

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

«УТВЕРЖДАЮ»

Зам. директора по УМР



Е.Н. Бралгина

«23» марта 2023 г.

## РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

### Б1.О.Д12 Алгоритмизация и программирование



Направление подготовки

**09.03.03 «Прикладная информатика»**

Квалификация выпускника

**БАКАЛАВР**

**Форма обучения - очная/заочная**


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.

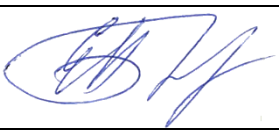
## Разработчик(и) рабочей программы дисциплины(модуля)

<b>ФИО</b>	<b>Ученая степень, звание, должность</b>	<b>Контактная информация (служебные E-mail и телефон)</b>
Мамрыкин О.В.	К.т.н., доцент	omamrykin@mail.ru

### *Экспертиза рабочей программы*

<b>Первый уровень</b> (оценка качества содержания программы, соответствие целям и задачам ООП ВО)	
<b>Руководитель ООП ВО</b>	<b>Подпись руководителя ООП ВО</b>
Мамрыкин О.В., к.т.н., доцент	
<i>Выписка из решения</i>	

<b>Второй уровень</b> (оценка качества содержания программы и применяемых педагогических технологий)		
<b>Наименование кафедры</b>	<b>№ протокола, дата</b>	<b>Подпись зав. кафедрой</b>
ИИТ	№7 от 14.03.23	
<i>Выписка из решения</i>		

<b>Третий уровень</b> (соответствие целям подготовки и учебному плану образовательной программы)		
<b>Методическая комиссия института, в структуре ООП которого будет реализовываться данная программа</b>	<b>№ протокола, дата</b>	<b>Подпись председателя МК</b>
	№3 от 21.03.23	
<i>Выписка из решения</i>		

## СОДЕРЖАНИЕ

1. Цель и задачи освоения дисциплины (модуля).....	4
2. Место дисциплины (модуля) в структуре основной образовательной программы.....	5
3. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения основной образовательной программы.....	5
4. Объем дисциплины (модуля) в зачетных единицах с указанием количества часов, выделенных на контактную работу обучающихся с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающихся .....	7
5. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества часов и видов учебных занятий.....	8
6. Перечень учебно-методического обеспечения для самостоятельной работы студентов по дисциплине (модулю).....	20
7. Фонд оценочных средств для проведения промежуточной аттестации по дисциплине (модулю) .....	22
8. Учебно-методическое и информационное обеспечение дисциплины (модуля).....	44
9. Методические указания для обучающихся по освоению дисциплины (модуля).....	46
10. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю) .....	47
11. Особенности организации образовательного процесса по дисциплине (модулю) для инвалидов и лиц с ограниченными возможностями здоровья.....	48

Рабочая программа дисциплины составлена в соответствии с требованиями ФГОС ВО 3++ по направлению подготовки 09.03.03 Прикладная информатика утвержденного Приказом Минобрнауки РФ от 19.09.2017г., № 922

### **1. Цель и задачи освоения дисциплины (модуля)**

Целью освоения дисциплины является формирование систематизированных знаний в области разработки алгоритмов для электронно-вычислительных машин, реализации этих алгоритмов на языке программирования высокого уровня, формирование у будущих специалистов практических навыков по программному решению вычислительных задач с помощью языков C++, C#, развитие умения работы в среде Microsoft Visual Studio, приобретение опыта отладки и тестирования программ.

Задачи освоения дисциплины (модуля):

- обучение основам разработки алгоритмов;
- обучение основам программирования на языке высокого уровня
- изучение языков программирования C++ и C#;
- изучение методов работы в современных средах разработки программного обеспечения;
- обучения способам разработки программ, их отладки и верификации;
- изучение основ работы в команде разработчиков при коллективной разработке программного обеспечения;

научить студентов:

- ориентироваться в современных языках программирования, читать блок-схемы алгоритмов, исходные тексты программ на языках C++, C#;

- объектно-ориентированному подходу при разработке программ и их модулей;
- выполнять постановку и формализацию задач разработки алгоритмов на бумаге и на языке высокого уровня;
- применять опыт программирования при решении прикладных задач;

студенты должны иметь представление:

- о способах разработки и записи алгоритмов;
- о различных языках программирования высокого уровня;
- о современных подходах к программированию алгоритмов, правильному и обоснованному выбору языка программирования для решения конкретной программной задачи;

## **2. Место дисциплины (модуля) в структуре основной образовательной программы**

Дисциплина входит в базовую часть образовательной программы. Дисциплина требует базовых школьных знаний информатики.

Успешное освоение дисциплины продолжит изучение Информационная безопасность, Программирование а .NET, Проектирование информационных систем и др.

## **3. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения основной образовательной программы**

Планируемые результаты обучения по дисциплине (модулю) – это знания, умения, навыки и (или) опыт деятельности. Планируемые результаты освоения образовательной программы – это формируемые дисциплиной (модулем) компетенции.

Освоение дисциплины (модуля) направлено на формирование элементов следующих компетенций в соответствии с ФГОС ВО и ООП ВО по данному направлению подготовки/специальности:

Результаты освоения ООП ВО (компетенции)	Индикатор достижения компетенции	Результаты обучения по дисциплине (модулю)	
ОПК-7 Способен разрабатывать алгоритмы и программы, пригодные для практического применения	ОПК-7.1. Осваивает современные языки программирования и программные среды для разработки программ, пригодных для практического применения	Способен: адаптировать прикладное программное обеспечение для решения поставленных задач	Уровень 1,2,3
	ОПК-7.2. Применяет языки программирования, современные программные среды для разработки и сопровождения программ, пригодных для практического применения	Знать: методы структурного программирования Уметь: разрабатывать и отлаживать эффективные алгоритмы с использованием современных технологий программирования Владеть: навыками программирования в современных средах	Уровень 1,2,3
	ОПК-7.3. Использует навыки алгоритмизации, программирования, отладки и тестирования информационных систем	Знать: методы тестирования программ Уметь: разрабатывать тесты Владеть: навыками тестирования компонентов программного обеспечения	Уровень 1,2,3
ПК-2 Способен разрабатывать и адаптировать прикладное программное обеспечение	ПК-2.1 Использует основы организационного и технологического обеспечения процесса разработки прикладного программного обеспечения, принципы оптимизации и рефакторинга	Способен разрабатывать алгоритмы, программы и их модули, проводить их отладку, оптимизацию и рефакторинг	Уровень 1,2,3
	ПК-2.2 Формализует и проводит алгоритмизацию прикладных задач, разрабатывает структуру программы и решения по интеграции ее модулей	Способен формализовать и проводить алгоритмизацию прикладных задач, разрабатывать структуру программы и решения по интеграции ее модулей	Уровень 1,2,3

		лей	
	ПК-2.3 Выполняет написание, отладку и оптимизацию программного кода, осуществляет интеграцию программных модулей и компонентов	Способен писать, отлаживать и оптимизировать код программы, проводить интеграцию программных модулей и их компонентов	Уровень 1,2,3

\*Уровень 1 (**повышенный**) предполагает готовность решать практические задачи повышенной сложности, нетиповые задачи, принимать профессиональные и управленческие решения в условиях неполной определенности, при недостаточном документальном, нормативном и методическом обеспечении (соответствует оценке «отлично» при оценивании освоенности компетенции.

\*\*Уровень 2 (**базовый**) позволяет решать типовые задачи, принимать профессиональные и управленческие решения по известным алгоритмам, правилам и методикам (соответствует оценке «хорошо» при оценивании освоенности компетенции.

\*\*\*Уровень 3 (**пороговый**) дает общее представление о виде деятельности, основных закономерностях функционирования объектов профессиональной деятельности, методов и алгоритмов решения практических задач (соответствует оценке «удовлетворительно» при оценивании освоенности компетенции.

#### 4. Объем дисциплины (модуля) в зачетных единицах с указанием количества часов, выделенных на контактную работу обучающихся с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающихся

Объем дисциплины	Всего часов	
	Очная форма обучения	Заочная (очно-заочная) форма обучения*
Общая трудоемкость, з.е./часов	14 /504	14/504
Контактная работа (всего), часов		
Аудиторная:		
Лекции	138	32
Практические занятия		
Лабораторные занятия	276	58
Групповые и индивидуальные консультации		
Контрольная работа	+	+
Зачет/экзамен	Зачет 1,2,3 сем. 4 сем –экз.	Зачет 1,2,3 сем. 4 сем –экз.
Внеаудиторная:		
Индивидуальные консультации		
иные формы		
В ЭИОС:		
Лекции		
Практические занятия		
Групповые и индивидуальные консультации		
Самостоятельная работа (всего), з.е./часов	61	401
Контроль самостоятельной работы (КСР)	2	4
Подготовка и написание курсовой работы		

**5. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества часов и видов учебных занятий**

Очная форма

№ п/п	Разделы, темы дисциплины, аннотация темы	Неделя семестра	Виды учебных занятий, включая самостоятельную работу обучающихся и трудоемкость (в часах)				С Р С	Формы текущего контроля успеваемости	Формируемые компетенции /индикаторы достижения компетенций
			Контактная работа с преподавателем						
			Лек	Пра кт	Лаб.	КСР			
	<b>Раздел 1 Алгоритмы и структурное программирование на языке C++</b>				1		Тесты, контрольные работы	ОПК-7, ПК-2	
	Тема 1. Что понимается под термином язык программирования? Что называется исходным кодом программы? Что такое интерпретатор и компилятор? Что называется алгоритмом? Что такое алгоритмизация? Основные требования к составлению алгоритмов. Основные этапы разработки программы Основные алгоритмические структуры (линейная последовательность действий, ветвление, циклы). Их назначение. Примеры применения. Графическое представление алгоритмов. Блок-схемы. Назначение блок-схем. Достоинства и недостатки представления алгоритмов в виде блок-схем.		5		10		2	Проверка выполненных заданий	ОПК-7, ПК-2
	Тема 2. Алфавит языка C++. Переменные. Правила составления имен переменных. Основные типы данных языка C++, их назначение. Тип данных auto. Особенности использования. Основные сведения об операциях в языке C++. Операция sizeof(). Приоритет выполнения операций. Выражения. Правила составления выражений. Описание переменных. Директива #define. Модификатор const. Оператор присваивания. Тернарная операция «?». Арифметические операции. Операции отношения. Логические операции. Битовые операции. Арифметические сдвиги. Бинарные и унарные операции. Лабораторные работы. Написание простой консольной программы на языке c++. Знакомство со структурой программы. Компиляция программы. Русификация вывода в консоль.		5		10			Проверка выполненных заданий	ОПК-7, ПК-2
	Тема 3. Общие сведения об операторах языка C++. Простой оператор. Составной оператор. Область видимости переменных. Метки в		5		10			Проверка выполненных заданий	ОПК-7, ПК-2



	<p>C++. Оператор безусловного перехода goto. 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>						ний	
	<p>Тема 4. Циклы в C++. Назначение циклов. Типы циклов. Цикл с предусловием while. Операторы break и continue.</p> <p>Циклы в C++. Назначение циклов. Типы циклов. Цикл с постусловием do ... while.</p> <p>Циклы в C++. Назначение циклов. Типы циклов. Цикл с параметром for.</p> <p>Лабораторные работы. Вычисление суммы ряда. Поиск минимума/максимума. Вычисление факториала, арифметической и геометрической прогрессии.</p>	5		10		2	Проверка выполненных заданий	ОПК-7, ПК-2
	<p>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?</p> <p>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.</p> <p>Структурированные типы данных. Структуры. Описание структур. Назначение структур. Примеры структур.</p> <p>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.</p>	5		10		2	Проверка выполненных заданий	ОПК-7, ПК-2
	<p>Тема 6. Функции в C++. Назначение функций. Описание функций, реализация функций. Имя функции. Тип возвращаемого результата. Параметры функции. Возврат результата из функции.</p> <p>Прототипы функций. Назначение. Параметры по умолчанию и правила их применения. 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>	6		10		2	Проверка выполненных заданий	ОПК-7, ПК-2

<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>Тема 7. Динамическая память (куча). Указатели в C++. Назначение указателей. Определение указателей. Типы указателей. Операция разадресации. Адресная арифметика. Специальное значение NULL.</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>	6		10		4	Проверка выполненных заданий	ОПК-7, ПК-2
<p>Тема 8. Ввод-вывод в консоль в C++. 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 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>	5		10		2	Проверка выполненных заданий	ОПК-7, ПК-2
<p>Тема 9. Понятие исключения (exception) в программе. Причины возникновения исключений. Обработка исключений. Операторы try...catch. Оператор throw. Стандартные классы исключений.</p> <p>Лабораторные работы. Создание функции</p>	5		10		2	КР	ОПК-7, ПК-2

вычисления корней квадратного уравнения, выбрасывающей исключение при невозможности их вычисления и обработка данного исключения. Создание функции поиска заданного элемента в массиве, выбрасывающей исключение <code>out_of_range</code> при задании неверного индекса элемента, с которого начать поиск. Создание библиотеки пользовательских функций с выбрасыванием и обработкой исключений.							
Тема 10. Алгоритмы обработки данных. Поиск данных в массиве. Поиск данных в несортированном массиве. Поиск данных в сортированном массиве. 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.	6		12		4		ОПК-7, ПК-2
Тема 11. Динамические структуры данных. Списки. Линейные списки. Циклические списки. Однонаправленные и двунаправленные списки. Назначение и область применения списков. Достоинства и недостатки списков. Динамические структуры данных. Стеки. Особенности организации стеков. Область применения стеков. Достоинства и недостатки стеков. Динамические структуры данных. Очереди. Особенности организации очереди. Область применения очереди. Достоинства и недостатки очереди.	5		10		2	Проверка выполненных заданий	ОПК-7, ПК-2
Тема 12. Функции обратного вызова и их применение.	2		10		2		ОПК-7, ПК-2
Тема 13. Анонимные функции, лямбда-функции.	2		10		2		ОПК-7, ПК-2
<b>Раздел 2. Объектно-ориентированное программирование</b>				1		Тесты, контрольные работы	ОПК-7, ПК-2
Тема 14. Объектно-ориентированное программирование (ООП). История развития ООП. Базовые понятия ООП: класс, его свойства и методы. Различие между классом и объектом. Основные принципы ООП: инкапсуляция, наследование, полиморфизм. Примеры созда-	6		12		4	Проверка выполненных заданий	ОПК-7, ПК-2

<p>ния классов. Стандартные классы, используемые в C++.</p> <p>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.</p>							
<p>Тема 15. Применение объектно-ориентированного программирования при решении прикладных задач. Особенности использования спецификаторов управления областью видимости (public, protected, private).</p>	5		10		2	Тест	ОПК-7, ПК-2
<p>Тема 16 Создание простой консольной игры с применением объектно-ориентированного подхода.</p>	6		10		2	Проверка выполненных заданий	ОПК-7, ПК-2
<p>Тема 17. Создание собственной библиотеки классов. Создание динамических библиотек (DLL).</p>	5		10		2	Проверка выполненных заданий	ОПК-7, ПК-2
<p>Тема 18. Объектно-ориентированный подход к разработке графического интерфейса. Стандартные компоненты приложения Windows Forms. Разработка графических оконных приложений.</p>	6		10		2	Проверка выполненных заданий	ОПК-7, ПК-2
<p>Тема 19. Основы платформы .NET. Основная идеология, преимущества, достоинства и недостатки. JIT-компилятор. Особенности разработки программ, выполняемых в среде CLR. Разница в управляемом и неуправляемом коде на языке C++.</p>	5		10		3	Проверка выполненных заданий	ОПК-7, ПК-2
<p>Тема 20. Основы языка C#. Основные отличия от C++. Типы данных, особенности реализации.</p>	6		10		2	Проверка выполненных заданий	ОПК-7, ПК-2
<p>Тема 21. Переменные, операторы языка, выражения, операции. Общая структуры программы.</p>	6		10		2	Проверка выполненных заданий	ОПК-7, ПК-2
<p>Тема 22. Классы. Основные понятия, отличия от C++. Иерархия классов. Понятия интерфейсов.</p>	6		10		2	Проверка выполненных заданий	ОПК-7, ПК-2
<p>Тема 23. Делегаты, события и потоки выполнения,</p>	6		10		2	Проверка выполненных заданий	ОПК-7, ПК-2
<p>Тема 24. Работа с файлами и потоками.</p>	5		10		2	Проверка выполненных заданий	ОПК-7, ПК-2
<p>Тема 25. Многозадачность и многопоточность. Асинхронность и асинхронные методы.</p>	4		12		2	Проверка выполненных заданий	ОПК-7, ПК-2
<p>Тема 26. Разработка оконных приложений на C#.</p>	6		10		2	Проверка выполненных заданий	ОПК-7, ПК-2

Тема 27. Тестирование программ. Создание модульных тестов	6	10	2	ОПК-7, ПК-2
<b>ИТОГО</b>	<b>138</b>	<b>276</b>	<b>2</b> <b>6</b> <b>1</b>	

Форма промежуточной аттестации – зачет в 1,2,3 семестры, экзамен – 4 семестр.

### Заочная форма

№ п/п	Разделы, темы дисциплины, аннотация темы	Виды учебных занятий, включая самостоятельную работу обучающихся и трудоемкость (в часах)				Формы текущего контроля успеваемости	Формируемые компетенции /индикаторы достижения компетенций	
		Контактная работа с преподавателем						СРС
		Лек.	Практ	Лаб.	КСР			
	<b>Раздел 1</b> <b>Алгоритмы и структурное программирование на языке C++</b>				1		Тесты, контроль контрольные работы	ОПК-7, ПК-2
	Тема 1. Что понимается под термином язык программирования? Что называется исходным кодом программы? Что такое интерпретатор и компилятор? Что называется алгоритмом? Что такое алгоритмизация? Основные требования к составлению алгоритмов. Основные этапы разработки программы Основные алгоритмические структуры (линейная последовательность действий, ветвление, циклы). Их назначение. Примеры применения. Графическое представление алгоритмов. Блок-схемы. Назначение блок-схем. Достоинства и недостатки представления алгоритмов в виде блок-схем.	2				15	Проверка выполненных заданий	ОПК-7, ПК-2
	Тема 2. Алфавит языка C++. Переменные. Правила составления имен переменных. Основные типы данных языка C++, их назначение. Тип данных auto. Особенности использования. Основные сведения об операциях в языке C++. Операция sizeof(). Приоритет выполнения операций. Выражения. Правила составления выражений. Описание переменных. Директива #define. Модификатор const. Оператор присваивания. Тернарная операция «?». Арифметические операции. Операции отношения. Логические операции. Битовые операции. Арифметические сдвиги. Бинарные и унарные операции. Лабораторные работы. Написание простой консольной программы на языке c++. Знакомство со структурой программы. Компиляция программы. Русификация вывода в консоль.	2				16	Проверка выполненных заданий	ОПК-7, ПК-2
	Тема 3. Общие сведения об операторах языка	2				16	Провер-	ОПК-7,

<p>C++. Простой оператор. Составной оператор. Область видимости переменных. Метки в C++. Оператор безусловного перехода goto. 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>						ка вы-полнен-ных за-даний	ПК-2
<p>Тема 4. Циклы в C++. Назначение циклов. Типы циклов. Цикл с предусловием while. Операторы break и continue.</p> <p>Циклы в C++. Назначение циклов. Типы циклов. Цикл с постусловием do ... while.</p> <p>Циклы в C++. Назначение циклов. Типы циклов. Цикл с параметром for.</p> <p>Лабораторные работы. Вычисление суммы ряда. Поиск минимума/максимума. Вычисление факториала, арифметической и геометрической прогрессии.</p>	2				16	Провер-ка вы-полнен-ных за-даний	ОПК-7, ПК-2
<p>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?</p> <p>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.</p> <p>Структурированные типы данных. Структуры. Описание структур. Назначение структур. Примеры структур.</p> <p>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.</p>	2		4		16	Провер-ка вы-полнен-ных за-даний	ОПК-7, ПК-2
<p>Тема 6. Функции в C++. Назначение функций. Описание функций, реализация функций. Имя функции. Тип возвращаемого результата. Параметры функции. Возврат результата из функции.</p> <p>Прототипы функций. Назначение. Параметры по умолчанию и правила их применения. 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,</p>	2		4		16	Провер-ка вы-полнен-ных за-даний	ОПК-7, ПК-2

<p>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>Тема 7. Динамическая память (куча). Указатели в C++. Назначение указателей. Определение указателей. Типы указателей. Операция разадресации. Адресная арифметика. Специальное значение NULL.</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>	2		4		16	Проверка выполненных заданий	ОПК-7, ПК-2
<p>Тема 8. Ввод-вывод в консоль в C++. 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 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>	2		4		16	Проверка выполненных заданий	ОПК-7, ПК-2
<p>Тема 9. Понятие исключения (exception) в программе. Причины возникновения исключений. Обработка исключений. Операторы try...catch. Оператор throw. Стандартные</p>			4		16	КР	ОПК-7, ПК-2

классы исключений. 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.							
Тема 10. Алгоритмы обработки данных. Поиск данных в массиве. Поиск данных в несортированном массиве. Поиск данных в сортированном массиве. 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.			4		16		ОПК-7, ПК-2
Тема 11. Динамические структуры данных. Списки. Линейные списки. Циклические списки. Однонаправленные и двунаправленные списки. Назначение и область применения списков. Достоинства и недостатки списков. Динамические структуры данных. Стеки. Особенности организации стеков. Область применения стеков. Достоинства и недостатки стеков. Динамические структуры данных. Очереди. Особенности организации очереди. Область применения очереди. Достоинства и недостатки очереди.			4		16	Проверка выполненных заданий	ОПК-7, ПК-2
Тема 12. Функции обратного вызова и их применение.					16		ОПК-7, ПК-2
Тема 13. Анонимные функции, лямбда-функции.					16		ОПК-7, ПК-2
<b>Раздел 2. Объектно-ориентированное программирование</b>				1		Тесты, контроль трольные работы	ОПК-7, ПК-2
Тема 14. Объектно-ориентированное программирование (ООП). История развития ООП. Базовые понятия ООП: класс, его свой-	1		12		12	Проверка выполнен-	ОПК-7, ПК-2



ства и методы. Различие между классом и объектом. Основные принципы ООП: инкапсуляция, наследование, полиморфизм. Примеры создания классов. Стандартные классы, используемые в C++. 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.							ных заданий	
Тема 15. Применение объектно-ориентированного программирования при решении прикладных задач. Особенности использования спецификаторов управления областью видимости (public, protected, private).		1				14	Тест	ОПК-7, ПК-2
Тема 16 Создание простой консольной игры с применением объектно-ориентированного подхода.		1		2		14	Проверка выполненных заданий	ОПК-7, ПК-2
Тема 17. Создание собственной библиотеки классов. Создание динамических библиотек (DLL).		1		1		14	Проверка выполненных заданий	ОПК-7, ПК-2
Тема 18. Объектно-ориентированный подход к разработке графического интерфейса. Стандартные компоненты приложения Windows Forms. Разработка графических оконных приложений.		1		2		14	Проверка выполненных заданий	ОПК-7, ПК-2
Тема 19. Основы платформы .NET. Основная идеология, преимущества, достоинства и недостатки. JIT-компилятор. Особенности разработки программ, выполняемых в среде CLR. Разница в управляемом и неуправляемом коде на языке C++.		1		1		14	Проверка выполненных заданий	ОПК-7, ПК-2
Тема 20. Основы языка C#. Основные отличия от C++. Типы данных, особенности реализации.		1		2		14	Проверка выполненных заданий	ОПК-7, ПК-2
Тема 21. Переменные, операторы языка, выражения, операции. Общая структуры программы.		1		2		14	Проверка выполненных заданий	ОПК-7, ПК-2
Тема 22. Классы. Основные понятия, отличия от C++. Иерархия классов. Понятия интерфейсов.		2		2		14	Проверка выполненных заданий	ОПК-7, ПК-2
Тема 23. Делегаты, события и потоки выполнения,		1		2		14	Проверка выполненных заданий	ОПК-7, ПК-2
Тема 24. Работа с файлами и потоками.		1		2		14	Проверка вы-	ОПК-7,

							полнен-ных за-даний	ПК-2	
	Тема 25. Многозадачность и многопоточность. Асинхронность и асинхронные методы.		2		2		14	Проверка выполненных заданий	ОПК-7, ПК-2
	Тема 26. Разработка оконных приложений на C#.		1		2		14	Проверка выполненных заданий	ОПК-7, ПК-2
	Тема 27. Тестирование программ. Создание модульных тестов		1				14		ОПК-7, ПК-2
	<b>ИТОГО</b>		<b>32</b>		<b>58</b>	<b>4</b>	<b>401</b>		

Тематика лабораторных работ:

- Написание простой консольной программы на языке c++. Знакомство со структурой программы. Компиляция программы. Русификация вывода в консоль.
- Реализация программы вычисления линейного уравнения. Реализация программы решения квадратного уравнения. Реализация вычисления кусочно-линейной функции. Реализация простого консольного калькулятора. Тестирование и пошаговая отладка программы. Отслеживание значений переменных на каждом шаге. Точки останова
- Вычисление суммы ряда. Поиск минимума/максимума. Вычисление факториала, арифметической и геометрической прогрессии.
- 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.
- 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.
- 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.
- 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.
- 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. 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 повторений). 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.
- Организация базы данных товаров в виде линейного однонаправленного списка. Реализация функций по добавлению, удалению, вставке, поиску элементов в списке.
- Организация стека в виде связного 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.
- 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.
- 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.
- 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 фоновое сохранения 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.
- Создание консольной компьютерной игры с использованием объектно-ориентированного подхода.
- Создание приложения Windows Form.

## 6. Перечень учебно-методического обеспечения для самостоятельной работы студентов по дисциплине (модулю)

### Структура СРС

Код индикатора формируемой компетенции*	Тема*	Вид	Форма	Объем учебной работы (часов)	Учебно-методические материалы
ОПК-7, ПК-2	Тема 1	Изучение материала, конспектирование материала	СРС без участия преподавателя	2	1,2,3
ОПК-7, ПК-2	Тема 2	Изучение материала, конспектирование материала	СРС без участия преподавателя	2	1,2,3
ОПК-7, ПК-2	Тема 3	Изучение материала, конспектирование материала	СРС без участия преподавателя	2	1,2,3
ОПК-7, ПК-2	Тема 4	Изучение материала, конспектирование материала	СРС без участия преподавателя	2	1,2,3
ОПК-7, ПК-2	Тема 5	Изучение материала, конспектирование материала	СРС без участия преподавателя	2	1,2,3
ОПК-7, ПК-2	Тема 6	Изучение материала, конспектирование материала	СРС без участия преподавателя	2	1,2,3
ОПК-7, ПК-2	Тема 7	Изучение материала, конспектирование материала	СРС без участия преподавателя	4	1,2,3
ОПК-7, ПК-2	Тема 8	Изучение материала, конспектирование материала	СРС без участия преподавателя	2	1,2,3
ОПК-7, ПК-2	Тема 9	Изучение материала, конспектирование материала	СРС без участия преподавателя	2	1,2,3
ОПК-7, ПК-2	Тема 10	Изучение материала, конспектирование материала	СРС без участия преподавателя	4	1,2,3
ОПК-7, ПК-2	Тема 11	Изучение материала, конспектирование материала	СРС без участия преподавателя	2	1,2,3
ОПК-7, ПК-2	Тема 12	Изучение материала, конспектирование материала	СРС без участия преподавателя	2	1,2,3
ОПК-7, ПК-2	Тема 13	Изучение материала, конспектирование материала	СРС без участия преподавателя	2	1,2,3
ОПК-7, ПК-2	Тема 14	Изучение материала, конспектирование материала	СРС без участия преподавателя	4	1,2,3
ОПК-7, ПК-2	Тема 15	Изучение материала, конспектирование материала	СРС без участия преподавателя	2	1,2,3
ОПК-7, ПК-2	Тема 16	Изучение материала, конспектирование материала	СРС без участия преподавателя	2	1,2,3
ОПК-7, ПК-2	Тема 17	Изучение материала, конспектирование материала	СРС без участия преподавателя	2	1,2,3
ОПК-7, ПК-2	Тема 18	Изучение матери-	СРС без уча-	2	1,2,3

		ала, конспектирование материала	ствия преподавателя		
ОПК-7, ПК-2	Тема 19	Изучение материала, конспектирование материала	СРС без участия преподавателя	3	<b>1,2,3</b>
ОПК-7, ПК-2	Тема 20	Изучение материала, конспектирование материала	СРС без участия преподавателя	2	<b>1,2,3</b>
ОПК-7, ПК-2	Тема 21	Изучение материала, конспектирование материала	СРС без участия преподавателя	2	<b>1,2,3</b>
ОПК-7, ПК-2	Тема 22	Изучение материала, конспектирование материала	СРС без участия преподавателя	2	<b>1,2,3</b>
ОПК-7, ПК-2	Тема 23	Изучение материала, конспектирование материала	СРС без участия преподавателя	2	<b>1,2,3</b>
ОПК-7, ПК-2	Тема 24	Изучение материала, конспектирование материала	СРС без участия преподавателя	2	<b>1,2,3</b>
ОПК-7, ПК-2	Тема 25	Изучение материала, конспектирование материала	СРС без участия преподавателя	2	<b>1,2,3</b>
ОПК-7, ПК-2	Тема 26	Изучение материала, конспектирование материала	СРС без участия преподавателя	2	<b>1,2,3</b>
ОПК-7, ПК-2	Тема 27	Изучение материала, конспектирование материала	СРС без участия преподавателя	2	<b>1,2,3</b>

Виды СРС (выбираем и прописываем конкретный вид СРС):

подготовка к контрольной работе;  
подготовка к коллоквиуму;  
подготовка реферата, доклада;  
подготовка к деловым играм;  
решение задач;  
выполнение расчетно-графических работ;  
выполнение заданий в ЭИОС;  
написание курсовой работы.

По одной теме может быть несколько видов СРС.

Формы СРС (выбираем и прописываем конкретные формы СРС):

СРС (без участия преподавателя);

КСР (контроль самостоятельной работы студента).

\* Несколько индикаторов достижения компетенций могут реализовываться одной (или несколькими) темой (темами) СРС.

### Содержание СРС:

- Подготовка к теоретическому опросу, проработка конспектов лекций, учебной и научно-исследовательской литературы. Поиск информации в Интернет – ресурсах по теме задания.
- Построить алгоритмы различной конструкции по индивидуальным заданиям. Проанализировать примеры алгоритмов (определение НОД 2-х чисел, раскладка шаров по цвету, выбор большего числа).
- Составление конспекта на тему «История развития термина алгоритма». Изучить приоритет выполнения действий в выражениях, подготовить конспект.
- Составление выражений. Составление таблиц истинности логических операций. Изменение приоритета операций. Составление алгоритмов простых программ (решение линейного уравнения, квадратного уравнения. Вычисление значений кусочно-линейных функций). Пошаговое выполнение программ. Построение сложных логических выражений в условных операторах
- 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.

- Подготовка к контрольным работам
- Обзор алгоритмов обработки данных в массивах. Стандартные функции по работе с массивами. Копирование массивов. Подготовка с проверочным работам
- знакомство со стандартными функциями библиотек с++. Написание реферата по обзору часто используемых функций.
- 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.
- 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++
- 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 быстрдействию. Стандартные классы исключений.
- Разобрать алгоритмы сортировки слиянием, сортировки пирамидой. Построение двоичного дерева поиска. Построение алгоритма поиска в бинарном дереве. Обход дерева. Написание рефератов по одному из алгоритмов.
- Разработка комплексной программы на основе связанных списков (по вариантам).
- Обзор возможностей объектно-ориентированного программирования в разных языках (C++, C#, Python, Visual Basic, Object Pascal, Java и т.п.)
- Обзор организации многопоточности в разных языках программирования (C++, C#, Python, Visual Basic, Object Pascal, Java и т.п.)
- Обзор средств разработки пользовательского интерфейса программ.

## **7. Фонд оценочных средств для проведения промежуточной аттестации по дисциплине (модулю)**

Оценка качества освоения дисциплины включает текущий контроль и промежуточную аттестацию обучающихся.

Текущий контроль освоения дисциплины (модуля) осуществляется в виде проверки выполнения практических занятий, тестов и контрольной работы.

- 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.

Промежуточная аттестация по итогам освоения дисциплины проводится в форме зачета.

## Оценочные средства по дисциплине

### 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:  
ВЕС = 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!
```



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

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

- Запишите на языке C++ выражение  $\frac{1}{\left(\frac{f}{k-1} - \frac{f^3}{5+4k}\right) \cdot k}$
- Какие из значений являются символьными константами, а какие строковыми? ( '5' "12" '7' "0.0 -0.0" -67.9 1E3 "45")
- 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.
- 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.

- 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

- 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!**

- 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;
```

- Приведите фрагмент блок-схемы алгоритма, реализующий в алгоритме линейную последовательность действий.
- Напишите на языке C++ функцию, возвращающую сумму чисел в заданном диапазоне от  $a$  до  $b$ . Значения параметров  $a$  и  $b$  должны передаваться в функцию.
- Каким будет значение переменной  $X$  при выполнении следующего фрагмента программы:

```
int A=20;
int B=10;
int X=A>B?A-B:A--;
```

- Запишите на языке C++ выражение  $(a + b) \cdot \frac{1}{a - b} + 3c$
- Какие из значений являются символьными константами, а какие строковыми? ( '5' "TRUE" '7' "0.0" '- ' "0.0" -67.9 1E3 "45")
- 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.
- Нужно вывести на экран следующее:

ВАШ СЧЕТ: 200

Значение счета (200) содержится в переменной  $X$ . Запишите оператор, позволяющий выполнить указанное действие.

- 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

- 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:

<p>Введите два числа <math>a</math> и <math>b</math>: 5 7 <b>5-7=-2</b></p>
---

- 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;
```
- Приведите блок-схему алгоритма для задачи, указанной в вопросе 1,
- Напишите на языке C++ функцию, возвращающую сумму чисел в заданном диапазоне от  $a$  до  $b$ . Значения параметров  $a$  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 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 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 a и b: 5 7 <b>5+7=12</b>
---

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 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++: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:

<p>Введите два числа a и b: 5 7</p> <p><b>5*7=35</b></p>
--

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 ( $n \leq 255$ ). 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 `c++`, 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  $s$  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 `c++`, 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»?
- a. стек является частным случаем связанного списка, а очередь – нет.
  - b. очередь является частным случаем связанного списка, а стек – нет.
  - c. очередь использует для доступа к элементам принцип FIFO, добавление элемента производится в конец очереди, а извлечение – из начала очереди
  - d. стек использует для доступа к элементам принцип FIFO, добавление элемента производится в конец стека, а извлечение – из начала стека
  - e. очередь использует для доступа к элементам принцип LIFO, добавление и извлечение элемента возможно только в вершине очереди.
  - f. стек использует для доступа к элементам принцип LIFO, добавление и извлечение элемента возможно только в вершине стека
8. Сколько элементов может содержать стек?
- a. 10
  - b. 100
  - c. 1000
  - d. 10000
  - e. все варианты верны
  - f. все варианты не верны
9. Пусть линейный связанный список описан следующим образом:

```

struct sList
{
    int x;
    double y;
    bool z;
    sList next;
};

```

- a. укажите хранимые данные, укажите где указатель на следующий элемент.
  - b. объявите переменную F, указывающую на начало такого списка
  - c. создайте новый элемент такого списка, выделите под него память и присвойте полям с данными какие-либо значения. Поместите его в начало списка.
  - d. создайте еще один элемент этого списка и сделайте его первым в списке. Присвойте произвольные значения полям данных этого элемента.
  - e. Что означает выражение  $F \rightarrow next$  ?
  - f. Чему равно значение  $F \rightarrow next \rightarrow next$  ?
  - g. Чему равно значение  $F \rightarrow y$  ?
  - h. 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.

### Примерная тематика вопросов для тестирования

#### Тест 1

- 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:
  - a) stdio    б) conio
  - в) iostream    г) math
  
- 4) Именованный участок памяти, в котором хранится значение определенного типа, называется:
  - a) 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;`

## Тест 2

- 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:  
 а) `stdio`  
 б) `conio`  
 в) `iostream`  
 г) `math`
- 4) Величина, которая не изменяет своего значения в процессе выполнения программы, называется:  
 а) идентификатором  
 б) константой  
 в) файлом  
 г) переменной

- 5) Модуль вещественного числа в C++ обозначается:  
 а) 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:  
 а) \b б) \n  
 в) \t г) \r
- 8) Поточковые операции для ввода-вывода имеют вид:  
 а) <<>> б) { }  
 в) () г) % &
- 9) Оператор множественного выбора в C++ имеет вид: а) select  
 б) case  
 в) include г) switch
- 10) Оператор цикла с предусловием:  
 а) for (начальные\_присваивания; выражение; приращение) оператор; б) while (выражение) оператор;  
 в) do оператор while (выражение); г) goto метка

### Тест 3

1. Укажите конструктор, который будет вызван компилятором, если объект А класса Point объявить таким образом — Point A:

Конструктор по умолчанию;  
 Конструктор копирования;  
 Конструктор присваивания;  
 Конструктор перемещения;  
 Конструктор с параметрами

2. Выберите, какие виды конструкторов является в C +:

Конструктор по умолчанию;  
 Конструктор копирования;  
 Конструктор присваивания;  
 Конструктор перемещения;  
 Конструктор с параметрами

3. Укажите, когда вызывается деструктор объекта

После вызова конструктора этого объекта;  
 При завершении программы;  
 Когда программа выходит из области действия объекта;  
 После вызова деструктора этого объекта  
 + При исполнении операции delete для динамического объекта

4. Определите, что такое производный класс:

Класс, построенный на основе другого класса  
 Класс, на основе которого построен другой класс

Класс, на основе которого построено ни одного класса

Класс, у которого кроме элементов являются методы

5. Укажите, какие спецификаторы доступа не существуют

public

protected

private

open

6. Определите, имеет доступ производный класс к закрытым элементам своего базового класса:

имеет

нет

Имеет при открытом наследовании

Имеет при отсутствии спецификатора наследования

7. Определите, что такое множественное наследование:

Наследование нескольких производных классов от одного базового

Наследования производного класса от нескольких базовых

Наследование производного класса с спецификатором `final`

Наследование производного класса от класса, который является производным от другого класса

8. Даны объявления класса. Определить количество полей и методов:

```
class Товар
```

```
{ Char * nazva; int price;
```

```
void Show ();
```

```
Товар ();
```

```
Товар (char *, int);
```

```
};
```

2 поля, 1 способ

2 поля, 2 метода

2 поля, 3 метода

3 поля, 2 метода

5 полей, метод нет

9. Даны два экземпляра класса. Каким может быть его объявление?

```
Pro P1, P2;
```

```
Class Pro
```

```
{ Int a, b; Pro (int, int); }
```

```
Class Pro
```

```
{ Int a, b; }
```

```
Class Pro
```

```
{ Int a, b; Pro (Pro &) Pro (int, int); }
```

```
Class Pro
```

```
{ Int a, b; Pro (Pro &) }
```

10. Даны объявления класса. Создать 2 экземпляра класса:

```
class Point
```

```
{ Int x, y;
```

```
public:
```

```
void Show ();
```

```
};  
Point t1; Point t2 (10,50)  
Point t1; Point t2;  
Point t1 (100,100) Point t2 (10,50)  
Point t1 (50,50) Point t2 ();
```

11. Дано описание класса. Определить правильную реализацию метода:

```
class person  
{  
char name [20];  
int age;  
void Init (char *, int);  
} X;  
Предоставить переменной x конкретных значений  
Void Init (char * a, int b) {strcpy (name, a) age = b};  
Void Init (char * a, int b) {name = a; age = b};  
Void person :: Init (char * a, int b) {strcpy (name, a) age = b};  
Void person :: Init (char * a, int b) {name = a; age = b};
```

Критерии оценивания:

- отлично – правильно сделано более 90% задания.
- хорошо – правильно сделано более 75% задания.
- удовлетворительно – правильно сделано не менее 50 % задания.
- неудовлетворительно – правильно сделано менее 50 % задания.

Примерный перечень вопросов к зачету:

Задания к промежуточной аттестации

Зачет 1 семестр.

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. Шаблоны функций. Преимущества шаблонов.
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. Поточковый ввод-вывод в файлы. Поток `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`.

#### Зачет 2 семестр.

1. Алгоритмы обработки данных. Поиск данных в массиве. Поиск данных в несортированном массиве. Поиск данных в отсортированном массиве.
2. Алгоритмы обработки данных. Сортировка данных. Назначение сортировки. Алгоритмы сортировки. Пузырьковая сортировка.
3. Алгоритмы обработки данных. Сортировка данных. Назначение сортировки. Алгоритмы сортировки. Выборочная сортировка.
4. Алгоритмы обработки данных. Сортировка данных. Назначение сортировки. Алгоритмы сортировки. Сортировка слиянием.
5. Алгоритмы обработки данных. Сортировка данных. Назначение сортировки. Алгоритмы сортировки. Быстрая сортировка (`quick sort`).
6. Алгоритмы поиска данных. Сквозной поиск. Бинарный поиск. Особенности реализации алгоритмов бинарного поиска.
7. Деревья. Двоичные деревья. Задачи на деревьях.
8. Иерархические структуры. Графы. Основные задачи на графах.
9. Понятие рекурсии. Рекурсивные алгоритмы. Примеры рекурсивных алгоритмов. Достоинства и недостатки рекурсивных алгоритмов.
10. Рекурсивные алгоритмы поиска, сортировки и обхода дерева.
11. Алгоритмы обработки данных. Парсинг строк. Для чего это нужно? Особенности парсинга. Пример парсинга строки.
12. Понятие рекурсии. Рекурсивные алгоритмы. Примеры рекурсивных алгоритмов. Достоинства и недостатки рекурсивных алгоритмов.
13. Динамические структуры данных. Списки. Линейные списки. Циклические списки. Однонаправленные и двунаправленные списки. Назначение и область применения списков. Достоинства и недостатки списков.
14. Динамические структуры данных. Стеки. Особенности организации стеков. Область применения стеков. Достоинства и недостатки стеков.
15. Динамические структуры данных. Очереди. Особенности организации очереди. Область применения очереди. Достоинства и недостатки очереди.

#### Зачет 3 семестр.



1. Понятие ООП. Абстракция данных. Абстрактные типы данных. Класс. Объект.
2. Основные принципы ООП. Инкапсуляция. Наследование. Полиморфизм.
3. Принципы описания класса. Члены класса. Поля. Методы. Использование неявного параметра `this`.
4. Построение иерархии классов. Основные правила. Приведите пример класса и иерархии классов.
5. Спецификаторы управления областью видимости. `public`, `protected`, `private`. Что они означают. Как правильно использовать.
6. Конструкторы класса. Типы конструкторов. Конструктор по умолчанию. Конструктор копирования. Конструктор с параметром.
7. Правила вызова конструкторов из производных классов при единичном и множественном наследовании. Списки инициализации при описании конструктора.
8. Статические поля и методы. Особенности, область применения. Вызов статических методов.
9. Деструкторы. Описание деструктора. Назначение деструктора. Условия вызова деструктора.
10. Наследование. Правила объявления производных классов. Множественное наследование.
11. Полиморфизм в ООП. Назначение. Приведите примеры полиморфизма. Механизмы раннего и позднего связывания методов. Таблицы виртуальных методов. Отличие директив `virtual` и `override`.
12. Шаблоны классов. Правила объявления шаблонов, преимущества шаблонов.
13. Стандартные контейнерные классы. Строки (`string`), достоинства и недостатки по сравнению с массивами символов (`char[]`). Описание строк. Основные операции и методы работы со строками.
14. Стандартный контейнерный класс – `array`. Создание и использование данных контейнеров. Понятие итератора. Назначение итераторов. Доступ к элементам, добавление, вставка, удаление и изменение элементов.
15. Стандартный контейнерный класс – `list`. Создание и использование данных контейнеров. Понятие итератора. Назначение итераторов. Доступ к элементам, добавление, вставка, удаление и изменение элементов.
16. Стандартный контейнерный класс – `vector`. Создание и использование данных контейнеров. Понятие итератора. Назначение итераторов. Доступ к элементам, добавление, вставка, удаление и изменение элементов.
17. Ассоциативный контейнер `map`. Назначение. Что такое ключ, что такое значение. Добавление и вставка элементов в контейнер. Удаление элементов. Доступ к элементу по ключу.
18. 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.

#### Экзамен 4 семестр

1. Основные типы данных языка C++, их назначение. Тип данных `auto`. Особенности использования. Основные сведения об операциях в языке C++. Приоритет выполнения операций. Выражения.

2. Общие сведения об операторах языка C++. Простой оператор. Составной оператор. Область видимости переменных. Общая структура программы на языке C++. Назначение разделов программы. Функция main.
3. Условный оператор. Оператор выбора switch.
4. Циклы в C++. Назначение циклов. Типы циклов.
5. Структурированные типы данных. Массивы. Одномерные и многомерные массивы. Цепочки (массивы) символов (строки). Представление строки в памяти. Операции со строками. Присваивание строк, сравнение строк, копирование строк и т.д. Управляющие символы. Структуры. Описание структур. Назначение структур. Примеры структур.
6. Функции в C++. Назначение функций. Описание функций, реализация функций. Имя функции. Тип возвращаемого результата. Параметры функции. Возврат результата из функции. Прототипы функций. Назначение. Параметры по умолчанию и правила их применения. Перегрузка функций. Назначение перегрузки. Правила перегрузки. Шаблоны функций. Преимущества шаблонов. Передача аргументов в функцию. Передача по значению. Передача по ссылке.
7. Динамическая память (куча). Указатели в C++. Назначение указателей. Определение указателей. Типы указателей. Операция разадресации. Адресная арифметика. Специальное значение NULL. Операции выделения и освобождения памяти (new и delete).
8. Поточковый ввод-вывод в файлы. Потoki ifstream, ofstream,fstream. Их назначение и отличие. Типовая процедура работы с файлом (чтение, запись), последовательность действий. Основные методы для работы с файловыми потоками. Методы open(), close(), eof(), seekg(), tellg(), get(), getline(), read(), write(). Режимы доступа к файлу. Типы файлов. Текстовые файлы, бинарные файлы. Отличия в работе с этими файлами. Методы чтения и записи данных в бинарный файл.
9. Понятие исключения (exception) в программе. Причины возникновения исключений. Обработка исключений. Операторы try...catch. Оператор throw.
10. Основные понятия ООП. Инкапсуляция. Наследование. Полиморфизм.
11. Классы, структуры, объекты. Отличия и сходства. Спецификаторы доступа. Статические и динамические члены.
12. Конструкторы и деструкторы. Типы. Списки инициализации.
13. Механизмы наследования. Единичное и множественное наследование. Полиморфизм. Реализация полиморфизма.
14. Библиотека STL. Стандартные контейнерные классы. Стандартные контейнеры. Ассоциативные контейнеры.
15. Класс thread. 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?
16. Назначение платформы .NET. Основная идеология, преимущества, достоинства и недостатки. JIT-компилятор. Особенности разработки программ, выполняемых в среде CLR. Разница в управляемом и неуправляемом коде на языке C++.
17. Основы языка C#. Основные отличия от C++. Типы данных, особенности реализации.

18. Переменные, операторы языка C#, выражения, операции. Общая структуры программы.
19. Классы. Основные понятия, отличия от C++. Иерархия классов. Понятия интерфейсов.
20. Делегаты, события и потоки выполнения. Назначение. Область применения.
21. Работа с файлами и потоками в C#.
22. Многозадачность и многопоточность в C#. Асинхронность и асинхронные методы.
23. Разработка оконных приложений на C#.
24. Тестирование программ. Важность процесса тестирования. Степень охвата программ тестами. Модульное тестирование. Создание модульных тестов на C#.

К сдаче зачета допускаются студенты, не имеющие задолженности по практическим и контрольным работам.

Критерии оценивания ответа на зачете:

**Зачтено:** Студент ответил на основные положения теоретического вопроса, допускаются незначительные ошибки, которые студент исправляет на месте.

Студент выполнил практическую работу не менее, чем на 65%.

**Не зачтено:** Студент не ответил на основные положения теоретического вопроса

Студент выполнил практическую работу менее, чем на 65%

К сдаче экзамена допускаются студенты, не имеющие академической задолженности по данной дисциплине за предыдущие семестры и по практическим и контрольным работам текущего семестра.

Критерии оценивания ответа на экзамене: *качество устных ответов на вопросы.*

Каждый вид работы оценивается по пятибалльной 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 разрознен-

ные, бессистемные знания, допускает ошибки в определении базовых понятий, искажает их смысл; не может практически применять теоретические знания.

## 8. Учебно-методическое и информационное обеспечение дисциплины (модуля)

### 8.1. Рекомендуемая литература

#### 8.1.1. Основная литература

1. Лубашева, Т. В. Основы алгоритмизации и программирования : учебное пособие / Т. В. Лубашева, Б. А. Железко. — Минск : Республиканский институт профессионального образования (РИПО), 2016. — 379 с. — ISBN 978-985-503-625-9. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <http://www.iprbookshop.ru/67689.html>
2. Карпович, Е. Е. Языки программирования интеллектуальных систем : учебник / Е. Е. Карпович. — Москва : Издательский Дом МИСиС, 2018. — 172 с. — ISBN 978-5-906953-51-3. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <http://www.iprbookshop.ru/84436.html>
3. Огнева, М. В. Программирование на языке C++: практический курс : учебное пособие для среднего профессионального образования / М. В. Огнева, Е. В. Кудрина. — Москва : Издательство Юрайт, 2020. — 335 с. — (Профессиональное образование). — ISBN 978-5-534-05780-5. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <http://www.biblio-online.ru/bcode/454231>
4. Токманцев, Т. Б. Алгоритмические языки и программирование : учебное пособие для СПО / Т. Б. Токманцев ; под редакцией В. Б. Костоусова. — 2-е изд. — Саратов, Екатеринбург : Профобразование, Уральский федеральный университет, 2019. — 102 с. — ISBN 978-5-4488-0510-3, 978-5-7996-2899-4. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <http://www.iprbookshop.ru/87785.html>
5. С#. Программирование на языке высокого уровня: Учебник для вузов. — СПб.: Питер, 2014. — 432 с.: ил. — (Серия «Учебник для вузов»).

#### 8.1.2. Дополнительная 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>

## 8.2. Перечень ресурсов информационно-телекоммуникационной сети «Интернет»

Интернет-ресурсы:

[metanit.com](http://metanit.com)

[professorweb.ru](http://professorweb.ru)

[habr.ru](http://habr.ru)

[stackoverflow.com](http://stackoverflow.com)

[ravesli.com](http://ravesli.com)

<https://docs.microsoft.com/>

### Электронно-библиотечные системы (ЭБС)

1. Удмуртская научно-образовательная Электронная библиотека (УдНОЭБ) (<http://elibrary.udsu.ru/xmlui/>)
2. ЭБС «Издательство Лань» (<https://e.lanbook.com/>)
3. ЭБС «Юрайт» (<https://www.biblio-online.ru/>)
- 4.

## 8.3. Перечень программного обеспечения

Microsoft Windows 7 – 10, Microsoft Office 7 – 2016

#### 8.4. Перечень профессиональных баз данных и информационных справочных систем

<https://docs.microsoft.com>

#### 9. Методические указания для обучающихся по освоению дисциплины (модуля)

1. Коврижных А.Ю. Основы алгоритмизации и программирования. Часть 1. Задачи и упражнения. Практикум [Электронный ресурс] : учебно-методическое пособие / А.Ю. Коврижных, Е.А. Конончук, Г.Е. Лузина. — Электрон. текстовые данные. — Екатеринбург: Уральский федеральный университет, ЭБС АСВ, 2016. — 52 с. — 978-5-7996-1886-5. — Режим доступа: <http://www.iprbookshop.ru/68449.html>
2. Коврижных А.Ю. Основы алгоритмизации и программирования. Часть 2. Расчетные работы. Практикум [Электронный ресурс] : учебно-методическое пособие / А.Ю. Коврижных, Е.А. Конончук, Г.Е. Лузина. — Электрон. текстовые данные. — Екатеринбург: Уральский федеральный университет, ЭБС АСВ, 2016. — 44 с. — 978-5-7996-1887-2. — Режим доступа: <http://www.iprbookshop.ru/68450.html>
3. Костюкова, Н. И. Программирование на языке Си : методические рекомендации и задачи по программированию / Н. И. Костюкова. — Новосибирск : Сибирское университетское издательство, 2017. — 160 с. — ISBN 978-5-379-02016-3. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <http://www.iprbookshop.ru/65289.html>
4. Мардашев, А. М. Задачи по программированию на C/C++ : учебно-методическое пособие / А. М. Мардашев, А. С. Панкратов, С. И. Салпагаров. — Москва : Российский университет дружбы народов, 2017. — 72 с. — ISBN 978-5-209-08034-3. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <http://www.iprbookshop.ru/90997.htm>
5. Технологии обработки информации на языках высокого уровня : учеб.-метод. пособие / М-во образования и науки РФ, ФГБОУ ВО "Удмуртский государственный университет", Ин-т математики, информ. технологий и физики, Каф. высокопроизводит. вычислений и парал. программирования ; сост. М. А. Ключков. - Ижевск : Удмуртский университет, 2018. - 37, [1] с. ; 60x84/16. - Библиогр.: с. 37. - Лицензион. договор № 40лб от 26.01.2018 (Интернет). - Режим доступа : <http://elibrary.udsu.ru/xmlui/handle/123456789/17017>
6. Носова, Л. С. Основы программной инженерии : учебно-методическое пособие / Л. С. Носова. — 2-е изд. — Челябинск, Саратов : Южно-Уральский институт управления и экономики, Ай Пи Эр Медиа, 2019. —

78 с. — ISBN 978-5-4486-0671-7. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <http://www.iprbookshop.ru/81488.html>

7. Костюкова, Н. И. Программирование на языке Си : методические рекомендации и задачи по программированию / Н. И. Костюкова. — Новосибирск : Сибирское университетское издательство, 2017. — 160 с. — ISBN 978-5-379-02016-3. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <http://www.iprbookshop.ru/65289.html>
8. Новиков П.В. Объектно-ориентированное программирование [Электронный ресурс] : учебно-методическое пособие к лабораторным работам / П.В. Новиков. — Электрон. текстовые данные. — Саратов: Вузовское образование, 2017. — 124 с. — 978-5-4487-0011-8. — Режим доступа: <http://www.iprbookshop.ru/64650.html>

#### **10. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)**

Материально-техническая база, необходимая для осуществления

На занятиях используются средства мультимедиа (чтение лекций с использованием слайд-презентаций, графических объектов, видео- аудио- материалов (через Интернет)), специализированных и офисных программ, баз данных (см. таблицу программного обеспечения). Преподаватель организует взаимодействие с обучающимися посредством электронной почты, компьютерного тестирования и локальной сети филиала. Также через электронную информационно-образовательную среду (ЭИОС) для студентов предусмотрена доступность рабочих программ и примерных фондов оценочных средств для любого участника учебного процесса, возможность консультирования обучающихся с преподавателем (проверка домашних заданий и т.д.) в любое время и в любой точке посредством сети Интернет (через электронную почту и социальные сети).

| №п/п | Название ПП             | Договор                             | Дата       | Кол-во лицензий           |
|------|-------------------------|-------------------------------------|------------|---------------------------|
| 1    | Microsoft Office 2010   | 0313100004015000052-0006194-01/1858 | 30.11.2015 | Не ограничено             |
| 2    | Microsoft Visual Studio | -                                   | -          | Свободно-распространяемое |

|  |                                                                       |  |  |    |
|--|-----------------------------------------------------------------------|--|--|----|
|  | Express 2010<br>или<br>Microsoft Visual<br>Studio Com-<br>munity 2019 |  |  | ПО |
|--|-----------------------------------------------------------------------|--|--|----|

## **11. Особенности организации образовательного процесса по дисциплине (модулю) для инвалидов и лиц с ограниченными возможностями здоровья**

Реализация дисциплины для лиц с ограниченными возможностями здоровья осуществляется с учетом особенностей психофизического развития, индивидуальных возможностей и состояния здоровья таких обучающихся.

Для адаптации программы освоения дисциплины используются следующие методы:

- для лиц с нарушениями слуха используются методы визуализации информации (презентации, использование компьютера для передачи текстовой информации, интерактивная доска, участие сурдолога и др.)
- для лиц с нарушениями зрения используются такие методы, как увеличение текста и картинки (в программах Windows), программы-синтезаторы речи, в том числе в ЭБС, звукозаписывающие устройства (диктофоны), компьютеры с соответствующим программно-аппаратным обеспечением и портативные компьютеризированные устройства.

Для маломобильных групп населения имеется необходимое материально-техническое обеспечение (пандусы, оборудованные санитарные комнаты, кнопки вызова персонала, оборудованные аудитории для лекционных и практических занятий), возможно применение ассистивных технологий и средств.

Форма проведения текущей и промежуточной аттестации для инвалидов и лиц с ограниченными возможностями здоровья устанавливается с учетом индивидуальных психофизических особенностей (устно, письменно на бумаге, письменно на компьютере, в форме тестирования и т.п.), при необхо-



димости выделяется дополнительное время на подготовку и предоставляются необходимые технические средства.