

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

УТВЕРЖДАЮ

Заведующий кафедрой  
Математического обеспечения ЭВМ  
наименование кафедры, отвечающей за реализацию дисциплины



Абрамов Г.В.  
подпись, расшифровка подписи  
23.03.2024г.

**РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ**  
**Б1.О.23 Объектно-ориентированное программирование**  
*Код и наименование дисциплины в соответствии с учебным планом*

1. Код и наименование направления подготовки/специальности:

01.03.02 Прикладная математика и информатика

2. Профиль подготовки/специализация: Прикладная математика и компьютерные технологии

3. Квалификация (степень) выпускника: бакалавр

4. Форма обучения: очная

5. Кафедра, отвечающая за реализацию дисциплины: МО ЭВМ

6. Составители программы: Чернышов М.К., к.ф.-м.н., доцент  
(ФИО, ученая степень, ученое звание)

7. Рекомендована: НМС факультета ПММ, протокол № 5 от 22.03.2024

*отметки о продлении вносятся вручную)*

8. Учебный год: 2024-2025

Семестр(ы): 3

9. Цели и задачи учебной дисциплины:

Цель изучения дисциплины: изучение студентами теоретических основ технологии объектно-ориентированного программирования, принципов ее реализации, методов разработки программ, обработки данных.

Задачи учебной дисциплины: обучение студентов использовать современные технологии разработки программ с учетом требований предметной области и потребностей пользователей. Обучение грамотному использованию парадигмы объектно-ориентированного программирования для представления и хранения информации. Выработка практических навыков применения полученных знаний с использованием современных языков программирования.

**10. Место учебной дисциплины в структуре ООП:** учебная дисциплина относится к обязательной части Блока 1. Предшествующей для данной дисциплины является дисциплина Информатика и программирование.

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

Код	Название компетенции	Код(ы)	Индикатор(ы)	Планируемые результаты обучения
ОПК-2	Способен использовать и адаптировать существующие математические методы и системы программирования для разработки и реализации алгоритмов решения прикладных задач	ОПК-2.2	Сравнивает системы программирования для обоснования выбора программной среды для разработки компонентов программного обеспечения	уметь: проектировать программные приложения на основе принципов ООП
ОПК-3	Способен применять и модифицировать математические модели для решения задач в области профессиональной деятельности	ОПК-3.3	Применяет системы компьютерного моделирования для построения и анализа моделей при решении задач в области профессиональной деятельности	владеть: практическими навыками применения технологии ООП
ОПК-4	Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности	ОПК-4.2	Осуществляет выбор и применяет информационные технологии для решения задач профессиональной деятельности	уметь: проектировать программные приложения на основе принципов ООП
ОПК-5	Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения	ОПК-5.3	Использует основные положения и концепции прикладного и системного программирования, современные языки программирования, технологии создания и эксплуатации программ и программных комплексов при решении задач профессиональной деятельности.	знать: основы технологии объектно-ориентированного программирования, принципы ее реализации  владеть: практическими навыками применения технологии ООП
ОПК-5	Способен разрабатывать алгоритмы и компьютерные программы, пригодные	ОПК-5.2	Разрабатывает программное и информационное обеспечение компьютерных сетей,	владеть: практическими навыками применения технологии ООП

для практического применения		автоматизированных систем, сервисов, операционных систем и баз данных с учетом основных требований информационной безопасности	
------------------------------	--	--	--

**12. Объем дисциплины в зачетных единицах/час.**(в соответствии с учебным планом) — 2 / 72.

**Форма промежуточной аттестации**(зачет/экзамен) зачет.

### 13. Трудоемкость по видам учебной работы

Вид учебной работы	Трудоемкость	
	Всего	По семестрам
		3 семестр
Аудиторные занятия	48	48
в том числе:	лекции	16
	практические	16
	лабораторные	16
Самостоятельная работа	24	24
Форма промежуточной аттестации (зачет – 0 час. / экзамен – ___ час.)	0	0
Итого:	72	72

#### 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	Методология программирования	Декомпозиция и абстракция. Абстракция через параметризацию и спецификацию. Процедурная абстракция и абстракция данных. Классы операций. Полнота.	Объектно-ориентированное программирование (2021-2022), ПМИ <a href="https://edu.vsu.ru/course/view.php?id=17770">https://edu.vsu.ru/course/view.php?id=17770</a>
1.2	Объектно-ориентированный подход в программировании	Развитие технологии и языков программирования. История ООП. Инкапсуляция. Наследование. Полиморфизм.	
1.3	Объектно-ориентированные средства C++	Перегружаемые функции и операторы. Объектные типы данных: структуры, объединения, классы. Конструкторы и деструкторы. Конструкторы с параметрами, перегрузка конструкторов. Встраиваемые функции. Присваивание объектов. Передача параметров в функции и возвращение функцией объекта. Конструктор копирования. Указатели и ссылки на объекты.	
1.4	Наследование в языке C++	Модификаторы наследования. Конструкторы при наследовании.	

		Деструкторы при наследовании. Совместимость типов.	
1.5	Виртуальные функции	Раннее и позднее связывание. Полиморфизм и виртуальные методы. Использование указателей на базовые классы при адресации объектов производных классов. Абстрактный класс.	
1.6	«Дружественные» функции	«Дружественные» (friend) функции. Переопределение операторов с помощью дружественных функций.	
1.7	Шаблоны (параметризованные типы) функций и классов	Шаблоны функций. Шаблоны классов.	
<b>2. Практические занятия</b>			
2.1	Объектно-ориентированные средства C++	Структуры и классы. Конструкторы и деструкторы.	Объектно-ориентированное программирование (2021-2022), ПМИ <a href="https://edu.vsu.ru/course/view.php?id=17770">https://edu.vsu.ru/course/view.php?id=17770</a>
		Конструктор копирования. Статические члены класса.	
		Перегрузка операций	
2.2	Наследование в языке C++	Наследование. Конструкторы и деструкторы при наследовании	
2.3	Виртуальные функции	Раннее и позднее связывание. Полиморфизм и виртуальные методы. Абстрактные классы	
2.4	«Дружественные» функции	Переопределение операторов с помощью дружественных функций.	
2.5	Шаблоны (параметризованные типы) функций и классов	Шаблоны функций. Шаблоны классов.	
<b>3. Лабораторные работы</b>			
3.1	Объектно-ориентированные средства C++	Простой класс с двумя полями. Релизация конструкторов, деструкторов, простых методов класса.	Объектно-ориентированное программирование (2021-2022), ПМИ <a href="https://edu.vsu.ru/course/view.php?id=17770">https://edu.vsu.ru/course/view.php?id=17770</a>
3.2	Наследование в языке C++	Наследование классов. Реализация конструкторов и деструкторов, методов класса при наследовании.	
3.3	Виртуальные функции	Виртуальные методы и абстрактные классы	
3.4	«Дружественные» функции	Использование дружественных функций	
3.5	Шаблоны (параметризованные типы) функций и классов	Реализация шаблонов классов	

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

№ п/п	Наименование темы (раздела) дисциплины	Виды занятий (часов)				
		Лекции	Практические	Лабораторные	Самостоятельная работа	Всего
1.1	Методология программирования	2	-	-	2	4
1.2	Объектно-ориентированный подход в программировании	2	-	-	2	4
1.3	Объектно-ориентированные средства C++	2	6	2	4	14
1.4	Наследование в языке C++	2	2	2	4	10
1.5	Виртуальные функции	2	4	4	4	14
1.6	«Дружественные» функции	2	2	4	4	14
1.7	Шаблоны (параметризованные типы) функций и классов	2	2	4	4	12
	Итого:	16	16	16	24	72

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

При работе с лекционным материалом рекомендуется использовать рекомендуемую литературу по соответствующим темам.

При использовании дистанционных образовательных технологий и электронного обучения выполнять все указания преподавателей по работе на LMS-платформе, своевременно подключаться к online-занятиям, соблюдать рекомендации по организации самостоятельной работы

#### 15. Перечень основной и дополнительной литературы, ресурсов интернет, необходимых для освоения дисциплины (список литературы оформляется в соответствии с требованиями ГОСТ и используется общая сквозная нумерация для всех видов источников)

а) основная литература:

№ п/п	Источник
1.	Зыков, С.В. Введение в теорию программирования. Объектно-ориентированный подход / С.В. Зыков. - 2-е изд., испр. - М. : Национальный Открытый Университет «ИНТУИТ», 2016. - 189 с. : схем. - (Основы информационных технологий). - Библиогр. в кн. - ISBN 5-9556-0009-4 ; То же [Электронный ресурс]. - URL: <a href="https://biblioclub.lib.vsu.ru/index.php?page=book&amp;id=429073">https://biblioclub.lib.vsu.ru/index.php?page=book&amp;id=429073</a> (04.10.2016).
2.	Объектно-ориентированное программирование / . - Новосибирск : НГТУ, 2010. - 44 с. ; То же [Электронный ресурс]. - URL: <a href="https://biblioclub.lib.vsu.ru/index.php?page=book&amp;id=229136">https://biblioclub.lib.vsu.ru/index.php?page=book&amp;id=229136</a> (04.10.2016).
3.	Мейер, Б. Объектно-ориентированное программирование и программная инженерия / Б. Мейер. - 2-е изд., испр. - М. : Национальный Открытый Университет «ИНТУИТ», 2016. - 286 с. : ил. ; То же [Электронный ресурс]. - URL: <a href="https://biblioclub.lib.vsu.ru/index.php?page=book&amp;id=429034">https://biblioclub.lib.vsu.ru/index.php?page=book&amp;id=429034</a> (04.10.2016).

б) дополнительная литература:

№ п/п	Источник
1.	Клочков Д.П., Павлов Д.А. Введение в объектно-ориентированное программирование / Учебно-методическое пособие. - Изд. Нижегород. ун-та, 1995. - 70с.

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

№ п/п	Ресурс
1.	<a href="http://www.lib.vsu.ru">www.lib.vsu.ru</a>
2.	Объектно-ориентированное программирование (2021-2022), ПМИ <a href="https://edu.vsu.ru/course/view.php?id=17770">https://edu.vsu.ru/course/view.php?id=17770</a>

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

№ п/п	Источник
1.	Авсеева О.В. Объектно-ориентированное программирование, часть 1 (учебно-методическое пособие) // О.В.Авсеева, М.К. Чернышов. - Воронеж : Издательский дом ВГУ, 2021. – Тираж 50. – 82 с.
2.	Авсеева О.В. Объектно-ориентированное программирование, часть 2 (учебно-методическое пособие) // О.В.Авсеева, М.К. Чернышов. - Воронеж : Издательский дом ВГУ, 2021. – Тираж 50. – 96 с.

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

При реализации дисциплины используются модульно-рейтинговая и личностно-ориентированные технологии обучения (ориентированные на индивидуальность студента, компьютерные и коммуникационные технологии). В рамках дисциплины предусмотрены следующие виды лекций: информационная, лекция с применением обратной связи.

Дисциплина реализуется с применением электронного обучения и дистанционных образовательных технологий, для организации самостоятельной работы обучающихся используется онлайн-курс, размещенный на платформе Электронного университета ВГУ (LMS moodle), а также другие Интернет-ресурсы, приведенные в п.15в

## 18. Материально-техническое обеспечение дисциплины:

1. Аудитория для проведения лекционных занятий - 394018, г. Воронеж, площадь Университетская, д. 1, пом. I, ауд. 433, Компьютер преподавателя, мультимедиа оборудование (проектор, средства звуковоспроизведения), доска меловая, специализированная мебель.

2. Аудитория для проведения лабораторных занятий - 394018, г. Воронеж, площадь Университетская, д. 1, пом. I, ауд. 10, Компьютер преподавателя; компьютер учащегося (22 шт.), мультимедиа оборудование (проектор, средства звуковоспроизведения); доска магнитно-маркерная на стенде, 2-сторонняя, специализированная мебель.

3. Аудитория для проведения практических занятий - 394018, г. Воронеж, площадь Университетская, д. 1, пом. I, ауд. 226, Компьютер преподавателя, мультимедиа оборудование (проектор, средства звуковоспроизведения, микрофон), маркерные панели, специализированная мебель.

## 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.	Методология программирования	ОПК-4	ОПК-4.2	<i>Тест</i>
2.	Объектно-ориентированный подход в программировании	ОПК-5 ОПК-2	ОПК-5.3 ОПК-2.2	<i>Лабораторные работы</i>
3.	Объектно-ориентированные средства C++	ОПК-5	ОПК-5.2	<i>Тест Лабораторные работы</i>
4.	Наследование в языке C++	ОПК-5	ОПК-5.2	<i>Тест Лабораторные работы</i>
5.	«Дружественные» функции	ОПК-5	ОПК-5.3	<i>Тест Лабораторные работы</i>
6.	Шаблоны (параметризованные типы) функций и классов	ОПК-5	ОПК-5.3	<i>Лабораторные работы</i>
7.	Преобразования типов	ОПК-3	ОПК-3.3	<i>Тест</i>
Промежуточная аттестация форма контроля – зачет				<i>Перечень вопросов Практическое задание</i>

## 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:

*Лабораторные работы*

*Тестовые задания*

**Пример задания на лабораторную работу**

Создать класс с полями, указанными в индивидуальном задании. Реализовать в классе методы:

- конструктор по умолчанию;
- конструктор с параметрами;
- деструктор для освобождения памяти (с сообщением об уничтожении объекта);
- функции обработки данных (1 и 2), указанные в индивидуальном задании;
- функцию формирования строки информации о структуре объекта.

#### **Пример тестового задания:**

Разработать структуру класса, описывающего трехмерный физический вектор, реализовав операции сложения, вычитания векторов, нахождения скалярного произведения векторов и вычисления их длины.

## **20.2 Промежуточная аттестация**

Промежуточная аттестация по дисциплине осуществляется с помощью следующих оценочных средств:

### *Тестовые задания*

---

#### **Перечень вопросов к зачету:**

1. Объектно-ориентированный подход в программировании
2. Сущность объектно-ориентированного подхода к программированию
3. Введение в классы
4. Перегружаемые функции и операторы
5. Встраиваемые функции
6. Конструкторы и деструкторы
7. Конструкторы с параметрами и перегрузка конструкторов
8. Присваивание объектов
9. Передача в функции и возвращение объекта
10. Указатели и ссылки на объекты
11. Статические компоненты класса
12. Наследование в языке C++
13. Модификаторы наследования
14. Конструкторы и деструкторы при наследовании
15. Пример построения классов при наследовании
16. Совместимость типов
17. Дружественные функции
18. Доступ к защищенным элементам класса
19. Переопределение операторов с помощью дружественных функций
20. Виртуальные методы
21. Раннее и позднее связывание
22. Виртуальные функции
23. Полиморфизм и виртуальные методы
24. Указатели на базовые классы
25. Абстрактный класс
26. Шаблоны функций
27. Требования к фактическим параметрам шаблона
28. Отождествление типов аргументов
29. Шаблоны классов
30. Наследование в шаблонах классов
31. Декомпозиция и абстракция
32. Абстракция через параметризацию
33. Абстракция через спецификацию
34. Модификатора доступа и наследования. Как изменяются атрибуты элементов класса при наследовании?
35. В чем разница между struct, class и union?

36. Что такое встраиваемая функция? В чем ее преимущества и недостатки?
37. Какие два способа существуют для создания встраиваемой функции?
38. Как можно передать в функцию в качестве аргумента адрес объекта?
39. Что такое дружественная функция?
40. Когда следует переопределять операторы с помощью дружественных функций, а когда с помощью функций элементов класса?
41. Что происходит при присваивании одного объекта другому (без переопределенной операции =) и какие побочные эффекты могут возникнуть?
42. При передаче объекта в функцию в качестве аргумента создается копия этого объекта. Изменение копии в теле функции не отражается на оригинале. Возможно ли нарушение этого правила?
43. Какая функция может иметь доступ к защищенным членам одного класса? ... двух классов?
44. Что такое указатель this. Приведите пример использования этого указателя.
45. Для чего необходимы операторы new и delete. В чем их отличие от функций malloc() и free()?
46. Что такое ссылка? Какое имеется преимущество при использовании ссылки в качестве параметра функции, а в чем недостаток? В чем разница между ссылкой и указателем?
47. Назовите причины, по которым может понадобиться перегрузка конструкторов и деструкторов (в одном классе).
48. Какова основная форма конструктора копирования и когда он вызывается?
49. Что такое аргумент по умолчанию? Как аргумент по умолчанию связан с перегрузкой функций?
50. Почему может потребоваться перегрузка оператора присваивания?
51. Можно ли изменить приоритет перегруженного оператора?
52. Какие функции не могут быть виртуальными?
53. Что такое абстрактный класс и чем может быть вызвана необходимость построений абстрактного класса?
54. Чем виртуальные функции отличаются от перегружаемых?
55. Совместимость типов. Какую роль в достижении совместимости объектов могут играть абстрактные классы?
56. Почему шаблоны называют параметризованными типами?
57. Когда следует в программе применять шаблоны, а когда нет?
58. Чем шаблоны лучше макроподстановок?
59. Для каких типов данных может применяться конкретный шаблон, а для каких нет?
60. В чем разница между классом и шаблоном класса?
61. Что может выступать в качестве параметра для шаблона класса?

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

Для оценивания результатов обучения на зачете используются следующие показатели:

- 1) знание учебного материала и владение понятийным аппаратом объектно-ориентированного программирования;
- 2) умение связывать теорию с практикой;
- 3) умение иллюстрировать ответ примерами;
- 4) умение применять принципы объектно-ориентированного программирования при проектировании классов

Для оценивания результатов обучения на зачете используется 2-балльная шкала: «зачтено», «не зачтено».

Соотношение показателей, критериев и шкалы оценивания результатов обучения.

Критерии оценивания компетенций	Уровень сформированности компетенций	Шкала оценок
---------------------------------	--------------------------------------	--------------



<p>Полное соответствие ответа обучающегося всем перечисленным критериям. Продемонстрировано знание принципов объектно-ориентированного программирования, умение проектировать классы с использованием принципов ООП, владение объектно-ориентированными средствами языка C++. Сданы все лабораторные работы.</p>	<p>Повышенный уровень</p>	<p>Зачтено</p>
<p>Ответ на контрольно-измерительный материал не соответствует одному (двум) из перечисленных показателей, но обучающийся дает правильные ответы на дополнительные вопросы. Недостаточно продемонстрировано знание принципов ООП или содержатся отдельные пробелы в знании средств реализации ООП на языке C++. Сданы все лабораторные работы.</p>	<p>Базовый уровень</p>	<p>Зачтено</p>
<p>Ответ на контрольно-измерительный материал не соответствует любым двум(трем) из перечисленных показателей, обучающийся дает неполные ответы на дополнительные вопросы. Демонстрирует частичные знания принципов ООП, или не умеет использовать средства языка C++ или допускает существенные ошибки в их применении. Сданы все лабораторные работы.</p>	<p>Пороговый уровень</p>	<p>Зачтено -</p>
<p>Сданы не все лабораторные работы. Ответ на контрольно-измерительный материал не соответствует любым трем(четырем) из перечисленных показателей. Обучающийся демонстрирует отрывочные, фрагментарные знания, допускает грубые ошибки</p>	<p>–</p>	<p>Не зачтено</p>