя.

40. Напишите фрагмент программы, которая открывает текстовый файл, и записывает в него массив вещественных чисел. Размер массива определите самостоятельно.
41. Напишите фрагмент программы, выводящей на экран количество слов, находящихся в текстовом файле. Предполагается, что знаки препинания отсутствуют.
42. Опишите структуру на с++, позволяющую хранить следующую информацию о городе: Название (до 20 символов), год основания (целое число), количество жителей (целое число), является ли город столицей субъекта России (логическое значение).
43. Напишите фрагмент программы, которая открывает текстовый файл, и считывает из него в массив 10 вещественных чисел и выводит их на экран.
44. Напишите фрагмент программы, выводящей на экран слова, находящиеся в текстовом файле. Предполагается, что знаки препинания отсутствуют.
45. Напишите фрагмент программы, позволяющей найти заданное число в целочисленном массиве. Искомое число запросите у пользователя с клавиатуры.
46. Создайте контейнер vector из элементов типа string. Организуйте добавление текстовых строк, введенным пользователем с клавиатуры в данный контейнер.
47. Напишите программу, печатающую на экране все четные числа из целочисленного массива.
48. Напишите программу, печатающую на экране все нечетные числа из целочисленного массива.
49. Напишите программу, печатающую на экране количество четных чисел из целочисленного массива.
50. Напишите программу, печатающую на экране количество четных чисел из целочисленного массива.
51. Дан двумерный целочисленный массив. Инициализируйте его начальными значениями. Необходимо найти сумму всех его элементов.
52. Дан двумерный целочисленный массив. Инициализируйте его начальными значениями, как положительными, так и отрицательными. Выведите его на экран таким образом, чтобы неотрицательные значения выводились без изменения, а отрицательные – нулевым значением «0».
53. Дан одномерный целочисленный массив. Замените все положительные четные значения массива соответствующими им отрицательными значениями, а отрицательные нечетные – соответствующими положительными значениями.
54. Найдите максимальное число в одномерном целочисленном массиве и выведите его на экран.

#### **в. Система оценивания комплекта ФОС текущего контроля и промежуточной аттестации**

При оценивании контрольной, практической и самостоятельной работы обучающегося учитывается следующее:

*- качество выполнения практической части работы;*  
*- качество устных ответов на контрольные вопросы при защите самостоятельной или практической работы.*

При оценивании ответа на вопросы дифференцированного зачета и экзамена учитывается следующее:

*- качество устных ответов на вопросы.*

Каждый вид работы оценивается по пятибальной шкале.

«5» (отлично) – за глубокое и полное овладение содержанием учебного материала, в котором обучающийся свободно и уверенно ориентируется; за умение



практически применять теоретические знания, высказывать и обосновывать свои суждения. Оценка «5» (отлично) предполагает грамотное и логичное изложение ответа.

«4» (хорошо) – если обучающийся полно освоил учебный материал, владеет научно-понятийным аппаратом, ориентируется в изученном материале, осознанно применяет теоретические знания на практике, грамотно излагает ответ, но содержание и форма ответа имеют отдельные неточности.

«3» (удовлетворительно) – если обучающийся обнаруживает знание и понимание основных положений учебного материала, но излагает его неполно, непоследовательно, допускает неточности, в применении теоретических знаний при ответе на практико-ориентированные вопросы; не умеет доказательно обосновать собственные суждения.

«2» (неудовлетворительно) – если обучающийся имеет разрозненные, бессистемные знания, допускает ошибки в определении базовых понятий, искажает их смысл; не может практически применять теоретические знания.

### **Критерии оценивания устного ответа:**

- оценка «отлично» ставится в случае, если студент демонстрирует прекрасное знание материала, умение оперировать основными понятиями, определениями и может уверенно, последовательно, грамотно и логически стройно, исчерпывающе изложить в своем ответе материал, касающийся затронутой темы, не затрудняясь с ответом при видоизменении задания, умеет самостоятельно обобщать материал;
- оценка «хорошо» ставится за хорошее знание студентом материала по теме, умение ясно и чётко осветить рассматриваемый материал, однако его ответ содержит некоторые незначительные неточности, студент во время изложения материала не вполне уверенно рассказывает о некоторых деталях вопроса, и поэтому его ответ остается недостаточно четким и исчерпывающим;
- оценка «удовлетворительно» выставляется в случае, если студент в целом знает рассматриваемую тему, в основном верно отвечает на поставленные вопросы, однако его ответ содержит существенные ошибки, неточности, а сам студент демонстрирует заметные пробелы в знаниях по курсу;
- оценка «неудовлетворительно» выставляется в случае, если студент не в состоянии более или менее чётко и внятно изложить материал, его ответ содержит настолько грубые ошибки, существенные неточности, что тема рассматриваемого вопроса остается на деле нераскрытой; кроме того,

студент демонстрирует очень существенные пробелы в знании или полное незнание рассматриваемой темы и совершенное неумение пользоваться её методами.

### **Критерии оценивания (конспект урока, контрольная, практическая)**

1. Оценка **«отлично»** выставляется при условии, что студент полностью выполнил задание и проявил отличные знания учебного материала. При этом работа оформлена в соответствии с требованиями, к ней можно предъявить минимум замечаний.
2. **«Хорошо»** ставится тогда, когда студент выполнил все задания, показал хорошие знания по пройденному материалу, но есть недочеты в оформлении работы и общие небольшие замечания, не влияющие на ее качество.
3. Оценку **«удовлетворительно»** студент получает за полностью выполненное задание при наличии в ней существенных неточностей и недочетов, не умения студента верно применить полученные знания, в оформлении работы есть нарушения, не аргументированные ответы, неактуальные или ненадежные источники информации.
4. **«Неудовлетворительно»** студент получает в том случае, когда он не полностью выполнил задание проявил недостаточный уровень знаний, не смог объяснить полученные результаты. Такая работа не отвечает требованиям, содержит противоречивые сведения.