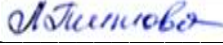


МИНОБРНАУКИ РОССИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»  
(ФГБОУ ВО «ВГУ»)

**УТВЕРЖДАЮ**

Заведующий кафедрой  
ядерной физики

 Титова Л. В.  
16.06.2022 г.

**РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ**

**Б1.О.32 Технология и языки программирования**

**1. Код и наименование специальности:**

14.05.02 Атомные станции: проектирование, эксплуатация и инжиниринг

**2. Специализация:**

Проектирование и эксплуатация атомных станций

**3. Квалификация выпускника:** инженер – физик

**4. Форма обучения:** очная

**5. Кафедра, отвечающая за реализацию дисциплины:**

кафедра ядерной физики

**6. Составители программы:**

к.ф.-м.н. доцент, Титова Лариса Витальевна

**7. Рекомендована:**

Научно – методическим советом физического факультета, протокол №6 от 14.06.2022 г.

**8. Учебный год:** 2025/2026

**Семестр(ы):** 8

## 9. Цели и задачи учебной дисциплины

*Целями освоения учебной дисциплины являются:*

- освоение современных инструментальных средств разработки программ.

*Задачи учебной дисциплины:*

- научить применять требования методологии структурного программирования при проектировании информационных моделей;
- разрабатывать и записывать на языке высокого уровня алгоритмы решения классических задач программирования;
- реализовывать технологию проектирования сверху-вниз; выбирать оптимальную структуру для представления данных.
- владеть объектно-ориентированным подходом, лежащим в основе большинства современных систем программирования;
- владеть компонентным подходом к построению программ;
- владеть навыками проектирования и реализации программ, управляемых событиями.

## 10. Место учебной дисциплины в структуре ООП:

Учебная дисциплина Технология и языки программирования относится к обязательной части Блока 1.

## 11. Планируемые результаты обучения по дисциплине/модулю (знания, умения, навыки), соотнесенные с планируемыми результатами освоения образовательной программы (компетенциями) и индикаторами их достижения:

Код	Название компетенции	Код(ы)	Индикатор(ы)	Планируемые результаты обучения
ОПК-4	Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения	ОПК-4.1	Умеет разрабатывать алгоритмы на языках программирования высокого уровня	Владеть: навыками решения задач научных исследований в области ядерной физики с помощью современных методов и средств компьютерных технологий.  Уметь: самостоятельно ставить и решать конкретные физические задачи научных исследований в области ядерной физики с использованием современной аппаратуры и компьютерных технологий.
		ОПК-4.2	Способен реализовать компьютерную программу для решения физических задач	
ПК-1	Способен проводить производственно - технологические исследования систем и оборудования атомных электрических станций и ядерных энергетических установок, участвовать во внедрении результатов исследований	ПК-1.2	Решает задачи применительно к реальным процессам, в том числе реализует решение в виде законченных компьютерных программ на языках программирования высокого уровня	Знать: методы и способы постановки и решения задач физических исследований, принципы действия, функциональные и метрологические возможности современной аппаратуры для физических исследований, возможности, методы и системы компьютерных технологий для физических теоретических и экспериментальных исследований

12. Объем дисциплины в зачетных единицах/час. — 4/144.

Форма промежуточной аттестации – зачет с оценкой

### 13. Трудоемкость по видам учебной работы

Вид учебной работы		Трудоемкость	
		Всего	По семестрам
			8 семестр
Аудиторные занятия		64	64
в том числе:	лекции	32	32
	практические	32	32
	лабораторные		
Самостоятельная работа		80	80
в том числе: курсовая работа (проект)			
Форма промежуточной аттестации			Зачет с оценкой
Итого:		144	144

#### 13.1. Содержание дисциплины

№ п/п	Наименование раздела дисциплины	Содержание раздела дисциплины	Реализация раздела дисциплины с помощью онлайн-курса, ЭУМК*
<b>1. Лекции</b>			
1.1	Понятие алгоритма и его характеристики как основного элемента программирования.	Формы представления алгоритмов, основные алгоритмические структуры.	<a href="https://edu.vsu.ru/enrol/index.php?id=10963">https://edu.vsu.ru/enrol/index.php?id=10963</a>
1.2	Механизмы работы с указателями	Машинная реализация механизма работы с указателями. Синтаксис и типы указателей. Инициализация указателей. Динамические переменные. Операторы выделения и освобождения памяти. Операции с указателями. Арифметические операции с указателями. Ссылки.	<a href="https://edu.vsu.ru/enrol/index.php?id=10963">https://edu.vsu.ru/enrol/index.php?id=10963</a>
1.3	Массивы	Определение массива в C++. Обработка одномерных и многомерных массивов. Использование датчика случайных чисел для формирования массива. Псевдодинамические массивы. Динамические массивы. Сортировка массивов. Поиск. Указатели и доступ к элементам массивов.	<a href="https://edu.vsu.ru/enrol/index.php?id=10963">https://edu.vsu.ru/enrol/index.php?id=10963</a>
1.4	Функции	Объявление и определение функций. Функции и массивы. Передача строк в качестве параметров функции.	<a href="https://edu.vsu.ru/enrol/index.php?id=10963">https://edu.vsu.ru/enrol/index.php?id=10963</a>
1.5	Типы данных, определяемые пользователем	Переименование типов. Перечисления. Структуры. Использование массивов и указателей при формировании структур. Сложные модели данных (списки, деревья, очереди, стеки).	<a href="https://edu.vsu.ru/enrol/index.php?id=10963">https://edu.vsu.ru/enrol/index.php?id=10963</a>
1.6	Потоки ввода-вывода в C++	Заголовочные файлы. Предопределенные объекты и потоки. Операции помещения и извлечения. Форматирование. Флаги форматирования. Манипуляторы. Ошибки потоков. Файловый ввод-вывод с применением потоков C++. Конструкторы файловых потоков. Открытие файлов в разных режимах. Ввод-вывод в файлы. Форматирование в памяти.	<a href="https://edu.vsu.ru/enrol/index.php?id=10963">https://edu.vsu.ru/enrol/index.php?id=10963</a>
<b>2. Практические занятия</b>			
2.1	Понятие алгоритма и его характеристики как основного элемента программирования.	Основы языка программирования C++, базовые алгоритмические конструкции и их реализация в C++. Запись алгоритмов на языке C++.	<a href="https://edu.vsu.ru/enrol/index.php?id=10963">https://edu.vsu.ru/enrol/index.php?id=10963</a>

2.2	Механизмы работы с указателями	Линейная, разветвляющаяся и циклическая структуры в языке С++	<a href="https://edu.vsu.ru/enrol/index.php?id=10963">https://edu.vsu.ru/enrol/index.php?id=10963</a>
2.3	Массивы	Работа с массивами, указателями, ссылками	<a href="https://edu.vsu.ru/enrol/index.php?id=10963">https://edu.vsu.ru/enrol/index.php?id=10963</a>
2.4	Функции	Функции и их обработка в языке С++.	<a href="https://edu.vsu.ru/enrol/index.php?id=10963">https://edu.vsu.ru/enrol/index.php?id=10963</a>
2.5	Потоки ввода-вывода в С++	Понятие объекта как совокупности свойств и событий в языке С++. Классы. Методы классов в языке С++	<a href="https://edu.vsu.ru/enrol/index.php?id=10963">https://edu.vsu.ru/enrol/index.php?id=10963</a>

### 13.2. Темы (разделы) дисциплины и виды занятий

№ п/п	Наименование темы (раздела) дисциплины	Виды занятий (количество часов)				Всего
		Лекции	Практические	Лабораторные	Самостоятельная работа	
1.	Понятие алгоритма и его характеристики как основного элемента программирования.	4	4		8	16
2.	Механизмы работы с указателями	4	4		8	16
3.	Массивы	6	6		14	26
4.	Функции	6	6		16	28
5.	Типы данных, определяемые пользователем	6	6		16	28
6.	Потоки ввода-вывода в С++	6	6		18	30
	Итого:	32	32		80	144

### 14. Методические указания для обучающихся по освоению дисциплины:

Студентам на лекциях необходимо вести подробный конспект и стараться понять материал курса. Для полного понимания материала следует активно использовать консультации. Для самостоятельного изучения разделов курса, рекомендованных преподавателем, необходимо пользоваться основной и дополнительной литературой, интернет-ресурсами.

На практических занятиях необходимо уметь решать задачи и анализировать решение, на устных опросах обучаемый должен уметь демонстрировать полученные на лекциях и практических занятиях знания, умения и навыки, отвечать на поставленные вопросы, поддерживать дискуссию по существу вопроса.

Методическое обеспечение аудиторной работы: учебно-методические пособия для студентов, учебники и учебные пособия, электронные и Интернет-ресурсы.

Методическое обеспечение самостоятельной работы: учебно-методические пособия по организации самостоятельной работы, контрольные задания и тесты в бумажном и электронном вариантах, тестирующие системы, дистанционные формы общения с преподавателем. Контроль самостоятельной работы реализуется с помощью опросов, тестов, вопросов по темам заданий и т.д.

### 15. Перечень основной и дополнительной литературы, ресурсов интернет, необходимых для освоения дисциплины

а) основная литература:

№ п/п	Источник
1.	Лафоре, Р. Объектно-ориентированное программирование в С+ / Р. Лафоре; Пер. с англ. А.Кузнецов и др. — 4-е изд. — СПб.: Питер, 2003.— 923 с.: ил. — (Классика Computer Science) .— Парал. ти. л. англ. — ISBN 5-94723-302-9

2.	Гниденко, И. Г. Технологии и методы программирования: учебное пособие для вузов / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — Москва: Издательство Юрайт, 2022. — 235 с. — (Высшее образование). — ISBN 978-5-534-02816-4. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: <a href="https://urait.ru/bcode/489920">https://urait.ru/bcode/489920</a>
3.	Оксюта, О. В. Объектно-ориентированное программирование: учебное пособие / О. В. Оксюта, В. В. Лавлинский. — Воронеж: ВГЛТУ, 2018. — 96 с. — Текст: электронный // Лань: электронно-библиотечная система. — URL: <a href="https://e.lanbook.com/book/118684">https://e.lanbook.com/book/118684</a>
4.	Подбельский, В. В. Курс программирования на языке Си : учебник / В. В. Подбельский, С. С. Фомин. — Москва: ДМК Пресс, 2012. — 384 с. — ISBN 978-5-94074-449-8. — Текст: электронный // Лань : электронно-библиотечная система. — URL: <a href="https://e.lanbook.com/book/4148">https://e.lanbook.com/book/4148</a>

б) дополнительная литература:

№ п/п	Источник
5.	Советов, Б. Я. Информационные технологии: теоретические основы: учебное пособие / Б. Я. Советов, В. В. Цехановский. — 2-е изд., стер. — Санкт-Петербург: Лань, 2022. — 444 с. — ISBN 978-5-8114-1912-8. — Текст: электронный // Лань: электронно-библиотечная система. — URL: <a href="https://e.lanbook.com/book/209876">https://e.lanbook.com/book/209876</a>
6.	Павловская, Т. А. Программирование на языке высокого уровня С#: учебное пособие / Т. А. Павловская. — 2-е изд. — Москва: ИНТУИТ, 2016. — 245 с. — Текст: электронный // Лань: электронно-библиотечная система. — URL: <a href="https://e.lanbook.com/book/100413">https://e.lanbook.com/book/100413</a>

в) информационные электронно-образовательные ресурсы (официальные ресурсы интернет)\*:

№ п/п	Ресурс
7.	<a href="http://www.lib.vsu.ru">www.lib.vsu.ru</a> – ЗНБ ВГУ.
8.	<a href="https://edu.vsu.ru">https://edu.vsu.ru</a> – Электронный университет ВГУ
9.	<a href="https://e.lanbook.com">https://e.lanbook.com</a> – ЭБС «Лань»
10.	<a href="https://www.studentlibrary.ru">https://www.studentlibrary.ru</a> – ЭБС «Консультант студента»
11.	<a href="https://urait.ru">https://urait.ru</a> – Образовательная платформа «ЮРАЙТ»
12.	<a href="https://rucont.ru">https://rucont.ru</a> - Информационно-телекоммуникационная система «Контекстум»

## 16. Перечень учебно-методического обеспечения для самостоятельной работы

№ п/п	Источник
1.	Пушкарев, А. Н. Языки программирования: учебно-методическое пособие / А. Н. Пушкарев. — Тюмень: ТюмГУ, 2018. — 48 с. — Текст: электронный // Лань: электронно-библиотечная система. — URL: <a href="https://e.lanbook.com/book/110182">https://e.lanbook.com/book/110182</a>

## 17. Образовательные технологии, используемые при реализации учебной дисциплины, включая дистанционные образовательные технологии (ДОТ, электронное обучение (ЭО), смешанное обучение):

При проведении занятий по дисциплине используются следующие образовательные технологии:

- активные и интерактивные формы проведения занятий;
- компьютерные технологии при проведении занятий;
- презентационные материалы и технологии при объяснении материала на лекционных и практических занятиях;
- специализированное оборудование при проведении лабораторных работ;
- разбор конкретных ситуаций при постановке целей и задач к разработке прикладных программ, при выборе программного обеспечения по установленным критериям, при разработке программ по предусмотренным алгоритмам и метода.

Для самостоятельной работы используется ЭБС Университетская библиотека online - [www.lib.vsu.ru](http://www.lib.vsu.ru) - ЗНБ ВГУ. Программное обеспечение, применяемое при реализации дисциплины – Microsoft Windows, LibreOffice, CodeBlocks, Adobe Reader, Mozilla FireFox.

Дистанционные образовательные технологии (ДОТ) применяются с использованием образовательного портала «Электронный университет ВГУ».

## 18. Материально-техническое обеспечение дисциплины:

Компьютерный класс (для проведения занятий семинарского типа, текущего контроля и промежуточной аттестации)

Специализированная мебель, компьютеры (10 шт.), объединенные в локальную сеть с возможностью подключения к сети «Интернет»

Microsoft Windows 10, LibreOffice, MikTex, Adobe Reader Mozilla Firefox, Maxima, Lazarus, CodeBlocks

Компьютерный класс, аудитория для групповых и индивидуальных консультаций, помещение для самостоятельной работы

Специализированная мебель, компьютеры с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду университета

Microsoft Windows 10, LibreOffice, Adobe Reader

---

## 19. Оценочные средства для проведения текущей и промежуточной аттестаций

Порядок оценки освоения обучающимися учебного материала определяется содержанием следующих разделов дисциплины:

№ п/п	Наименование раздела дисциплины (модуля)	Компетенция(и)	Индикатор(ы) достижения компетенции	Оценочные средства
1.	Понятие алгоритма и его характеристики как основного элемента программирования.	ОПК-4 ПК-1	ОПК-4.1 ОПК-4.2 ПК-1.2	Практические задания, собеседование по вопросам к зачету, открытые вопросы ФОС п. 21
2.	Механизмы работы с указателями			
3.	Массивы			
4.	Функции			
5.	Типы данных, определяемые пользователем			
6.	Потоки ввода-вывода в C++			
Промежуточная аттестация форма контроля – зачет с оценкой				Перечень вопросов к зачету Пункт 20.2

## 20. Типовые оценочные средства и методические материалы, определяющие процедуры оценивания

### 20.1. Текущий контроль успеваемости

Контроль успеваемости по дисциплине осуществляется с помощью следующих оценочных средств: практические задания.

#### Примерный перечень практических заданий:

1. Написать программу, которая в одномерном массиве, состоящем из  $n$  вещественных элементов, вычислит номер максимального по модулю элемента массива, сумму элементов массива, расположенных после первого положительного элемента, преобразовать массив таким образом, чтобы сначала располагались все элементы, целая часть которых лежит в интервале, заданном пользователем.

2. Написать программу работы с базой данных радиоактивных источников. База хранится в текстовом файле, каждая строка содержит информацию об одном источнике. Формат записи:

химический элемент, массовое число, заряд, период полураспада, вид распада, дата принятия в хранилище, дата утилизации. Программа должна обеспечивать поиск в базе по заданным критериям, корректировку и дополнение базы.

3. Написать программу, которая для целочисленной матрицы размером 10x20 определяет среднее арифметическое элементов и количество положительных элементов в каждой строке.

4. Написать программу, которая определяет максимальные элементы в одномерных массивах различных арифметических типов.

5. Написать программу вывода таблицы значений гиперболического косинуса для аргумента, изменяющегося в заданных пределах с заданным шагом. Значения функции вычислять с помощью разложения в ряд Тейлора с заданной точностью.

6. Написать программу упорядочения массива методом быстрой сортировки, используя рекурсию.

7. Написать программу, которая определяет, сколько чисел содержится в каждой строке текстового файла, длина каждой строки не превышает 100 символов.

8. Вывести на экран содержимое текстового файла, найти в файле самое длинное слово.

9. Написать программу, которая считывает из текстового файла три предложения и выводит их в обратном порядке.

10. Написать программу, которая упорядочивает строки прямоугольной целочисленной матрицы по возрастанию сумм их элементов.

Описание критериев и шкалы оценивания компетенций (результатов обучения) при аттестации

Критерии оценивания компетенций	Шкала оценок
Обучающийся демонстрирует полное соответствие знаний, умений, навыков приведенным в таблицах показателям, свободно оперирует приобретенными знаниями, умениями, применяет их при решении практических задач	Отлично
Обучающийся демонстрирует соответствие знаний, умений, навыков приведенным в таблицах показателям, но допускает незначительные ошибки, неточности, испытывает затруднения при решении практических задач	Хорошо
Обучающийся демонстрирует неполное соответствие знаний, умений, навыков приведенным в таблицах показателям, допускает значительные ошибки при решении практических задач	Удовлетворительно
Обучающийся демонстрирует явное несоответствие знаний, умений, навыков приведенным в таблицах показателям	Неудовлетворительно

## 20.2. Промежуточная аттестация

Промежуточная аттестация по дисциплине осуществляется с помощью следующих оценочных средств: собеседование по вопросам к зачету.

### Перечень вопросов к зачету:

1. Понятие алгоритма и его характеристики как основного элемента программирования.
2. Основные алгоритмические структуры.
3. Разработка схемы алгоритма. Составление программы на алгоритмическом языке.
4. Разработка алгоритмов «сверху - вниз».
5. Формы представления алгоритмов, основные алгоритмические структуры.
6. Нисходящее проектирование.
7. Словесное описание алгоритма на естественном языке (вербальная форма);
8. Идеология системы Windows, основанная на событиях.

9. Построчная запись алгоритма. Схема алгоритма.
10. События пользовательские, системные, программные.
11. Базовые элементы языка программирования C++.
12. Структура программы. Схема работы компилятора.
13. Переменные и выражения. Основные операторы языка: оператор «выражение», операторы ветвления, операторы цикла, операторы передачи управления.
14. Машинная реализация механизма работы с указателями. Синтаксис и типы указателей. Инициализация указателей
15. Динамические переменные. Операторы выделения и освобождения памяти. Операции с указателями. Арифметические операции с указателями. Ссылки.
16. Определение массива в C++. Обработка одномерных и многомерных массивов. Использование датчика случайных чисел для формирования массива.
17. Объявление и определение функций. Функции и массивы. Передача строк в качестве параметров функции.
18. Потоки ввода – вывода. Ошибки потоков. Файловый ввод-вывод с применением потоков C++. Конструкторы файловых потоков. Открытие файлов в разных режимах. Ввод-вывод в файлы.

Промежуточная аттестация проводится в соответствии с Положением о промежуточной аттестации обучающихся по программам высшего образования.

Контрольно-измерительные материалы промежуточной аттестации включают в себя теоретические вопросы, позволяющие оценить уровень полученных знаний и/или практическое(ие) задание(я), позволяющее(ие) оценить степень сформированности умений и(или) навыков, и(или) опыт деятельности.

При оценивании используются количественные или качественные шкалы оценок. Критерии оценивания приведены выше.

Описание критериев и шкалы оценивания компетенций (результатов обучения) при аттестации

Критерии оценивания компетенций	Шкала оценок
Обучающийся демонстрирует полное соответствие знаний, умений, навыков приведенным в таблицах показателям, свободно оперирует приобретенными знаниями, умениями, применяет их при решении практических задач	Зачтено
Обучающийся демонстрирует соответствие знаний, умений, навыков приведенным в таблицах показателям, но допускает незначительные ошибки, неточности, испытывает затруднения при решении практических задач	
Обучающийся демонстрирует неполное соответствие знаний, умений, навыков приведенным в таблицах показателям, допускает значительные ошибки при решении практических задач	
Обучающийся демонстрирует явное несоответствие знаний, умений, навыков приведенным в таблицах показателям	Незачтено

**Пример контрольно-измерительного материала (КИМ)**

УТВЕРЖДАЮ

Заведующий кафедрой  
ядерной физики





- 1) **int**
  - 2) char
  - 3) bool
  - 4) unsigned char
7. Поразрядные операции &, |, ^ и ~ применимы к
- 1) операндам типа bool
  - 2) **операндам типа int**
  - 3) операндам типа float
  - 4) операндам типа double
8. Чему равно числовое значение целой переменной  $x = 8/(\sqrt{4} * 0.281)$ ?
- 1) **14**
  - 2) 13
  - 3) 12
  - 4) 11
9. Какое выражение не содержит синтаксических ошибок?
- 1) **-0.18\*a/r-0.2\*t**
  - 2) (-0.18)\*a)/(r-0.2\*t)
  - 3) cos/^2\*x+0,2
  - 4) (-0.18)\*a)\(r-0.2\*t))
10. Где правильно инициализирована переменная:
- 1) int a
  - 2) int a;
  - 3) **int a=5;**
  - 4) int a=5
11. В списке параметров функции printf могут присутствовать
- 1) **имена переменных**
  - 2) ключевые слова
  - 3) вещественные числа
  - 4) операторы
12. В условном операторе между ключевыми словами if и else после выражения в скобках может находиться
- 1) любой оператор, кроме оператора перехода
  - 2) **любой простой или составной оператор**
  - 3) не больше двух операторов
  - 4) только один простой оператор
13. Какой из операторов не является оператором цикла?
- 1) for
  - 2) while
  - 3) do ... while
  - 4) **switch**
14. Выберите пример указателя-константы.
- 1) **char\* const pc**
  - 2) char\* pc
  - 3) const char\* const pc
  - 4) невозможно определить указатель-константу
15. Какая операция используется для доступа к полям структуры?
- 1) операция :
  - 2) **операция .**

- 3) операция ,  
4) операция ::
16. Какая операция используется для доступа к полям структуры через указатель?  
**1) операция ->**  
2) операция .  
3) операция ,  
4) операция ::
17. Выберите неверное утверждение.  
**1) глобальные переменные нужно передавать в функцию через список параметров**  
2) место под формальные параметры выделяется на стадии выполнения программы  
3) компилятор инициализирует нулями глобальные переменные  
4) функция может возвращать значение стандартного типа
18. Выберите неверное утверждение.  
1) для локальных переменных выделяется место в стеке  
**2) компилятор инициализирует нулями локальные переменные**  
3) для параметров, передаваемых по адресу, в стек заносится только их адрес  
4) функция может возвращать через список параметров более одного значения
19. Список параметров функции описан как "void". Что можно сказать о параметрах этой функции.  
Ответ:  
1) аргумент может быть только константой  
**2) у функции нет параметров**  
3) аргумент может быть только переменной  
4) аргумент может быть выражением любого типа
20. Заголовок функции имеет вид: `void f(int& a, float b, char c);` Укажите неверный вариант вызова функции, если переменные в вызывающей функции описаны так: `int a; char b; float d,x;`  
Ответ:  
1) `f(a,a,b);`  
2) `f(a,x,'b');`  
3) `f(a,d-x/2,b);`  
**4) `f(2,d,b);`**
21. Какое из следующих утверждений истинно?  
**1) перегруженные функции должны отличаться типом параметров**  
2) перегруженные функции должны отличаться именами параметров  
3) перегруженные функции должны отличаться типом возвращаемого значения  
4) перегруженные функции отличий не имеют
22. Как подключить стандартную библиотеку `iostream`?  
1) `#include "iostream.h"`  
2) `#include <iostream.h>`  
3) `#include 'iostream.h'`  
**4) `#include <iostream>`**
23. Как правильно объявить одномерный массив с именем `array`, состоящий из 10 элементов целочисленного типа?  
**1) `int array [10];`**  
2) `array [10]`  
3) `int array(10);`

- 4) `int array[0..10];`
24. Дан массив `int array[5] = { 3, 10, 7, 9, 2}`. Как обратиться к числу 7?
- 1) `array[7];`
  - 2) `array[2];`**
  - 3) `array[3];`
  - 4) `array[];`
25. Что такое двумерный массив:
- 1) Массив, объединяющий два массива
  - 2) Массив из двух строк
  - 3) Массив из двух столбцов
  - 4) Массив одномерных массивов**

Задания с развернутым ответом:

- 1) Чему равно значение выражения `(!( a && b || c))`, где `a`, `b` и `c` - величины типа `bool`, имеющие значения `true`, `true` и `false`. Ответ: `false`
- 2) Чему равно значение выражения `(a && ! b || c)`, где `a`, `b` и `c` - величины типа `bool`, имеющие значения `false`, `true` и `true` соответственно? Ответ: `true`
- 3) Можно ли присвоить значения целого типа - вещественной переменной? Ответ: можно
- 4) Можно ли присвоить значение символьного типа - целой переменной? Ответ: можно
- 5) Чему будет равно значение переменной `A` после выполнения фрагмента программы, если `B = 3`?

```
int A = 2, B=3, C;  
if (B <= 4)  
{C = 1;}  
else  
{C = 0;  
A = C;}  
Ответ: 1
```

- 6) Чему будет равно значение переменной `A` после выполнения фрагмента программы, если `B = 5` ?

```
if (B <= 4)  
C = 1;  
else C = 0;  
A = C;  
Ответ: 0
```

- 7) Чему будет равно значение переменной `a` после выполнения фрагмента программы, если `b = 2`?

```
int a = 1;  
if ((b < 0) || (b > 1)) a = 3;  
if (b > 5) a = 5;  
Ответ: 3
```

- 8) Чему будет равно значение переменной `a` после выполнения фрагмента программы, если `b = 2`?

```
a = 1;
if ((b > 0) && (b < 3)) a = 3;
else
if (b > 5) a = 5;
Ответ: 3
```

9) Чему будет равно значение переменной m после выполнения программы?

```
int main ()
{
int k;
int m;
k = 0; m = 0;
while (k < 3)
++k;
++m;
return m;
}
Ответ: 1
```

10) Чему будет равно значение переменной m после выполнения программы?

```
main()
{
int k;
int m;
k = 3; m = 0;
while (k < 3)
{
++k; ++m;
}
}
Ответ: 0
```

11) Чему будет равно значение переменной m после выполнения программы?

```
main()
{
int k;
int m;
m = 0;
for (k = 1; k <= 2; k++)
++m; ++m;
}
Ответ: 3
```

12) Чему будет равно значение переменной m после выполнения программы?

```
main()
{
int k;
int m;
m = 1;
for (k = 1; k <= 2; k++)
{
```

```
++m; k+=2;
}}
```

Ответ: 2

13) Что будет выведено на экран в результате работы фрагмента программы

```
for (int ii = 0; ii < 3; ++ ii)
{
    switch (ii)
    {
        case 0: cout << "ZERO";
        case 1: cout << "ONE"; continue;
        case 2: cout << "TWO"; break;
    }
    cout << endl;
}
```

Ответ: ZEROONEONETWO

14) Какое число получится в результате работы программы?

```
#include <stdio.h>
#include <math.h>
main()
{
    const int a[8] = {3,8,0,-6,0,-1,-9,3};
    int i,k;
    k = abs(a[7]);
    for (i = 0; i < 8; i++)
        if (abs(a[i]) > k) k = abs(a[i]) - 2;
    printf("%d",k);
}
```

Ответ: 7

15) Какое число получится в результате работы программы?

```
#include <stdio.h>
#include <math.h>
main()
{
    const int a[8] = {3,8,0,-6,0,-1,-9,3};
    int i,k;
    k = abs(a[2]);
    for (i = 0; i < 8; i++)
        if (a[i] > k) k = abs(a[i]) - 2;
    printf("%d",k);
}
```

Ответ: 6

16) Что будет выведено на экран в результате работы программы?

```
#include <stdio.h>
main()
```

```

{
const char a[] = "abcrcaab";
int i; char k;
k = a[0];
for (i = 1; i < 8; i++)
    if (a[i] <= k) k = a[i];
printf("%c",k);
}

```

Ответ: a

17) Какое значение получит p[0] в данном примере?

```

int a[2][3]={{1,2,3},{7,8,9}};
int *p=&a[0][0];

```

p+=3;

Ответ: 7

18) Какую величину вычисляет программа?

```

#include <iostream>
main(){
int a[3][4];
int i, j, k;
for (i = 0; i < 3; i++)
for (j = 0; j < 4; j++) cin>>a[i][j];
for (i = 0; i < 3; i++)
{
k = 0;
for (j = 0; j < 4; j++)
if (a[i][j] > 0) ++k;
}
cout<<k;
}

```

Ответ: количество положительных элементов массива в последней строке

19) Какую величину вычисляет программа?

```

#include <iostream>
main()
{ int a[3][4];
int i, j, k;
for (i = 0; i < 3; i++)
for (j = 0; j < 4; j++) cin>>a[i][j];
for (i = 0; i < 4; i++) {
k = 0;
for (j = 0; j < 3; j++)
if (a[j][i] > 0) ++k;
}
cout<<k;
}

```

Ответ: количество положительных элементов массива в последнем столбце

20) Какую величину вычисляет программа?

```
#include <stdio.h>
main()
{ int a[3][4];
  int i, j, k;
  for (i = 0; i < 3; i++)
    for (j = 0; j < 4; j++) cin>>a[i][j];
  k = -1;
  for (i = 0; i < 4; i++) {
    for (j = 0; j < 3; j++)
      if (a[j][i] == 0) k = i;
  }
  cout<<k;
}
```

Ответ: номер последнего из столбцов массива, содержащих хотя бы один элемент, равный 0

21) Что будет выведено на экран в результате работы программы?

```
#include <stdio.h>
void f(char a, char& d)
{
  a = 'в'; d = 'с';
}
int main()
{
  char a[]="глаз";
  f(a[0],a[3]);
  puts(a);
  return 0;
}
```

Ответ: глас

22) Что делает следующий код?

```
int main() {int n;
  cout<<"Введите целочисленное значение";
  cin>>n;
  int *a=new int[n];
  return 0;}
```

Ответ: Создает динамический массив из n элементов

23) Какой метод сортировки массива представлен в коде?

```
for (int i=0;i<size-1;i++)
{for (int j=0;j<size-i-1;j++)
{if (arr[j]>arr[j+1])
{temp=arr[j];
arr[j]=arr[j+1];
arr[j+1]=temp;}}}
```

Ответ: сортировка пузырьком



24) Что делает следующий код?

```
int b=0;  
for (int count = 0; count < 10; count++)  
b+=array[count];
```

Ответ: подсчитывает сумму первых 10 элементов массива array

25) Дан массив `int A[5] = { 3, 10, 7, 9, 2}`. Как обратиться к числу 7?

Ответ: `A[2]`