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

УТВЕРЖДАЮ

Заведующий кафедрой
функционального анализа
и операторных уравнений

Каменский М.И.
25.05.2023 г.

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

Б1.О.04.03 Технология и методы программирования

1. Код и наименование направления специальности: 10.05.04 информационно-аналитические системы безопасности

2. Профиль специализации: Автоматизация информационно-аналитической деятельности

3. Квалификация выпускника: специалист по защите информации

4. Форма обучения: очная

5. Кафедра, отвечающая за реализацию дисциплины: функционального анализа и операторных уравнений

6. Составители программы: Груздев Денис Владиславович, преподаватель, математический факультет, кафедра функционального анализа и операторных уравнений .

7. Рекомендована: НМС математического факультета, протокол от 25.05.2023, № 0500-06

8. Учебный год: 2024-2025 **Семестр:** 3,4

9. **Цели и задачи учебной дисциплины:**

Цели изучения дисциплины:

- изучение современных методов и технологий программирования, средств разработки алгоритмов и программ для решения широкого круга задач.

-подготовка в области применения основных методов программирования для решения конкретных задач в определенной языковой среде, а также задач обработки текстовой и числовой информации, алгоритмизации.

Задачи учебной дисциплины:

- изучить традиционные структуры данных, основные требования методологии структурного программирования, как технологической основы разработки качественных программных компонентов, понятие статических и динамических данных, реализацию вызова процедур в языках с блочной структурой, рекурсию, примеры базовых структур данных, подходы процедурного, модульного, программирования.

10. Место учебной дисциплины в структуре ООП: дисциплина Технология и методы программирования относится к обязательной части блока Б1.

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

Код	Название компетенции	Коды	Индикаторы	Планируемые результаты обучения
ОПК-7	Способен создавать программы на языках высокого уровня, применять методы и инструментальные средства программирования для решения профессиональных задач, осуществлять обоснованный выбор инструментария программирования;	ОПК-7.2	Способен применять технологии и методы программирования для разработки программного обеспечения	**знать:** Языки и системы программирования, инструментальные средства разработки **уметь:** Составлять программы для поставленных задач, Разрабатывать специальные ИАС **владеть:** Средствами программирования, стандартными алгоритмическими механизмами для обработки информации;

12. Объем дисциплины в зачетных единицах/час — 8/288

Форма промежуточной аттестации*: Экзамен, зачет.*

13. Трудоемкость по видам учебной работы

Виды учебной работы	Трудоемкость		
	Всего	По семестрам	
		№ семестра 3	4
Аудиторные занятия	136	68	68
В том числе: лекции	68	34	34

практические			
лабораторные	68	34	34
Самостоятельная работа	80	58	22
Форма промежуточной аттестации (*зачет – 0 час./экзамен – **36** час.)*		36	36
Итого:	288	162	126

13.1. Содержание дисциплины

№ п/п	Наименование раздела дисциплины	Содержание раздела дисциплины
1. Лекции		
1	Понятие об архитектуре ЭВМ	Процессор и система его команд, структура памяти ЭВМ и способы адресации, выполнение команды в процессоре, взаимодействие процессора, памяти и периферийных устройств.
2	Операционные системы	Понятие об операционной системе: процесс, состояние процесса, прерывание, планирование процессов, понятие о тупиках и способах их устранения.
3	Введение в C++	Общая характеристика языка, технология разработки программ.
4	Типы данных и выражения	Алфавит, идентификаторы, операции, выражения, операторы, классификация типов данных, переменные и константы
5	Управляющие структуры	Организация алгоритмов ветвления и циклов, выбор циклов.
6	Массивы и указатели	Понятие массива, инициализация массивов, ссылки и указатели, указатели и массивы, динамические массивы. Двумерные массивы.
7	Функции	Общие сведения о функциях, функции с переменным числом параметров, рекурсивные и подставляемые функции, области действия переменных, массивы в качестве параметров функций.
8	Сортировки	Сортировки массивов: пузырьком, вставками, выбором, быстрая сортировка, турнирная и пирамидальная сортировки.
9	Файлы и потоки ввода-вывода	Назначение файлов, потоки ввода-вывода, проверка ошибок выполнения файловых операций, символьный ввод-вывод, операторы ввода-вывода.
10	Динамические структуры	Стек, очередь, список, дерево, дерево поиска.
11	Основные принципы ООП	Структурный подход в программировании, инкапсуляция, наследование, полиморфизм.
12	Классы и объекты	Способы описания классов. Создание объектов. Обращение к атрибутам и методам объектов. Конструкторы и деструкторы классов. Перегрузка операций. Функции друзья.
13	Наследование классов	Полиморфное открытое наследование. Статическое и динамическое связывание. Виртуальные методы. Абстрактные классы. Закрытое и защищенное наследование.
14	Обработка ошибок	Попытка классификации ошибок. Сообщение об ошибке с помощью возвращаемого значения. Исключительные ситуации. Обработка исключительных ситуаций, операторы try и catch. Классы исключений.
15	Эволюция вычислительных систем. Архитектура вычислительных систем	Архитектура сетей. Классификации сетей. Понятия локальных, региональных, глобальных сетей. Требования, предъявляемые к сетям.
16	Многоуровневая модель OSI. Стандарты и стеки протоколов.	Модель OSI. Стек OSI. Протоколы и стеки протоколов. Спецификации стандартов. Стек протоколов TCP/IP.
17	Топологии построения сетей. Методы доступа. Адресация, IP адреса V4, V6	Топологии построения сетей. Методы доступа. ЛВС и компоненты ЛВС. Адресация, IP адреса V4, V6. DNS

18	Физическая среда передачи данных. Сетевое оборудование.	Физическая среда передачи данных. Кабели связи, линии связи и каналы связи. Сетевые адаптеры, повторители, концентраторы, мосты, коммутаторы, маршрутизаторы, шлюзы.
19	Сетевые операционные системы.	Сетевые ОС NetWare, семейство сетевых ОС Windows Server, семейство ОС UNIX, сетевая ОС Linux.
20	Стандартные локальные сети.	Сеть Ethernet, сет Fast Ethernet, сеть Token-Ring, сеть Arcnet, сеть FDDI, сеть 100VG-AnyLan, сверхвысокоскоростные сети
2. Лабораторные работы		
1	Форматы данных	Представление целочисленных, вещественных, символьных и др. данных
2	Введение в C++	Технология разработки программ.
3	Типы данных и выражения	Алфавит, идентификаторы, операции, выражения, операторы, классификация типов данных, переменные и константы
4	Управляющие структуры	Организация алгоритмов ветвления и циклов, выбор циклов.
5	Массивы и указатели	Понятие массива, инициализация массивов, ссылки и указатели, указатели и массивы, динамические массивы. Двумерные массивы.
6	Функции	Общие сведения о функциях, функции с переменным числом параметров, рекурсивные и подставляемые функции, области действия переменных, массивы в качестве параметров функций.
7	Сортировки	Сортировки массивов: пузырьком, вставками, выбором, быстрая сортировка, турнирная и пирамидальная сортировки.
8	Файлы и потоки ввода-вывода	Назначение файлов, потоки ввода-вывода, проверка ошибок выполнения файловых операций, символьный ввод-вывод, операторы ввода-вывода.
9	Динамические структуры	Стек, очередь, список, дерево, дерево поиска.
10	Основные принципы ООП	Структурный подход в программировании, инкапсуляция, наследование, полиморфизм.
11	Классы и объекты	Способы описания классов. Создание объектов. Обращение к атрибутам и методам объектов. Конструкторы и деструкторы классов. Перегрузка операций. Функции друзья.
12	Наследование классов	Полиморфное открытое наследование. Статическое и динамическое связывание. Виртуальные методы. Абстрактные классы. Закрытое и защищенное наследование.
13	Обработка ошибок	Попытка классификации ошибок. Сообщение об ошибке с помощью возвращаемого значения. Исключительные ситуации. Обработка исключительных ситуаций, операторы try и catch. Классы исключений.
14	Архитектура вычислительных систем	Архитектура сетей. Классификации сетей. Понятия локальных, региональных, глобальных сетей. Требования, предъявляемые к сетям.
15	Многоуровневая модель OSI. Стандарты и стеки протоколов.	Модель OSI. Стек OSI. Протоколы и стеки протоколов. Спецификации стандартов. Стек протоколов TCP/IP.
16	Топологии построения сетей. Методы доступа. Адресация, IP адреса V4, V6	Топологии построения сетей. Методы доступа. ЛВС и компоненты ЛВС. Адресация, IP адреса V4, V6. DNS
17	Физическая среда передачи данных. Сетевое оборудование.	Физическая среда передачи данных. Кабели связи, линии связи и каналы связи. Сетевые адаптеры, повторители, концентраторы, мосты, коммутаторы, маршрутизаторы, шлюзы.

18	Сетевые операционные системы.	Сетевые ОС NetWare, семейство сетевых ОС Windows Server, семейство ОС UNIX, сетевая ОС Linux.
19	Стандартные локальные сети.	Сеть Ethernet, сет Fast Ethernet, сеть Token-Ring, сеть Arcnet, сеть FDDI, сеть 100VG-AnyLan, сверхвысокоскоростные сети
3. Практические работы		

13.2. Темы (разделы) дисциплины и виды занятий

№ п/п	Наименование раздела дисциплины	Виды занятий (часов)				
		Лекции	Практические	Лабораторные	Самостоятельная работа	Всего
1	Понятие об архитектуре ЭВМ	2		2	2	6
2	Операционные системы	2		2	2	6
3	Введение в C++	2		2	2	6
4	Типы данных и выражения	2		2	2	6
5	Управляющие структуры	2		2	2	6
6	Массивы и указатели	2		2	2	6
7	Функции	4		4	6	14
8	Сортировки	4		4	6	14
9	Файлы и потоки ввода-вывода	4		4	4	12
10	Динамические структуры	4		4	6	14
11	Основные принципы ООП	4		4	6	14
12	Классы и объекты	4		4	6	14
13	Наследование	4		4	6	14
14	Обработка ошибок	4		4	4	12
15	Эволюция вычислительных систем. Архитектура вычислительных систем	4		4	4	12
16	Многоуровневая модель OSI. Стандарты и стеки протоколов.	4		4	4	12
17	Топологии построения сетей. Методы доступа. Адресация, IP адреса V4, V6	4		4	4	12
18	Физическая среда передачи данных. Сетевое оборудование.	4		4	4	12
19	Сетевые операционные системы.	4		4	4	12

20	Стандартные локальные сети.	4		4	4	12
	Итого	68		68	80	216

14. Методические указания для обучающихся по освоению дисциплины

В процессе преподавания дисциплины используются такие виды учебной работы, как лекции, лабораторные занятия, а также различные виды самостоятельной работы обучающихся. На лекциях рассказывается теоретический материал, на лабораторных занятиях решаются примеры по теоретическому материалу, прочитанному на лекциях.

При изучении курса «Технология и методы программирования» обучающимся следует внимательно слушать и конспектировать материал, излагаемый на аудиторных занятиях. Для его понимания и качественного усвоения рекомендуется следующая последовательность действий.

1. После каждой лекции студентам рекомендуется подробно разобрать прочитанный теоретический материал, выучить все определения, разобрать примеры, решенные на лекции. Перед следующей лекций обязательно повторить материал предыдущей лекции.

2. Перед лабораторным занятием обязательно повторить лекционный материал. После лабораторного занятия еще раз разобрать решенные на этом занятии примеры, после чего приступить к выполнению домашнего задания. Если при решении примеров, заданных на дом, возникнут вопросы, обязательно задать на следующем лабораторном занятии или в присутственный час преподавателю.

3. При подготовке к лабораторным занятиям повторить основные понятия по темам, изучить примеры. Решая задачи, предварительно понять, какой теоретический материал нужно использовать. Наметить план решения, попробовать на его основе решить практические задачи.

4. Выбрать время для работы с литературой по дисциплине в библиотеке: каждый вторник с 15:00 до 18:00

Самостоятельная учебная деятельность студентов по дисциплине «**Технология и методы программирования**» предполагает изучение рекомендуемой преподавателем литературы по вопросам лекционных и практических занятий (приведены выше), самостоятельное освоение понятийного аппарата и подготовку к текущим аттестациям (**выполнению практических заданий**) (примеры см. ниже).

Вопросы лекционных и практических занятий обсуждаются на занятиях в виде устного опроса – индивидуального и фронтального. При подготовке к лекционным и практическим занятиям, обучающимся важно помнить, что их задача, отвечая на основные вопросы плана занятия и дополнительные вопросы преподавателя, показать свои знания и кругозор, умение логически построить ответ, владение математическим аппаратом и иные коммуникативные навыки, умение отстаивать свою профессиональную позицию. В ходе устного опроса выявляются детали, которые по каким-то причинам оказались недостаточно осмысленными студентами в ходе учебных занятий. Тем самым опрос выполняет важнейшие обучающую, развивающую и корректирующую функции, позволяет студентам учесть недоработки и избежать их при подготовке к промежуточным аттестациям.

Все выполняемые студентами самостоятельно задания (выполнение контрольной работы и практических заданий) подлежат последующей проверке преподавателем. Результаты текущих аттестаций учитываются преподавателем при проведении промежуточной аттестации (**3 семестр – зачет, 4 семестр - экзамен**).

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

а) основная литература:

№ п/п	Источник
1.	**Рудалев, Валерий Геннадьевич**. Технология визуального программирования : учебное пособие для вузов / В.Г. Рудалев ; Воронеж. гос. ун-т .— Воронеж
2.	**Иванова, Галина Сергеевна**. Технология программирования : Учебник для студ. вузов, обуч. по направлению "Информатика и вычислительная техника", специальностям: "Вычислительные машины, комплексы, системы и сети", "Автоматизированные системы обработки информации и управления", "Программное обеспечение вычислительной техники и информационных систем" / Г.С. Иванова .— 2-е изд., стер. — М. : Изд-во МГТУ имени Н.Э. Баумана, 2003 .— 319 с.

б) дополнительная литература:

№ п/п	Источник
3.	***Камаев, Валерий Анатольевич**. Технологии программирования : учебник для вузов по направлению подгот. специалистов"Информатика и вычисл. техника" / В. А. Камаев, В. В. Костерин .— М. : Высш. шк., 2005 .— 358, [1] с. : ил. — Библиогр.: с. 354-355*
4.	***Кулямин, Виктор Вячеславович**. Технологии программирования. Компонентный подход : учебное пособие / В.В. Кулямин .— М. : Бином.Лаборатория знаний : Интернет-Университет Информационных Технологий, 2007 .— 463 с.*

в) информационные электронно-образовательные ресурсы**:**

№ п/п	Источник
6.	*Лекции по технологии программирования http://sp.cmc.msu.ru/info/3/techprog.htm*
7.	***Дейтел, Х.М.** Технологии программирования на Java 2 / Х.М. Дейтел, П.Д. Дейтел, С.И. Сантри .— М. : Бином, 2003*
8.	*http://eqworld.ipmnet.ru – интернет-портал, посвященный уравнениям и методам их решений*
9.	*http://www.lib.vsu.ru - электронный каталог ЗНБ ВГУ*
10.	*http://www.kuchp.ru – электронный сайт кафедры уравнений в частных производных и теории вероятностей, на котором размещены методические издания*

16. Перечень учебно-методического обеспечения для самостоятельной работы

№ п/п	Источник
1	Груздев Д.В. «Технология программирования»
2	Положение об организации самостоятельной работы обучающихся в Воронежском государственном университете

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

Дисциплина может реализовываться с применением электронного обучения и дистанционных образовательных технологий. При проведении занятий в дистанционной форме используются технические и информационные ресурсы Образовательного портала "Электронный университет ВГУ" (https://edu.vsu.ru), базирующегося на системе дистанционного обучения Moodle, развернутой в университете, а также другие доступные ресурсы в сети Интернет.

Перечень необходимого программного обеспечения : операционная система Windows , Linux, браузер Mozilla Firefox, Opera или Internet Expolorer, Denwer, PHP, MySQL, Экран, ноутбук.

18. Материально-техническое обеспечение дисциплины:

Проектор, ноутбук, экран. Для проведения лекционных и практических занятий используются аудитории, соответствующие действующим санитарно-техническим нормам и противопожарным правилам.

Для самостоятельной работы используется класс с компьютерной техникой, оснащенный необходимым программным обеспечением, электронными учебными пособиями и законодательно - правовой и нормативной поисковой системой, имеющий выход в глобальную сеть.

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	Компетенция(и)	Индикатор(ы) достижения компетенции	Оценочные средства
1	Понятие об архитектуре ЭВМ	ОПК-7	ОПК-7.2	Домашнее задание, контрольная работа
2	Операционные системы	ОПК-7	ОПК-7.2	Домашнее задание, контрольная работа
3	Введение в C++	ОПК-7	ОПК-7.2	Домашнее задание, контрольная работа
4	Типы данных и выражения	ОПК-7	ОПК-7.2	Домашнее задание, контрольная работа
5	Управляющие структуры	ОПК-7	ОПК-7.2	Домашнее задание, контрольная работа
6	Массивы и указатели	ОПК-7	ОПК-7.2	Домашнее задание, контрольная работа
7	Функции	ОПК-7	ОПК-7.2	Домашнее задание, контрольная работа
8	Сортировки	ОПК-7	ОПК-7.2	Домашнее задание, контрольная работа
9	Файлы и потоки ввода-вывода	ОПК-7	ОПК-7.2	Домашнее задание, контрольная работа
10	Динамические структуры	ОПК-7	ОПК-7.2	Домашнее задание, контрольная работа
11	Основные принципы ООП	ОПК-7	ОПК-7.2	Домашнее задание, контрольная работа
12	Классы и объекты	ОПК-7	ОПК-7.2	Домашнее задание, контрольная работа
13	Наследование	ОПК-7	ОПК-7.2	Домашнее задание, контрольная работа
14	Обработка ошибок	ОПК-7	ОПК-7.2	Домашнее задание, контрольная работа
15	Эволюция вычислительных систем. Архитектура вычислительных систем	ОПК-7	ОПК-7.2	Домашнее задание, контрольная работа
16	Многоуровневая модель OSI. Стан-	ОПК-7	ОПК-7.2	Домашнее задание, контрольная работа

	дарты и стеки протоколов.			
17	Топологии построения сетей. Методы доступа. Адресация, IP адреса V4, V6	ОПК-7	ОПК-7.2	Домашнее задание, контрольная работа
18	Физическая среда передачи данных. Сетевое оборудование.	ОПК-7	ОПК-7.2	Домашнее задание, контрольная работа
19	Сетевые операционные системы.	ОПК-7	ОПК-7.2	Домашнее задание, контрольная работа
20	Стандартные локальные сети.	ОПК-7	ОПК-7.2	Домашнее задание, контрольная работа
	Промежуточная аттестация форма контроля – зачёт и экзамен			Перечень вопросов Практическое задание

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: домашнее задание, контрольная работа, лабораторная работа.

Примеры заданий на контрольной работе:

1. Решить задачу, используя указатели. Дано целое число n (вводится с клавиатуры). Если оно кратно 10, разделить его на 10.

Задачи 2-4 решить, используя функции

2. Даны квадрат со стороной a и прямоугольник со сторонами b и c. Написав функцию, вычисляющую площадь прямоугольника по двум заданным сторонам, определить, площадь какой из фигур больше. (значения a, b и c вводятся с клавиатуры).

3. Дан символ (вводится с клавиатуры). Написать void-функцию, заменяющую этот символ на * в случае, если он является буквой.

4. Дан одномерный массив, все элементы которого различны. Поменять местами максимальный и минимальный элементы. (Написать функции поиска максимального и минимального элементов).

6. Дан целочисленный массив (считывается из текстового файла), среди элементов которого есть нулевые. Записать в другой текстовый файл их индексы.

7. Дан текст (считывается из текстового файла). Подсчитать количество букв в первом слове. Результат вывести на экран.

При оценивании используется следующая шкала:

5 баллов ставится, если обучающийся демонстрирует полное соответствие знаний, умений, навыков приведенным в таблицах показателям, свободно оперирует приобретенными знаниями, умениями, применяет их при решении практических задач;

4 балла ставится, если обучающийся демонстрирует соответствие знаний, умений, навыков приведенным в таблицах показателям, но допускает незначительные ошибки, неточности, испытывает затруднения при решении практических задач;

3 балла ставится, если обучающийся демонстрирует неполное соответствие знаний, умений, навыков приведенным в таблицах показателям, допускает значительные ошибки при решении практических задач;

2 балла ставится, если обучающийся демонстрирует явное несоответствие знаний, умений, навыков приведенным в таблицах показателям.

20.2 Промежуточная аттестация Промежуточная аттестация по дисциплине осуществляется с помощью следующих оценочных средств:

Собеседование по билетам к зачету

Собеседование по билетам к экзамену

Владение понятийным аппаратом данной области науки (теоретическими основами дисциплины), способность иллюстрировать ответ примерами, фактами, применять теоретические знания для решения практических задач в области информатики

Промежуточная аттестация, как правило, осуществляется в конце семестра. Результаты текущей аттестации обучающегося по решению кафедры могут быть учтены при проведении промежуточной аттестации. При несогласии студента, ему дается возможность пройти промежуточную аттестацию (без учета его текущих аттестаций) на общих основаниях.

При проведении зачета учитываются результаты контрольных работ.

Для оценивания результатов обучения на зачете используется шкала: «зачтено», «не зачтено».

Критерии оценивания компетенций	Уровень сформированности компетенций	Шкала оценок
Обучающийся в полной мере владеет понятийным аппаратом данной области науки (теоретическими основами дисциплины), способен иллюстрировать ответ примерами, фактами применять теоретические знания для решения практических задач	*Повышенный уровень*	зачтено
Обучающийся владеет понятийным аппаратом данной области науки (теоретическими основами дисциплины), способен иллюстрировать ответ примерами, фактами, допускает ошибки при решении практических задачи или способен применять теоретические знания для решения практических задач в области информатики, но допускает неточности при применении понятийного аппарата данной области науки, но отвечает на дополнительные вопросы	*Базовый уровень*	зачтено
Обучающийся владеет частично теоретическими основами дисциплины, фрагментарно способен иллюстрировать ответ примерами, фактами, не отвечает на дополнительные вопросы *Не умеет применять теоретические знания для решения практических задач*	*Пороговый уровень*	не зачтено
Ответ на контрольно-измерительный материал не соответствует любым трем(четырем) из перечисленных показателей. Обучающийся демонстрирует отрывочные, фрагментарные знания, допускает грубые ошибки	–	не зачтено

Перечень вопросов к экзамену:

1. Языки программирования. Классификация языков программирования.
2. Машинные команды. Языки программирования низкого уровня.
3. Языки программирования высокого уровня. Алгоритмические языки.
4. Объектно-ориентированное и обобщенное программирование.
5. Языки программирования С и С++ (создание, предназначение, возможности).
6. Инструментарий создания приложений (компиляторы, интерпретаторы, архиваторы, компоновщики, отладчики).
7. Интегрированная среда разработки приложений.
8. Препроцессор и макрообработка языка С++. Директивы препроцессора.
9. Приложение Microsoft Visual Studio 2010.
10. Этапы решения задач на компьютере.
11. Базовые средства языка С++ (состав языка, алфавит языка, идентификаторы, ключевые слова).
12. Вывод сообщений на экран. Комментарии. Примеры.
13. Данные (константы, переменные, типы данных).
14. Целые числа и их двоичное представление.
15. Вещественные числа с фиксированной и плавающей точкой. Вещественные числа однократной и двукратной точности.
16. Операторы объявления, присваивания и ввода данных с клавиатуры. Составные операторы присваивания. Примеры.
17. Арифметические операции. Примеры.
18. Операции отношений и логические операции. Примеры.
19. Оператор выбора if. Примеры.
20. Оператор «знак ?». Примеры.
21. Оператор выбора switch. Примеры.
22. Вложенные инструкции if и switch. "Лестничная" конструкция if-eise-if
23. Итерационный оператор for. Использование нескольких управляющих переменных. Отсутствие элементов заголовка и циклы без тела. Примеры.
24. Итерационный оператор whlie. Примеры.
25. Итерационный оператор do-whlie. Примеры.
26. Инструкции break и continue. Оператор goto. Примеры.
27. Структурированный тип данных – массив (одномерный, двумерный, многомерный). Примеры.
28. Структурированный тип данных – строка. Ввод строк с клавиатуры. Примеры.
29. Библиотечные функции обработки строк и их использование. Примеры.
30. Инициализация массивов (безразмерных массивов). Массивы строк и их инициализация. Примеры.
31. Указатели. Операторы, используемые с указателями. Примеры.
32. Базовый тип указателя. Присваивание значений и проведение вычислений с помощью указателей. Примеры.
33. Арифметические и логические операции над указателями. Примеры.
34. Указатели массивов. Примеры.
35. Индексирование указателя. Строковые константы и указатели. Примеры.
36. Массивы указателей. Нулевые указатели. Многоуровневая непрямая адресация. Примеры.

37. Функции языка C++. Общий формат функций. Создание void-функции и использование ее аргументов. Примеры.

38. Создание функции, возвращающей значение, и использование ее в выражениях. Использование инструкции return. Примеры.

39. Правила действия областей видимости функций. Локальная область видимости. Примеры.

40. Объявление локальных переменных внутри блока. Сокрытие имен. Примеры.

41. Глобальная область видимости. Примеры.

42. Передача функции указателя. Примеры.

43. Передача функции массива. Примеры.

44. Передача функции строк. Примеры.

45. Возвращение функцией указателя. Примеры.

46. Прототипы функций. Прототипы функций, содержащиеся в стандартных заголовках. Примеры.

47. Рекурсия. Функция main(). Примеры.

48. Сортировки массивов методом выбора и методом вставки. Примеры.

49. Сортировки массивов методом обмена (пузырька) и челночным методом. Примеры.

50. Сортировка массивов методом Шелла и быстрая сортировка (методом Хоара). Примеры.

51. Турнирная и пирамидальная сортировки массивов. Примеры.

52. Способы передачи аргументов функции. Примеры.

53. Ссылочные параметры функций. Примеры.

54. Перегрузка функций. Примеры.

55. Аргументы, передаваемые функции по умолчанию. Примеры.

56. Модификаторы типов данных. Спецификаторы типа const и volatile. Примеры.

57. Статические переменные (локальные и глобальные). Регистровые переменные. Примеры.

58. Перечисления. Примеры.

59. Ключевое слово typedef. Примеры.

60. Поразрядные операторы. Примеры.

61. Оператор “знак запятая”. Примеры.

62. Одновременное присваивание и составные операторы присваивания. Примеры.

63. Ключевое слово sizeof. Примеры.

64. Сводная таблица приоритетов C++-операторов. Примеры.

65. Системы свода-вывода. Потоки. Встроенные C++-потоки.

66. Текстовые файлы и «двоичные» файлы.

67. Работа с файлами. Текстовые потоки. Настройка открытия файла. Примеры.

68. Запись текстовой информации в дисковый файл. Примеры.

69. Чтение текстовой информации из дискового файла. Примеры.

70. Неявные и явные операции приведения типов. Примеры.

71. Структура. Примеры.

При сдаче экзамена
оценка «отлично» - 5 баллов

оценка «хорошо» - 4 балла
оценка «удовлетворительно» - 3 балла
оценка «неудовлетворительно» - 2 балла.

Критерии оценивания компетенций (экзамен)	Уровень сформированности компетенций	Шкала оценок
Обучающийся в полной мере владеет понятийным аппаратом в области программирования и технологии работы на ЭВМ, способен иллюстрировать ответ примерами, фактами, применять теоретические знания для решения практических задач программирования	*Повышенный уровень*	*Отлично*
У обучающегося сформированы знания, умения и навыки программирования и технологии работы на ЭВМ; он способен иллюстрировать ответ примерами, фактами, применять теоретические знания для решения практических задач; но допускает отдельные несущественные пробелы в своих знаниях, допускает ошибки при выполнении практических задач.	*Базовый уровень*	*Хорошо*
У обучающегося сформированы неполные знания, умения и навыки; он допускает отдельные существенные пробелы в своих знаниях, допускает существенные ошибки при выполнении практических задач.	*Пороговый уровень*	*Удовлетворительно*
Сформированы лишь фрагментарные знания, умения и навыки или знания, умения и навыки отсутствуют	–	*Неудовлетворительно*

20.3 Фонд оценочных средств сформированности компетенций студентов, рекомендуемый для проведения диагностических работ

Закрытый вопрос

1. Что означает директива #include <iostream> в программе:

```
#include <iostream>
using namespace std;
int main()
{
  cout << "Добро пожаловать в C++!\n";
  return 0;
}
```

1. сообщение, что программа успешно окончена
2. сообщение компилятору для включения стандартных возможностей потока ввода и вывода
3. сообщение для использовании стандартного пространства имен
4. сообщение компилятору для включения возможностей для работы с переменными

Правильный ответ: 2
Решение: #include <iostream>
сообщает компилятору, чтобы он включил стандартные возможности потока ввода и вывода, находящиеся в файле iostream

Открытый вопрос

2. Когда обратный слэш (\) встречается в цепочке символов, следующий символ комбинируется с обратным слэшем и формирует *управляющую последовательность (escape-последовательность).* Напишите

управляющую последовательность, которая означает переход на *новую строку.* Она вызывает перемещение курсора (т.е. индикатора текущей позиции на экране) к началу следующей строки на экране.

Правильный ответ: \n

Решение: **\n -** Новая строка. Позиционирование курсора к началу следующей строки.

Закрытый вопрос

3. Что означает операция cin >> integer;
    1. оператор использует объект входного потока и операцию взять из потока, чтобы получить от пользователя значение.
    2. оператор использует объект выходного потока и операцию вывести значение переменной на экран.
    3. описание целой переменной и присвоение ей значения, введенного с клавиатуры
    4. выделение памяти для переменной integer целого типа.

Правильный ответ: 1

Решение: Оператор cin >> integerl; использует объект входного потока cin и операцию взять из потока », чтобы получить от пользователя значение.

Открытый вопрос

4. Напишите оператор описания целой переменной **a**, которой присваивается значение **10**

Правильный ответ: int a=10;

Правильный ответ: int a; a=10;

Решение: int a – описывается целочисленная переменная. a=10 – присвоение целой переменной значения 10

Открытый вопрос

5. В программе вычисляется значение переменной b. Напишите, чему будет равно ее значение и что выведется на экране.

```
#include <iostream>
using namespace std;
int main()
{
        int c=5;
        int b;
        b=c++;
        cout <<b;
        return 0;
}
```

Правильный ответ: 5

Решение: Сначала выполнится операция присваивания, а затем только переменная **c** увеличится на единицу. Следовательно, значение переменной **b** будет равно 5.

Открытый вопрос

6. Для переменной **a** напишите префиксную форму инкремента.

Правильный ответ: ++a;

Правильный ответ: ++a

Решение: ++a - префиксная форма инкремента. Величина **a** увеличивается на 1 и это новое значение **a** используется в выражении, в котором оно встретилось

Закрытый вопрос

7. Что делает следующая программа?

```
int main()
{
        int sum, i, n, num;
        sum=0;
        n=10;
        for (i=0; i<n; i++)
        {
                cin>>num;
                if (num>=0)
                        continue;
                sum+=num;
        }
        cout<<sum;
        return 0;
}
```

1. Программа находит сумму 10 положительных чисел
2. Программа находит сумму 10 отрицательных чисел
3. Программа находит сумму отрицательных чисел среди 10 введенных
4. Программа находит сумму положительных чисел среди 10 введенных

Правильный ответ: 3

Решение: Данная программа вычислить сумму только отрицательных чисел из 10 введенных.

Закрытый вопрос

8. Что делает следующая программа?

```
int main()
{
        int a=10, b=5;
        char znak='-';
        switch(znak)
        {
                Case '-': cout << (a-b); break;
                Case '+': cout << (a+b); break;
                Case '*': cout << (a*b); break;
                Case '/': cout << (a/b); break;
        }
        return 0;
}
```

1. Программа находит 10-5=5
2. Программа находит 10+5=15
3. Программа находит 10*5=50
4. Программа находит 10/5=2

Правильный ответ: 1

Решение: Данная программа вычислить разность двух переменных, так как znak='-'.

Закрытый вопрос

9. В чем будет ошибка данного оператора цикла?

```
for (int i=0; i<5, i++)
```

1. Нельзя использовать описание переменной i внутри круглый скобок
2. Нельзя использовать операцию инкремента внутри круглый скобок

3. Вместо «запятой» должен стоять знак «точка с занятой»
4. Скобки должны быть не круглыми, а фигурными

Правильный ответ: 3
Решение: Любая из частей оператора for может быть опущена, но точки с запятой должны остаться на своих местах for (; ;)

Открытый вопрос

10. Напишите команду цикла, в которой переменна i должна изменяться от 10 до 20 с шагом 2

Правильный ответ: for (int i=10; i<=20, i=i+2)
Правильный ответ: for (int i=10; i<21, i=i+2)
Правильный ответ: for (int i=10; i<=20, i+=2)
Правильный ответ: for (int i=10; i<21, i+=2)

Закрытый вопрос

11. Что означает следующий оператор?

double months[12]

1. Описание целочисленного массива, состоящего из 12 элементов
2. Описание целочисленной переменной months и присвоение ей значения 12
3. Описание вещественной переменной months и присвоение ей значения 12
4. Описание вещественного массива, состоящего из 12 элементов

Правильный ответ: 4
Решение: double months[12] создается массив с именем months, который содержит 12 элементов, каждый из которых может хранить значение типа double. Каждый элемент массива можно обрабатывать как обычную переменную.

Открытый вопрос

12. Дано описание массива int hand[4]. Напишите команду присвоения четвертому элементу массива значения 10.

Правильный ответ: hand[3] = 10;
Правильный ответ: hand[3] = 10
Решение: Нумерация элементов массива начинается с нуля. Поэтому четвертый элемент массива будет иметь индекс, равный трем.

Закрытый вопрос

13. Что означает следующий оператор

int mas[4][5];

1. Описание целочисленного массива, состоящего из 4 элементов
2. Описание целочисленной переменной mas и присвоение ей значения 4
3. Описание целочисленного двумерного массива
4. Описание вещественного двумерного массива, у которого 4 строки и 5 столбцов

Правильный ответ: 3
Решение: int mas[4][5] означает, что mas является массивом из 4 элементов, причем каждый элемент данного массива представляет собой массив из 5 целочисленных значений. То есть получается, что это описание двумерного целочисленного массива.

Открытый вопрос

14. Дан цикл for (int i=1; i<=10, i++). Напишите команду в этом цикле для заполнения массива mas числами от 1 до 10

Правильный ответ: mas[i]=i;
Правильный ответ: mas[i]=i

Закрытый вопрос

15. Что делает следующий цикл?

```
for (int row=0; row<4; row++)
{
        for (int col=0; col<5; col++)
                cout << mas[row][col] <<" ";
        cout <<"\n";
}
```

1. Выводит двумерный массив mas на экран в виде 4 строк и 5 столбцов
2. Выводит двумерный массив mas на экран в виде 5 строк и 4 столбцов
3. Выводит двумерный массив mas на экран в виде одной строки, состоящей из 20 элементов
4. Выводит двумерный массив mas на экран в виде одного столбца, состоящего из 20 элементов

Правильный ответ: 1

Открытый вопрос

16. Напишите команду условного оператора для проверки пятого (индекс равен 4) элемента массива mas на четность, и в этом случае присвоить ему значение 10, а в противном случае, присвоить значение 20.

Правильный ответ:
if (mas[4]%2==0) mas[5]=10 else mas[4]=20
Правильный ответ:
if (mas[4]%2==0) mas[5]=10
else mas[4]=20

Закрытый вопрос

17. Что делает следующая программа

```
for (i=0;i<n;i++)
{
        for (j=0;j<m;j++)
        {
                a[i][j]=1 + rand() % 10 ;
        }
}
```

1. Заполняет двумерный массив числами, введенными с клавиатуры
2. Заполняет одномерный массив случайными числами
3. Заполняет двумерный массив случайными числами от 0 до 9
4. Заполняет двумерный массив случайными числами от 1 до 10

Правильный ответ: 4
Решение: rand() % 10 - генерация случайного числа до 0 до 9. Следовательно, 1 + rand() % 10 - генерация случайного числа до 1 до 10

Открытый вопрос

18. Напишите команду заполнения двумерного массива mas случайными двузначными числами. mas[i][j]=…

Правильный ответ: mas[i][j]= 10 + rand() % 90
Правильный ответ: mas[i][j]= rand() % 90 + 10
Решение: rand() % 90 - генерация случайного числа до 0 до 89. Следовательно, 10 + rand() % 90 - генерация случайного числа до 10 до 99 (двузначные числа)

Закрытый вопрос

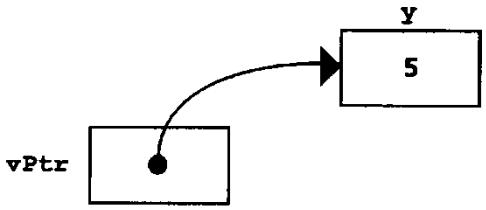
19.Даны описания переменных **int a = 5; int *b;**
Что означает оператор **b = &a?**

1. оператор присваивает адрес переменной **a** указателю **b**.
2. оператор присваивает значение переменной **a** указателю **b**.
3. оператор присваивает адрес переменной **b** указателю **a**.
4. оператор присваивает значение переменной **b** указателю **a**.

Правильный ответ: 1
Решение: **b = &a;** присваивает адрес переменной **a** указателю **b**. Говорят, что переменная b «указывает» на a.

Открытый вопрос

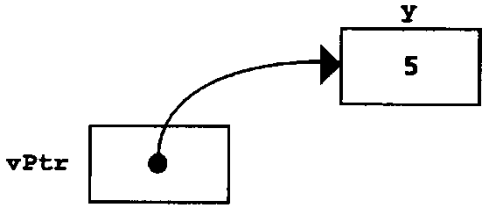
20.На рисунке изображено, что переменная **yPtr** «указывает» на **y.**
int y = 5; int *yPtr;



Напишите оператор такого присваивания
Правильный ответ: **yPtr = &y;**
Решение: **yPtr = &y;** присваивает адрес переменной y указателю yPtr. Говорят, что переменная yPtr «указывает» на y.

Закрытый вопрос

21.На рисунке изображено, что переменная **yPtr** «указывает» на **y**



Что будет выведено на экране?
cout « *yPtr « endl;

1. Значение переменной **y**, то есть 5.
2. Адрес переменной **y**.
3. Адрес переменной **yPtr**.
4. Будет выведена ошибка копилятора

Правильный ответ: 1
Решение: оператор **cout « *yPtr « endl;** печатает значение переменной y, а именно 5. Использование * указанным способом называется *разыменованием указателя.*

Открытый вопрос

22.Напишите команду объявления переменной **yPtr** двойным целочисленным указателем
Правильный ответ: **int **yPtr;**
Правильный ответ: **int **yPtr**

Закрытый вопрос

23.Описана функция нахождения факториала целого числа

```
void faktorial(int numb)
{
        int rezult = 1;
        for (int i = 1; i <= numb; i++)
        rezult *= i;
        cout << numb << "! = " << rezult << endl;
}
```

Выберите, какой функция является

1. Типизированной с параметрами
2. Нетипизированной с параметрами
3. Типизированной без параметров
4. Нетипизированной без параметров

Правильный ответ: 2

Решение: Тип данных **void** говорит о том, что данная функция не возвращает никаких значений. **void** никак по-другому не используется и нужен только для того, чтобы компилятор мог определить тип функции.

Открытый вопрос

24.Описана функция нахождения факториала целого числа

```
void faktorial(int numb)
{
        int rezult = 1;
        for (int i = 1; i <= numb; i++)
        rezult *= i;
        cout << numb << "! = " << rezult << endl;
}
```

Напишите команду присвоения целой переменной **a** значения суммы факториала 5 и факториала 6

Правильный ответ: int a=faktorial(5)+factorial(6)

Правильный ответ: a=faktorial(5)+factorial(6)

Закрытый вопрос

25.Что делает функция **atoi(str)?**

1. преобразует строку в целочисленное значение (типа *int*)
2. преобразует строку в длинное целое (типа *long*)
3. преобразует строку в значение с плавающей точкой (типа *double*)
4. выводит количество символов с строке

Правильный ответ: 1

Открытый вопрос

26.Даны переменные

```
double a;
char str=”3.14”;
```

Напишите команду присвоения переменной **a** числового значения, хранящегося в строке **str** (то есть a=3.14)

Правильный ответ: a=**atof(str)**

Решение: Функция **atof(str)** преобразует строку в значение с плавающей точкой (типа *double*), поэтому вещественной переменной a нужно присвоить значение функции.

Закрытый вопрос

27.Как называется данная функция

```
int factr(int n)
{
   int answer;
   if(n==1) return(1);
   answer = factr(n-1)*n;
   return(answer);
}
```

1. адресная
2. указательная
3. ссылочная
4. рекурсивная

Правильный ответ: 4

Решение: **Рекурсивная функция** — *это функция, которая вызывает сама себя.* Рекурсия, которую иногда называют *циклическим определением*, представляет собой процесс определения чего-либо на собственной основе.

Открытый вопрос

28.Дана структура и описана переменная inv_var.

```
struct inv_type
{
   char item[40]; // наименование товара
   double cost; // стоимость
   double retail; // розничная цена
   int on_hand; // имеющееся в наличии количество
   int lead_time; // число дней до пополнения запасов
};
inv_type inv_var;
```

Товар стал стоить 100 рублей в розницу. Напишите команду для этого действия

Правильный ответ: inv_var.retail=100

Решение: К отдельным членам структуры доступ осуществляется с помощью оператора *"точка"*.

Закрытый вопрос

29.Дана структура

```
struct inv_type
{
   char item[40]; // наименование товара
   double cost; // стоимость
   double retail; // розничная цена
   int on_hand; // имеющееся в наличии количество
   int lead_time; // число дней до пополнения запасов
};
```

Что означает следующее описание?

```
inv_type invtry[100];
```

1. Объявлена переменная типа *inv_type* и ей присвоено значение 100
2. Объявлен *100*-элементный массив структур типа *inv_type* и все поля стоимость заполнены числами 100
3. объявлен *100*-элементный массив структур типа *inv_type*
4. увеличение розничной цены на 100 рублей

Правильный ответ: 3

Открытый вопрос

30.Дана структура

```
struct inv_type
{
   char item[40]; // наименование товара
   double cost; // стоимость
   double retail; // розничная цена
   int on_hand; // имеющееся в наличии количество
   int lead_time; // число дней до пополнения запасов
};
inv_type invtry[100];
```

Напишите команду вывода на экран содержимое члена *on_hand* третьей (индекс=2) структуры

Правильный ответ: cout << invtry[2].on_hand;

Критерии и шкалы оценивания заданий ФОС:

1) Задания закрытого типа (выбор одного варианта ответа, верно/неверно):

- 1 балл – указан верный ответ;
- 0 баллов – указан неверный ответ.

2) Задания закрытого типа (множественный выбор):

- 2 балла – указаны все верные ответы;
- 0 баллов — указан хотя бы один неверный ответ.

3) Задания закрытого типа (на соответствие):

- 2 балла – все соответствия определены верно;
- 0 баллов – хотя бы одно сопоставление определено неверно.

4) Задания открытого типа (короткий текст):

- 2 балла – указан верный ответ;
- 0 баллов – указан неверный ответ.

5) Задания открытого типа (число):

- 2 балла – указан верный ответ;
- 0 баллов – указан неверный ответ.

Задания раздела 20.3 рекомендуются к использованию при проведении диагностических работ с целью оценки остаточных результатов освоения данной дисциплины (знаний, умений, навыков).