

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.  
подпись, расшифровка подписи  
25.05.2023г.

## **РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ**

Б1.О.12 Программирование для ЭВМ

- 1. Код и наименование направления подготовки:** 01.03.04 Прикладная математика
  - 2. Профиль подготовки:** Применение математических методов к решению инженерных и экономических задач
  - 3. Квалификация выпускника:** бакалавр
  - 4. Форма обучения:** очная
  - 5. Кафедра, отвечающая за реализацию дисциплины:** функционального анализа и операторных уравнений
  - 6. Составители программы:** Ушаков Сергей Николаевич, к.ф.-м.н.
  - 7. Рекомендована:** научно-методическим советом математического факультета, протокол от 25.05.2023, № 0500-06
  - 8. Учебный год:** 2023-2024, 2024-2025
- Семестр(ы):** первый-третий

## 9. Цели и задачи учебной дисциплины

*Целями освоения учебной дисциплины являются:*

- овладение базовыми навыками программирования, в том числе и объектно-ориентированного программирования, на языке C++, необходимыми для разработки прикладных программных проектов;

- подготовка в области применения современной вычислительной техники для решения практических задач обработки данных, математического моделирования, информатики, получение профессиональных умений и навыков (на уровне бакалавра), позволяющих выпускнику успешно работать в избранной сфере деятельности с применением современных компьютерных технологий.

*Задачи учебной дисциплины:*

- изучить основные элементы одного из самых распространенных языков программирования C++; основные приемы и алгоритмы программирования; основные численные методы решения задач;

- научить разрабатывать алгоритмы необходимые для решения математических, физических задач, разрабатывать алгоритмы, используя основные приемы программирования; проводить отладку, тестирование программы; проводить необходимые расчеты на ПК.

## 10. Место учебной дисциплины в структуре ООП:

Дисциплина Программирование для ЭВМ относится к обязательной части блока Б1.

Для изучения и освоения дисциплины нужны первоначальные знания из курсов математического анализа, алгебры, аналитической геометрии. Знания и умения, приобретенные студентами в результате изучения дисциплины, будут использоваться при изучении курсов численных методов, вычислительного практикума, при выполнении курсовых и дипломных работ, связанных с математическим моделированием и обработкой наборов данных.

Дисциплина "Программирование для ЭВМ" является предшествующей для изучения следующих дисциплин: «Технологии программирования», «Численные методы».

## 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-3	Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности	ОПК-3.1	Осуществляет поиск, сбор, хранение, обработку, представление информации при решении задач профессиональной деятельности	знать: принципы работы с поисковыми системами интернета и электронными ресурсами уметь: применять теоретические знания на практике владеть: навыками управления информацией
		ОПК-3.2	Подбирает и использует	знать: принципы работы с IDE на

			информационные технологии при решении задач профессиональной деятельности	примере Visual Studio 2015 уметь: настраивать Visual Studio 2015 для использования владеть: навыками программирования в Visual Studio 2015
ОПК-4	Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения	ОПК-4.1.	Использует основные принципы алгоритмизации задач в рамках профессиональной деятельности и разработки компьютерных программ	знать: алгоритмы ветвления, циклы, массивы, функции, принципы ООП. уметь: применять перечисленные в графе знать понятия для создания программ владеть: навыками создания программы на основе разработанного алгоритма
		ОПК-4.2.	Проводит тестирование и отладку компьютерных программ с целью апробации разработанных моделей и алгоритмов	знать: принципы работы с IDE на примере Visual Studio 2015 уметь: использовать механизмы отладки в Visual Studio 2015 владеть: навыками тестирования

**12. Объем дисциплины в зачетных единицах/час. — 10/360.**

**Форма промежуточной аттестации зачет, экзамен**

**13. Трудоемкость по видам учебной работы**

Вид учебной работы		Трудоемкость			
		Всего	По семестрам		
			1 семестр	2 семестр	3 семестр
Аудиторные занятия					
в том числе:	лекции	84	34	34	16
	практические				
	лабораторные	84	34	34	16
Самостоятельная работа		156	40	40	76
в том числе: курсовая работа (проект)					
Форма промежуточной аттестации (экзамен – __ час.)		36			36
Итого:		360	108	108	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	Введение в язык C++	Общая характеристика языка и история его создания. Технология разработки программ. Компиляция программ. Интегрированная среда разработки приложений.	
1.2	Типы данных и выражения	Алфавит, идентификаторы, операции, выражения, операторы, классификация типов данных, переменные и константы. Простейший ввод-вывод данных.	

1.3	Управляющие структуры	Организация алгоритмов ветвления и циклов, выбор циклов.	
1.4	Массивы и строки	Понятие массива. Одномерные, двумерные и многомерные массивы. Строки и некоторые библиотечные функции обработки строк. Инициализация массивов и строк.	
1.5	Указатели	Указатели и операторы, используемые с указателями. Указатели и массивы. Строковые константы и указатели. Массивы указателей. Многоуровневая непрямая адресация.	
1.6	Препроцессор и макрообработка языка C++	Понятия препроцессора и макрообработки языка C++. Директивы препроцессора языка C++.	
1.7	Функции	Общие сведения о функциях, функции с переменным числом параметров, рекурсивные и подставляемые функции, области действия переменных, массивы в качестве параметров функций. Передача аргументов по ссылке. Перегруженные функции. Значения параметров по умолчанию.	
1.8	О типах данных	Модификаторы типов данных. Типы const и volatile. Статические и регистровые переменные. Перечисления. Переименование типов.	
1.9	Поразрядные операторы	Поразрядные операторы И, ИЛИ, исключающее ИЛИ и НЕ. Поразрядные операторы сдвига.	
1.10	Операторы	Операторы “знак вопроса” и “знак запятой”. Составные операторы присваивания. Оператор определения размера переменной и типа. Сводная таблица приоритетов C++-операторов.	
1.11	Файлы и потоки ввода-вывода	Назначение файлов, потоки ввода-вывода, проверка ошибок выполнения файловых операций, символьный ввод-вывод, операторы ввода-вывода. Введение в двоичные операции.	
1.12	Динамические структуры	Стек, очередь, список, дерево, дерево поиска. Динамические массивы.	
1.13	Основные принципы ООП	Структурный подход в программировании, инкапсуляция, наследование, полиморфизм.	
1.14	Классы и объекты	Способы описания классов. Создание объектов. Обращение к атрибутам и методам объектов. Конструкторы и деструкторы классов. Перегрузка операций. Функции друга.	
1.15	Наследование классов	Полиморфное открытое наследование. Статическое и динамическое связывание. Виртуальные методы. Абстрактные классы. Закрытое и защищенное наследование.	
1.16	Шаблоны	Обобщенные функции. Явно заданная перегрузка обобщенной функции. Перегрузка шаблона функции. Ограничения при использовании обобщенных функций. Обобщенные классы. Создание обобщенного класса безопасного массива. Использование в обобщенных классах аргументов, не являющихся типами. Явно задаваемые специализации классов.	

1.17	Обработка ошибок	Попытка классификации ошибок. Сообщение об ошибке с помощью возвращаемого значения. Исключительные ситуации. Обработка исключительных ситуаций, операторы try и catch. Классы исключений.	
<b>2. Практические занятия</b>			
<b>3. Лабораторные занятия</b>			
3.1	Введение в язык C++	Технология разработки программ. Компиляция программ. Интегрированная среда разработки приложений.	
3.2	Типы данных и выражения	Алфавит, идентификаторы, операции, выражения, операторы, классификация типов данных, переменные и константы. Простейший ввод-вывод данных.	
3.3	Управляющие структуры	Организация алгоритмов ветвления и циклов, выбор циклов.	
3.4	Массивы и строки	Понятие массива. Одномерные, двумерные и многомерные массивы. Строки и некоторые библиотечные функции обработки строк. Инициализация массивов и строк.	
3.5	Указатели	Указатели и операторы, используемые с указателями. Указатели и массивы. Строковые константы и указатели. Массивы указателей. Многоуровневая непрямая адресация.	
3.6	Препроцессор и макрообработка языка C++	Понятия препроцессора и макрообработки языка C++. Директивы препроцессора языка C++.	
3.7	Функции	Общие сведения о функциях, функции с переменным числом параметров, рекурсивные и подставляемые функции, области действия переменных, массивы в качестве параметров функций. Передача аргументов по ссылке. Перегруженные функции. Значения параметров по умолчанию.	
3.8	О типах данных	Модификаторы типов данных. Типы const и volatile. Статические и регистровые переменные. Перечисления. Переименование типов.	
3.9	Поразрядные операторы	Поразрядные операторы И, ИЛИ, исключающее ИЛИ и НЕ. Поразрядные операторы сдвига.	
3.10	Операторы	Операторы “знак вопроса” и “знак запятой”. Составные операторы присваивания. Оператор определения размера переменной и типа. Сводная таблица приоритетов C++-операторов.	
3.11	Файлы и потоки ввода-вывода	Назначение файлов, потоки ввода-вывода, проверка ошибок выполнения файловых операций, символьный ввод-вывод, операторы ввода-вывода. Введение в двоичные операции.	
3.12	Динамические структуры	Стек, очередь, список, дерево, дерево поиска. Динамические массивы.	
3.13	Основные принципы ООП	Структурный подход в программировании, инкапсуляция, наследование, полиморфизм.	
3.14	Классы и объекты	Способы описания классов. Создание объектов. Обращение к атрибутам и методам объектов.	

		Конструкторы и деструкторы классов. Перегрузка операций. Функции друга.	
3.15	Наследование классов	Полиморфное открытое наследование. Статическое и динамическое связывание. Виртуальные методы. Абстрактные классы. Закрытое и защищенное наследование.	
3.16	Шаблоны	Обобщённые функции. Явно заданная перегрузка обобщенной функции. Перегрузка шаблона функции. Ограничения при использовании обобщенных функций. Обобщенные классы. Создание обобщенного класса безопасного массива. Использование в обобщенных классах аргументов, не являющихся типами. Явно задаваемые специализации классов.	
3.17	Обработка ошибок	Попытка классификации ошибок. Сообщение об ошибке с помощью возвращаемого значения. Исключительные ситуации. Обработка исключительных ситуаций, операторы try и catch. Классы исключений.	

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

№ п/п	Наименование темы (раздела) дисциплины	Виды занятий (количество часов)				Всего
		Лекции	Практические	Лабораторные	Самостоятельная работа	
1.1	Введение в язык C++	2		2	2	12
1.2	Типы данных и выражения	4		4	1	4
1.3	Управляющие структуры	8		8	1	5
1.4	Массивы и строки	6		10	4	18
1.5	Указатели	6		8	4	14
1.6	Препроцессор и макрообработка языка C++	2		1	1	4
1.7	Функции	10		2	1	7
1.8	О типах данных	2		1	1	6
1.9	Поразрядные операторы	2		1	1	8
1.10	Операторы	2		1	1	6
1.11	Файлы и потоки ввода-вывода	6		17	5	26
1.12	Динамические структуры	4		6	2	10
1.13	Основные принципы ООП	2		2	1	3
1.14	Классы и объекты	8		2	1	7
1.15	Наследование классов	4		4	2	8
1.16	Шаблоны	2		2	2	10

1.17	Обработка ошибок	2		2	2	10
	Итого:	84		84	156	324

**14. Методические указания для обучающихся по освоению дисциплины:** Аудиторные занятия, лекции и лабораторные занятия, предполагают самостоятельную работу студентов по данному курсу. Ряд тем выносятся для самостоятельного изучения, предлагаются темы для создания докладов с презентациями. Предусмотрены домашние задания и оформление отчетов выполнения лабораторных заданий, а также дополнительные задания для сильных студентов.

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

а) основная литература:

№ п/п	Источник
1	<i>Шилдт, Герберт. С++ : базовый курс / Герберт Шилдт ; [пер. с англ. и ред. Н.М. Ручко] .— 3-е изд. — Москва ; Санкт-Петербург ; Киев : Вильямс, 2015 .— 620 с.</i>
2	<i>Стивен Прата Язык программирования С++. Лекции и упражнения, Вильямс – 2012, 1248 с.</i>
3	<i>Павловская, Татьяна Александровна. С/С++. Программирование на языке высокого уровня : учебник для студ. вузов, обуч. по направлению "Информатика и вычислительная техника" / Т.А. Павловская .— СПб. [и др.] : Питер, 2008 .— 460 с.</i>

б) дополнительная литература:

№ п/п	Источник
1	<i>Лафоре Р., Объектно-ориентированное программирование в С++. Классика Computer Science. Питер - 2011, 928 стр.</i>
2	<i>Страуструп, Бьерн. Язык программирования С++. Специальное издание = The С++ programming language. Special edition. / Бьерн Страуструп ; пер. с англ. под ред. Н.Н. Мартынова .— Москва : Бином, 2015 .— 1135 с.</i>

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

№ п/п	Ресурс
1	lib.vsu.ru
2	Электронно-библиотечная система "Лань"
3	Электронно-библиотечная система "Университетская библиотека online"
4	Образовательный портал "Электронный университет ВГУ"

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

№ п/п	Источник

**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.

**18. Материально-техническое обеспечение дисциплины:**

Компьютеры, с установленным программным обеспечением: Microsoft Visual Studio 2015, LibreOffice.

Для проведения лекционных и лабораторных занятий используются аудитории, соответствующие действующим санитарно-техническим нормам и противопожарным правилам.

Для проведения лабораторных занятий и 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 (модуля)	Компетенция(и)	Индикатор(ы) достижения компетенции	Оценочные средства
1	Введение в язык C++	ОПК-3, ОПК-4	ОПК-3.1,ОПК-3.2, ОПК-4.1,ОПК-4.2	Домашнее задание, контрольная работа
2	Типы данных и выражения	ОПК-3, ОПК-4	ОПК-3.1,ОПК-3.2, ОПК-4.1,ОПК-4.2	Домашнее задание, контрольная работа
3	Управляющие структуры	ОПК-3, ОПК-4	ОПК-3.1,ОПК-3.2, ОПК-4.1,ОПК-4.2	Домашнее задание, контрольная работа
4	Массивы и строки	ОПК-3, ОПК-4	ОПК-3.1,ОПК-3.2, ОПК-4.1,ОПК-4.2	Домашнее задание, контрольная работа
5	Указатели	ОПК-3, ОПК-4	ОПК-3.1,ОПК-3.2, ОПК-4.1,ОПК-4.2	Домашнее задание, контрольная работа
6	Препроцессор и макрообработка языка C++	ОПК-3, ОПК-4	ОПК-3.1,ОПК-3.2, ОПК-4.1,ОПК-4.2	Домашнее задание, контрольная работа
7	Функции	ОПК-3, ОПК-4	ОПК-3.1,ОПК-3.2, ОПК-4.1,ОПК-4.2	Домашнее задание, контрольная работа
8	О типах данных	ОПК-3, ОПК-4	ОПК-3.1,ОПК-3.2, ОПК-4.1,ОПК-4.2	Домашнее задание, контрольная работа
9	Поразрядные операторы	ОПК-3, ОПК-4	ОПК-3.1,ОПК-3.2, ОПК-4.1,ОПК-4.2	Домашнее задание, контрольная работа
10	Операторы	ОПК-3, ОПК-4	ОПК-3.1,ОПК-3.2, ОПК-4.1,ОПК-4.2	Домашнее задание, контрольная работа
11	Файлы и потоки ввода-вывода	ОПК-3, ОПК-4	ОПК-3.1,ОПК-3.2, ОПК-4.1,ОПК-4.2	Домашнее задание, контрольная работа
12	Динамические структуры	ОПК-3, ОПК-4	ОПК-3.1,ОПК-3.2, ОПК-4.1,ОПК-4.2	Домашнее задание, контрольная работа
13	Основные принципы ООП	ОПК-3, ОПК-4	ОПК-3.1,ОПК-3.2, ОПК-4.1,ОПК-4.2	Домашнее задание, контрольная работа
14	Классы и объекты	ОПК-3, ОПК-4	ОПК-3.1,ОПК-3.2, ОПК-4.1,ОПК-4.2	Домашнее задание, контрольная работа
15	Наследование классов	ОПК-3, ОПК-4	ОПК-3.1,ОПК-3.2, ОПК-	Домашнее задание, контрольная работа



№ п/п	Наименование раздела дисциплины (модуля)	Компетенция(и)	Индикатор(ы) достижения компетенции	Оценочные средства
			4.1,ОПК-4.2	
16	Шаблоны	ОПК-3, ОПК-4	ОПК-3.1,ОПК-3.2, ОПК-4.1,ОПК-4.2	Домашнее задание, контрольная работа
17	Обработка ошибок	ОПК-3, ОПК-4	ОПК-3.1,ОПК-3.2, ОПК-4.1,ОПК-4.2	Домашнее задание, контрольная работа
Промежуточная аттестация форма контроля – зачет, 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: индивидуальные задания, контрольные работы.

#### Перечень индивидуальных заданий №1 (1 семестр)

1. Ввести три числа. Если они могут быть длинами сторон прямоугольного треугольника, вывести их в порядке возрастания, вычислить площадь полученного треугольника.
2. Вычислить сумму всех чисел Фибоначчи, которые меньше 1000.
3. Вычислить длину кривой, соответствующей функции  $y = f(x)$  на отрезке  $[a, b]$ , приближенно заменив кривую ломаной, полученной в результате разбиения отрезка  $[a, b]$  на  $n$  равных частей.
4. Поменяйте местами первую и последнюю цифры целого числа.
5. Переменной  $t$  присвоить значение *true*, если в массиве нет нулевых элементов и при этом положительные элементы чередуются с отрицательными и значение *false* в противном случае.

#### Перечень индивидуальных заданий №2 (2 семестр)

1. Дана строка. Напечатать текст, составленный из последних литер всех слов.\  
Найти максимум из чисел, сумма цифр которых больше 20.  
Указание: Написать и использовать логическую функцию, проверяющую число на удовлетворение поставленному условию.  
Указание: Не пользоваться константами типа `maxint` и `minint`.
2. В матрице  $A(n \times n)$  определить количество строк, элементы которой образуют арифметическую прогрессию. Использовать подпрограмму проверки строки.
3. При некоторых заданных  $x$ ,  $N$  и  $E$ , определяемых вводом, вычислить:
  - a) сумму  $N$  слагаемых заданного вида;
  - b) сумму тех слагаемых, которые по абсолютной величине больше  $E$ .

Для случая  $b$  выполнить суммирование для двух значений  $E$ , отличающихся на порядок, и при этом определить количество слагаемых, включенных в сумму. Сравнить результаты с точным значением функции, для которой данная сумма определяет приближенное значение при  $x$ , лежащем в интервале  $(-R, R)$ .

$$\arctg(x) = x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \frac{x^9}{9} - \dots \quad (R=1).$$

4.

Описать структуру с именем TRAIN, содержащую следующие поля: название пункта назначения; номер поезда; время отправления. Написать программу, выполняющую следующие действия:

1. ввод с клавиатуры данных в массив, состоящий из шести элементов типа TRAIN; записи должны быть упорядочены по времени отправления поезда;
2. вывод на экран информации о поездах, направляющихся в пункт, название которого введено с клавиатуры;
3. если таких поездов нет, выдать на дисплей соответствующее сообщение.
4. Написать программу, которая считывает текст из файла и определяет, сколько в нем слов, состоящих из не более чем четырех буквами.
5. Написать программу, которая считывает текст из файла и определяет, сколько в нем слов, состоящих из не более чем четырех буквами.

### **Перечень индивидуальных заданий №3 (3 семестр)**

1. Создайте класс для представления книги. Данные-члены должны включать автора, издательство, год, число страниц, название. Функции- члены должны позволять следующее:

- создание объекта и его инициализация;
- отображение автора, названия и число страниц;
- изменение числа страниц (число страниц передаётся в аргументе);

2. В задании обязательно должны быть реализованы следующие методы:

- метод инициализации Init;
- ввод с клавиатуры Read;
- вывод на экран Display;
- конструктор по умолчанию с сообщением о работе
- деструктор с сообщением о работе
- функции для вывода закрытых полей при помощи this
- перегруженная операция <<
- Сделать перегрузку операторов, соответствующих методам, перечисленным в задании

Создайте класс Polynom для работы с многочленами. Обязательно должны быть реализованы операции сложения, вычитания и умножения многочленов, а также дифференцирования.

3. Создайте два класса: товар с полями стоимость, название, дата производства и класс продавец с полями ФИО, номер телефона. От этих двух базовых классов будет унаследован класс покупка с новым полем дата продажи. Классы должны содержать функции для ввода/вывода информации и её изменения, конструкторы с параметрами и без. Напишите функцию main() для тестирования возможностей.

4. Создать абстрактный базовый класс Body (тело) с виртуальными функциями вычисления площади, поверхности и объема. Создать производные классы: Ball (шар), Cone (конус) со своими функциями площади, поверхности и объема. В основной программе создать интерфейс для демонстрации возможностей классов.

5. Создать программу для сортировки массива различных типов данных (использовать шаблоны функций). В интерфейсе программы должен быть предусмотрен выбор количества элементов для сортировки, ввод с клавиатуры, выбор типа данных, вывод отсортированного массива. Пузырьковая сортировка. Типы данных: double, char.

6. Создать программу сложения и вычитания двух целых чисел с обработкой исключения, возникающего при ошибке переполнения.

## Перечень заданий для контрольных работ

### Форма контрольно-измерительного материала № 1

УТВЕРЖДАЮ  
Заведующий кафедрой функционального анализа и операторных уравнений

\_\_\_\_\_  
подпись, расшифровка подписи

\_\_ . \_\_ .20\_\_

Направление подготовки / специальность 01.03.04 Прикладная математика  
шифр, наименование

Дисциплина Б1.О.12 Программирование для ЭВМ

Форма обучения очная  
очное, очно-заочное, заочное

Вид контроля зачет  
экзамен, зачет;

Вид аттестации промежуточная  
текущая, промежуточная

#### Контрольно-измерительный материал № 1

1. Оператор выбора switch. Вложенные инструкции if и switch. "Лестничная" конструкция if-eise-if.  
Примеры.

2. Составьте алгоритм и напишите программу, выполняющие следующие вычисления:

На всей области определения функции  $y = \begin{cases} x-1 & \text{при } 1 \leq x \leq 2; \\ x^2-3 & \text{при } 2 \leq x < 3; \\ 6 & \text{при } 3 \leq x \leq 5 \end{cases}$  найти ее значения в точках  $x_k = a + \frac{b-a}{n}k$ ,

$k = \overline{0, n}$ . Величина  $n$  вводится с клавиатуры. Полученные значения выведите на экран в 6 столбиков.

Преподаватель \_\_\_\_\_ С.Н. Ушаков  
подпись расшифровка подписи

## Форма контрольно-измерительного материала № 2

УТВЕРЖДАЮ  
Заведующий кафедрой функционального анализа и операторных уравнений

\_\_\_\_\_  
подпись, расшифровка подписи

\_\_\_. \_\_. 20\_\_

Направление подготовки / специальность 01.03.04 Прикладная математика  
шифр, наименование  
Дисциплина Б1.О.12 Программирование для ЭВМ  
Форма обучения очная  
очное, очно-заочное, заочное  
Вид контроля экзамен  
экзамен, зачет;  
Вид аттестации промежуточная  
текущая, промежуточная

### Контрольно-измерительный материал № 1

1. Перегрузка операций. Ограничения перегрузки.

2. Чисто виртуальные функции и абстрактные классы. Раннее и позднее связывание.

3. Составьте алгоритм и напишите программу, создающую класс матриц со следующими нестандартными операциями: умножения  $A * B = (a_{kj}b_{kj} - a_{jk}b_{jk})$  и умножения на число  $c$ :  $c * A = (c - a_{kj})$  и  $A * c = (c + a_{kj})$ .

Преподаватель \_\_\_\_\_ С.Н. Ушаков  
подпись расшифровка подписи

## Оформление комплекта заданий для контрольной работы

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

### Кафедра функционального анализа и операторных уравнений

(наименование кафедры)

## Комплект заданий для контрольной работы № 1

по дисциплине *Программирование для ЭВМ*  
(наименование дисциплины)

**Темы:** Введение в язык C++; Типы данных и выражения.

**Задание.** Составьте алгоритм и напишите программу для решения ниже приведенной задачи. Ввод данных и вывод результатов оформляйте с выводом на экран поясняющих сообщений. Проведите вычисления при четырех различных значениях параметров задачи.

#### Вариант 1

Даны три точки A, B, C на числовой оси. Найти длины отрезков AC и BC, а также сумму и разность квадратов найденных длин.

#### Вариант 2

Дано число A. Вычислить  $A^8$ , используя вспомогательную переменную и три операции умножения. Для этого последовательно найдите  $A^2$ ,  $A^4$ ,  $A^8$ . Выведите число A и все найденные степени числа A.

#### Критерии оценки:

- оценка «отлично» выставляется студенту, если правильно и рационально составлен алгоритм, написана работающая программа, соответствующая составленному алгоритму;
- оценка «хорошо» выставляется студенту, если правильно, но нерационально составлен алгоритм, написана работающая программа, соответствующая составленному алгоритму;
- оценка «удовлетворительно» выставляется студенту, если составлен алгоритм с негрубыми ошибками и написана работающая программа, соответствующая составленному алгоритму;
- оценка «неудовлетворительно» выставляется студенту, если алгоритм не составлен или составлен алгоритм с грубыми ошибками; не написана программа или написана неработающая программа.

## Оформление комплекта заданий для контрольной работы

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

Кафедра функционального анализа и операторных  
уравнений

(наименование кафедры)

## Комплект заданий для контрольной работы № 2

по дисциплине *Программирование для ЭВМ*  
(наименование дисциплины)

**Тема** Управляющие структуры

**Задание.** Составьте алгоритм и напишите программу для вычисления ниже следующих сумм с заданной точностью  $\epsilon$ . Проведите вычисления при различных значениях параметра  $\epsilon$ .

**Вариант 1**  $\sum_{n=1}^{\infty} \frac{\sin n^2}{n\sqrt{n+1}};$

**Вариант 2**  $\sum_{n=1}^{\infty} \frac{1}{n \ln(\sin \frac{1}{n})}$

**Критерии оценки:**

- оценка «отлично» выставляется студенту, если правильно и рационально составлен алгоритм, написана работающая программа, соответствующая составленному алгоритму;
- оценка «хорошо» выставляется студенту, если правильно, но нерационально составлен алгоритм, написана работающая программа, соответствующая составленному алгоритму;
- оценка «удовлетворительно» выставляется студенту, если составлен алгоритм с негрубыми ошибками и написана работающая программа, соответствующая составленному алгоритму;
- оценка «неудовлетворительно» выставляется студенту, если алгоритм не составлен или составлен алгоритм с грубыми ошибками; не написана программа или написана неработающая программа.

## Оформление комплекта заданий для контрольной работы

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

### Кафедра функционального анализа и операторных уравнений

(наименование кафедры)

### Комплект заданий для контрольной работы № 3

по дисциплине *Программирование для ЭВМ*  
(наименование дисциплины)

**Тема** Массивы и строки

#### **Вариант 1**

Составьте алгоритм и напишите программу, позволяющую записать в отдельный массив сначала все отрицательные, а затем все положительные элементы исходного массива. Найти сумму всех положительных и сумму всех отрицательных элементов массива.

#### **Вариант 2**

Составьте алгоритм и напишите программу, позволяющую найти сумму отрицательных элементов массива и записать в отдельный массив их номера.

#### **Критерии оценки:**

- оценка «отлично» выставляется студенту, если правильно и рационально составлен алгоритм, написана работающая программа, соответствующая составленному алгоритму;
- оценка «хорошо» выставляется студенту, если правильно, но нерационально составлен алгоритм, написана работающая программа, соответствующая составленному алгоритму;
- оценка «удовлетворительно» выставляется студенту, если составлен алгоритм с негрубыми ошибками и написана работающая программа, соответствующая составленному алгоритму;
- оценка «неудовлетворительно» выставляется студенту, если алгоритм не составлен или составлен алгоритм с грубыми ошибками; не написана программа или написана неработающая программа.



## Оформление комплекта заданий для контрольной работы

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

Кафедра функционального анализа и операторных  
уравнений

(наименование кафедры)

### Комплект заданий для контрольной работы № 4

по дисциплине *Программирование для ЭВМ*  
(наименование дисциплины)

**Тема** Указатели

#### **Вариант 1**

Составьте алгоритм и напишите программу, позволяющую записать в отдельный массив сначала все отрицательные, а затем все положительные элементы исходного массива. Найти сумму всех положительных и сумму всех отрицательных элементов массива. При обработке массива используйте указатели. Исходный и полученный массивы выведите на экран.

#### **Вариант 2**

Составьте алгоритм и напишите программу, позволяющую найти сумму отрицательных элементов массива и записать в отдельный массив их номера. При обработке массива используйте указатели. Исходный и полученный массивы выведите на экран.

**Критерии оценки:**

- оценка «отлично» выставляется студенту, если правильно и рационально составлен алгоритм, написана работающая программа, соответствующая составленному алгоритму;
- оценка «хорошо» выставляется студенту, если правильно, но нерационально составлен алгоритм, написана работающая программа, соответствующая составленному алгоритму;
- оценка «удовлетворительно» выставляется студенту, если составлен алгоритм с негрубыми ошибками и написана работающая программа, соответствующая составленному алгоритму;
- оценка «неудовлетворительно» выставляется студенту, если алгоритм не составлен или составлен алгоритм с грубыми ошибками; не написана программа или написана неработающая программа.

## Оформление комплекта заданий для контрольной работы

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

Кафедра функционального анализа и операторных  
уравнений

(наименование кафедры)

### Комплект заданий для контрольной работы № 5

по дисциплине *Программирование для ЭВМ*  
(наименование дисциплины)

#### Тема Функции

Геометрическая фигура, заданная на плоскости координатами своих вершин и состоящая из набора отрезков, делит плоскость на три открытые области (без границы). Составьте алгоритм и напишите функции, которые проверяют, лежит ли данная точка на геометрической фигуре, а если не лежит, то какой области принадлежит. Используйте функции, позволяющие проверить: лежит ли данная точка на заданной прямой; принадлежит ли данная точка заданному отрезку; лежат ли две данные точки по одну или по разные стороны от заданной прямой.. Проверьте правильность работы функции и всей программы в целом.

#### Вариант 1

Треугольник со средней линией.

#### Вариант 2

Треугольник с медианой.

#### Критерии оценки:

- оценка «отлично» выставляется студенту, если правильно и рационально составлен алгоритм, написана работающая программа, соответствующая составленному алгоритму;
- оценка «хорошо» выставляется студенту, если правильно, но нерационально составлен алгоритм, написана работающая программа, соответствующая составленному алгоритму;
- оценка «удовлетворительно» выставляется студенту, если составлен алгоритм с негрубыми ошибками и написана работающая программа, соответствующая составленному алгоритму;
- оценка «неудовлетворительно» выставляется студенту, если алгоритм не составлен или составлен алгоритм с грубыми ошибками; не написана программа или написана неработающая программа.

## Оформление комплекта заданий для контрольной работы

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

### Кафедра функционального анализа и операторных уравнений

(наименование кафедры)

### Комплект заданий для контрольной работы № 6

по дисциплине *Программирование для ЭВМ*  
(наименование дисциплины)

**Темы** О типах данных; Поразрядные операторы

Составьте алгоритм и напишите программу для выполнения указанных операций с двумя целыми числами. Программа не должна содержать стандартные операции сложения (+), вычитания (–), умножения (\*), деления (/) и остатка от деления (%). Используйте поразрядные операторы. Операции оформите в виде отдельных функций.

**Вариант 1**

Сложение и вычисление целой части частного и остатка при делении целого числа на число 2

**Вариант 2**

Вычитание и умножения целого числа на число 2

**Критерии оценки:**

- оценка «отлично» выставляется студенту, если правильно и рационально составлен алгоритм, написана работающая программа, соответствующая составленному алгоритму;

- оценка «хорошо» выставляется студенту, если правильно, но нерационально составлен алгоритм, написана работающая программа, соответствующая составленному алгоритму;

- оценка «удовлетворительно» выставляется студенту, если составлен алгоритм с негрубыми ошибками и написана работающая программа, соответствующая составленному алгоритму;

- оценка «неудовлетворительно» выставляется студенту, если алгоритм не составлен или составлен алгоритм с грубыми ошибками; не написана программа или написана неработающая программа.

## 20.2. Промежуточная аттестация

Промежуточная аттестация по дисциплине осуществляется с помощью следующих оценочных средств: КИМ1, КИМ2 и Перечень вопросов к экзамену.

### Перечень вопросов к экзамену:

1. Основные принципы ООП (Инкапсуляция, наследование, полиморфизм).
2. Классы. Общий формат объявления. Доступ к членам класса.
3. Конструкторы и деструкторы. Параметризованные конструкторы. Альтернативный вариант инициализации конструктора.
4. Классы и структуры — родственные типы. Классы и объединения — родственные типы.
5. Встраиваемые функции. Перегрузка конструкторов.
6. Массивы объектов. Инициализация массивов объектов. Указатели на объекты. Ссылки на объекты.
7. Ключевое слово `this`. Функции-друзья.
8. Конструкторы, деструкторы и передача объектов. Потенциальные проблемы при передаче параметров.
9. Возвращение объектов функциями. Потенциальная проблема при возвращении объектов функциями.
10. Создание и использование конструктора копии. Конструкторы копии и параметры функции. Использование конструкторов копии при инициализации объектов. Использование конструктора копии при возвращении функцией объекта.
11. Перегрузка операций. Ограничения перегрузки.
12. Перегрузка унарных операций. Постфиксная операция. Временные безымянные объекты. Перегрузка операции индексации массива (`[]`).
13. Наследование. Понятие о наследовании. Управление доступом к членам базового класса.
14. Использование спецификаторов `public`, `protected` и `private` для наследования базового класса.
15. Наследование нескольких базовых классов. Конструкторы, деструкторы и наследование.
16. Передача параметров конструкторам базового класса.
17. Виртуальные базовые классы. (Проблемы неоднозначности. 2 способа решения проблемы.)
18. Указатели на производные типы. Ссылки на производные типы.
19. Виртуальные функции. Наследование виртуальных функций.
20. Чисто виртуальные функции и абстрактные классы. Раннее и позднее связывание.
21. Обобщённые функции. Функция с двумя обобщёнными типами.
22. Явно заданная перегрузка обобщенной функции. Перегрузка шаблона функции. Ограничения при использовании обобщенных функций.
23. Обобщенные классы. Пример класса с двумя обобщенными типами данных. Создание обобщенного класса безопасного массива.
24. Использование в обобщенных классах аргументов, не являющихся типами. Использование в шаблонных классах аргументов по умолчанию. Явно задаваемые специализации классов.

25 Основы обработки исключительных ситуаций. Функции exit() и abort().

26 Перехват исключений классового типа. Использование нескольких catch-инструкций. Перехват исключений базового класса.

27 Варианты обработки исключений Перехват всех исключений. Ограничения, налагаемые на тип исключений, генерируемых функциями.

28 Повторное генерирование исключения. Обработка исключений, сгенерированных оператором new. Перегрузка операторов new и delete

Для оценивания результатов обучения на экзамене используется 4-балльная шкала: «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

Критерии оценивания компетенций	Уровень сформированности компетенций	Шкала оценок
<i>Обучающийся в полной мере владеет понятийным аппаратом языка C++, способен иллюстрировать ответ примерами, применять теоретические знания для создания алгоритмов и реализации в виде программ на языке C++</i>	<i>Повышенный уровень</i>	<i>Отлично</i>
<i>Обучающийся владеет понятийным аппаратом языка C++, способен иллюстрировать ответ примерами допускает ошибки при создании алгоритмов и реализации в виде программ на языке C++ или способен применять теоретические знания для для создания алгоритмов и реализации в виде программ на языке C++, но допускает неточности при применении понятийного аппарата языка C++, но отвечает на дополнительные вопросы</i>	<i>Базовый уровень</i>	<i>Хорошо</i>
<i>Обучающийся владеет частично теоретическими основами дисциплины, фрагментарно способен иллюстрировать ответ примерами, фактами, не отвечает на дополнительные вопросы Не умеет применять теоретические знания для создания алгоритмов и реализации в виде программ на языке C++</i>	<i>Пороговый уровень</i>	<i>Удовлетворительно</i>
<i>Ответ на контрольно-измерительный материал не соответствует любым трем(четырем) из перечисленных показателей. Обучающийся демонстрирует отрывочные, фрагментарные знания, допускает грубые ошибки.</i>	<i>–</i>	<i>Неудовлетворительно</i>

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

Задания открытого типа:

1. Определите значение переменной a после выполнения фрагмента программы:

```
int a = 10, b = 5;  
if (a > 5 && a < b)  
    a -= 5;
```

Ответ: 10.

Решение. Поскольку условие if не выполнено, значение переменной a остается равным 10.

2. Определите значение переменной a после выполнения фрагмента программы:

```
int a = 10, b = 5;  
if (a > 1 || a < b)  
    a -= 5;  
if (a > 1 && a == b)
```

```
a -= 5;
```

Ответ: 0.

Решение. Поскольку условие в первом if выполнено, значение переменной a становится равным 5. Тогда и условие во втором if выполнено, и значение переменной a становится равным 0.

3. Какую логическую операцию нужно добавить в программу вместо многоточия, чтобы значение переменной a после выполнения фрагмента программы стало равно 17?

```
int a = 10, b = 5;  
if (a > 1 ... a < b)  
    a -= 7;  
else  
    a += 7;
```

Ответ: &&.

Решение. Нужно добавить такую логическую операцию, чтобы условие в if не выполнялось, а это операция &&.

4. Чему будет равно значение целой переменной a после выполнения этого фрагмента программы?

```
int a = 10;  
for (int i = 1; i <= 3; i++)  
    a -= 2;
```

Ответ: 4.

Решение. Тело цикла for выполнится ровно 3 раза, следовательно, значение переменной a уменьшится на 6 и станет равным 4.

5. Укажите тип возвращаемого значения следующей функции:

```
int func(char x, float v, double t);
```

Ответ: int.

Решение. Тип возвращаемого значения указывается в самом начале объявления функции.

Задания закрытого типа:

1. Какой из ниже перечисленных операторов, не является циклом в C++?

- a) for
- b) do while
- c) while
- d) repeat until

Ответ: d) repeat until.

2. До каких пор будут выполняться операторы в теле цикла while (x < 100)?

- a) Пока x равен ста.
- b) Пока x меньше или равен ста.
- c) Пока x строго меньше ста.
- d) Пока x больше ста.

Ответ: c) Пока x строго меньше ста.

3. Какой из следующих операторов – оператор сравнения двух переменных?

- a) equal
- b) :=
- c) =
- d) ==

Ответ: d) ==.

4. Укажите правильный вызов функции. Предполагается, что функция была объявлена ранее.

- a) int func();
- b) func x, y;
- c) func;
- d) func();

Ответ: d) func();

5. Укажите правильное объявление массива.

- a) array an array[10];
- b) int anarray;
- c) int anarray[10];
- d) anarray{10};

Ответ: c) int anarray[10];

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