

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

УТВЕРЖДАЮ

Заведующий кафедрой МО ЭВМ

  
Г.В.Абрамов

23.03.2024 г.

**РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ**

**Б1.В.07 Программирование на C#**

- 1. Код и наименование направления подготовки/специальности:**  
02.03.02 Фундаментальная информатика и информационные технологии
- 2. Профиль подготовки/специализация:** Инженерия программного обеспечения
- 3. Квалификация (степень) выпускника:** бакалавр
- 4. Форма обучения:** очная
- 5. Кафедра, отвечающая за реализацию дисциплины:**  
Математического обеспечения ЭВМ (МО ЭВМ)
- 6. Составители программы:** Курченкова Т.В., к.т.н., доцент
- 7. Рекомендована:** НМС протокол №5 от 22.03.2024  
(наименование рекомендующей структуры, дата, номер протокола,

---

*отметки о продлении вносятся вручную)*

**8. Учебный год:** 2026/2027

**Семестр(ы):** 5



## 9. Цели и задачи учебной дисциплины

Цель изучения дисциплины: обучение основам и принципам объектно-ориентированного программирования, приобретение навыков создания программ на языке C#, способности реализовывать и адаптировать прикладное программное обеспечение.

Задачи учебной дисциплины: изучение основных принципов парадигмы объектно-ориентированного программирования, изучение языка C#, его стандартных библиотек, классов и шаблонов; выполнение работ по созданию (модификации) и сопровождению ППО; разработка требований и проектирование программного обеспечения.

**10. Место учебной дисциплины в структуре ООП:** часть, формируемая участниками образовательных отношений (вариативная) блока Б1. Требования к входным знаниям: программирование, структуры и алгоритмы обработки данных. Дисциплина является предшествующей для следующих:

- программная инженерия;
- язык программирования Java;
- проектирование информационных систем;
- интеллектуальные информационные системы;
- компьютерное моделирование;
- разработка приложений с использованием фреймворков.

**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
ПК-6	Способность применять в профессиональной деятельности современные языки программирования и методы параллельной обработки данных, операционные системы, электронные библиотеки и пакеты программ, сетевые технологии	ПК-6.1	Использует в профессиональной деятельности объектно-ориентированные системы программирования	<b>Знать:</b> основные этапы компьютерного решения задач; понятие алгоритма и структуры управления; структуры данных; основные требования методологии объектно-ориентированного программирования как технологической основы разработки качественных программных компонентов. <b>Уметь:</b> применять требования методологии объектно-ориентированного программирования при разработке программ; разрабатывать и записывать на языке программирования алгоритмы решения классических задач информатики; применять средства поддержки объектно-ориентированного программирования в языке программирования; выбрать оптимальную структуру для представления данных <b>Владеть:</b> навыками проектирования, реализации программ

**12. Объем дисциплины в зачетных единицах/час. (в соответствии с учебным планом) — 4 / 144.**

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

### 13. Трудоемкость по видам учебной работы

Вид учебной работы		Трудоемкость	
		Всего	По семестрам
			№ 8
Аудиторные занятия		64	64
в том числе:	лекции	32	32
	практические		
	лабораторные	32	32
Самостоятельная работа		44	44
в том числе: курсовая работа (проект)			
Форма промежуточной аттестации (зачет – 0 час. / экзамен – __ час.)			
Итого:		144	144

#### 13.1. Содержание дисциплины

п/п	Наименование раздела дисциплины	Содержание раздела дисциплины	Реализация раздела дисциплины с помощью онлайн-курса, ЭУМК *
<b>1. Лекции</b>			
1.1	Философия .NET	Сравнение программирования на WIN32/C, C++/MFC, Visual Basic, Java, COM, Windows DNA. Возможности платформы .NET. Состав .NET Framework. Технология CLR, CTS, CLS.	
1.2	История создания языка C#. Структура приложения на языке C#.	Решение, проект(project), пространство имен(namespace), сборка(assembly). Консольный проект.	
1.3	Система типов в C#	Типы-значения, ссылочные типы(reference), указатели(pointer), тип void. Встроенные типы C#. Семантика присваивания. Преобразование типов в выражениях.	
1.4	Алфавит и лексемы языка.	Алфавит языка C#. Имена, ключевые слова, знаки операций, разделители, литералы. Комментарии.	
1.5	Переменные, операции, выражения.	Структура программы. Именованные константы. Операции и выражения. Консольный ввод-вывод.	
1.6	Операторы.	Оператор присваивания. Условный оператор.	

		Оператор выбора. Операторы цикла while, do, for. Операторы передачи управления goto, break, continue, return.
1.7	Классы.	Спецификаторы класса. Поля и константы класса. Методы класса. Параметры-значения, параметры-ссылки. Конструкторы. Свойства. Индексаторы. Статические методы класса. Деструкторы.
1.8	Функции.	Синтаксис описания методов. Список формальных параметров. Тело метода. Синтаксис и семантика вызова метода. Функции с побочным эффектом. Перегрузка методов.
1.9	Массивы.	Массивы одномерные, динамические, прямоугольные, многомерные, ступенчатые. Массивы объектов.
1.10	Символы и строки.	Массивы символов. Строки типа string. Форматирование строк.
1.11	Операции класса.	Унарные операции. Бинарные операции. Операции преобразования типа.
1.12	Отношения между классами	Отношения «является» и «имеет».
1.13	Наследование	Описание класса-потомка. Конструкторы родителей и потомков. Абстрактные классы.
1.14	Исключения.	Схема обработки исключений. Создание объектов Exception. Блоки try catch и finally, оператор throw.
1.15	Интерфейсы	Две стратегии реализации интерфейса. Преобразование к классу интерфейса. Проблемы множественного наследования. Реализация интерфейсов. Встроенные интерфейсы Comparable, Cloneable, Enumerale.
1.16	Контейнерные классы.	ArrayList, Stack, Queue, Hashtable.
1.17	Делегаты	Описание делегатов. Использование делегатов.
1.18	События	Определение события. Механизм события. Обработка событий.
1.19	Универсальность	Универсальные классы. Ограничения на универсальные классы. Универсальные структуры. Универсальные делегаты. Универсальные интерфейсы.
1.20	Сериализация	Понятие сериализации. Глубокая сериализация. Бинарная сериализация. XML-сериализация.
1.21	Паттерны программирования	Понятие паттерна проектирования. Порождающие паттерны. Структурные паттерны. Поведенческие паттерны. Обзор паттернов проектирования. Паттерн MVC.
1.22	Многопоточность	Класс Thread. Средства синхронизации потоков.
1.23	Небезопасный код	Управляемый и неуправляемый код. Указатели. Системные функции. Вызов неуправляемых функций.
1.24	Сборка мусора	Время жизни объектов. Корневой элемент приложения. Граф объектов. Поколения объектов. Выживание и переходы. Параллельная сборка мусора. Фоновая сборка мусора. Класс GC. Финализация
<b>2. Практические занятия</b>		
2.1	Алфавит и лексемы	Алфавит языка C#. Имена, ключевые слова,

	языка.	знаки операций, разделители, литералы. Комментарии.	
2.2	Переменные, операции, выражения.	Структура программы. Именованные константы. Операции и выражения. Консольный ввод-вывод.	
2.3	Операторы.	Оператор присваивания. Условный оператор. Оператор выбора. Операторы цикла while, do, for. Операторы передачи управления goto, break, continue, return.	
2.4	Классы.	Спецификаторы класса. Поля и константы класса. Методы класса. Параметры-значения, параметры-ссылки. Конструкторы. Свойства. Индексаторы. Статические методы класса. Деструкторы.	
2.5	Функции.	Синтаксис описания методов. Список формальных параметров. Тело метода. Синтаксис и семантика вызова метода. Функции с побочным эффектом. Перегрузка методов.	
2.6	Массивы.	Массивы одномерные, динамические, прямоугольные, многомерные, ступенчатые. Массивы объектов.	
2.7	Символы и строки.	Массивы символов. Строки типа string. Форматирование строк.	
2.8	Операции класса.	Унарные операции. Бинарные операции. Операции преобразования типа.	
2.9	Отношения между классами	Отношения «является» и «имеет».	
2.10	Наследование	Описание класса-потомка. Конструкторы родителей и потомков. Абстрактные классы.	
2.11	Исключения.	Схема обработки исключений. Создание объектов Exception. Блоки try catch и finally, оператор throw.	
2.12	Интерфейсы	Две стратегии реализации интерфейса. Преобразование к классу интерфейса. Проблемы множественного наследования. Реализация интерфейсов. Встроенные интерфейсы IComparable, ICloneable, Enumerable.	
2.13	Контейнерные классы.	ArrayList, Stack, Queue, Hashtable.	
2.14	Делегаты	Описание делегатов. Использование делегатов.	
2.15	События	Определение события. Механизм события. Обработка событий.	
2.16	Универсальность	Универсальные классы. Ограничения на универсальные классы. Универсальные структуры. Универсальные делегаты. Универсальные интерфейсы.	
2.17	Сериализация	Понятие сериализации. Глубокая сериализация. Бинарная сериализация. XML-сериализация.	
2.18	Паттерны программирования	Понятие паттерна проектирования. Порождающие паттерны. Структурные паттерны. Поведенческие паттерны. Обзор паттернов проектирования. Паттерн MVC.	
2.19	Создание графического интерфейса с помощью технологии Windows Forms	Обзор пространств имен Windows Forms. Стандартные и пользовательские элементы управления. Обработчики событий.	
2.20	Доступ к БД с использованием	Обзор пространств имен ADO. NET. Обзор классов для хранения данных и классов для	

	технологии ADO.NET	доступа к данным.	
2.21	Язык интегрированных запросов LINQ	Основы LINQ. Стандартные операторы запросов. Анонимные типы данных. Работа с коллекциями объектов. Использование LINQ с БД	
2.22	Многопоточность	Класс Thread. Средства синхронизации потоков.	
<b>3. Лабораторные занятия</b>			
3.1	Переменные, операции, выражения Операторы.	Знакомство со средой Visual Studio. Выполнение индивидуальных заданий простых алгоритмов.	<a href="http://www.lib.vsu.ru/elib/texts/method/vsu/m19-59.pdf">http://www.lib.vsu.ru/elib/texts/method/vsu/m19-59.pdf</a>  <a href="http://www.lib.vsu.ru/elib/texts/method/vsu/m19-209.pdf">http://www.lib.vsu.ru/elib/texts/method/vsu/m19-209.pdf</a>
3.2	Функции.	Выполнение индивидуальных заданий, использующих функции.	
3.3	Массивы.	Выполнение индивидуальных заданий, использующих массивы.	
3.4	Символы и строки.	Выполнение индивидуальных заданий, использующих строки.	
3.5	Классы. Операции класса.	Выполнение индивидуальных заданий, использующих классы.	
3.6	Наследование	Выполнение индивидуальных заданий, использующих иерархию классов и полиморфизм.	
3.7	Интерфейсы	Выполнение индивидуальных заданий, использующих иерархию классов и полиморфизм и интерфейсы.	
3.8	Делегаты. События. Универсальность.	Выполнение индивидуальных заданий, использующих универсальные классы, делегаты и события.	
3.9	Сериализация	Выполнение индивидуальных заданий, использующих сериализацию.	
3.10	Паттерны программирования	Выполнение индивидуальных заданий, использующих иерархию классов и полиморфизм и интерфейсы и паттерны проектирования.	
3.11	Многопоточность	Выполнение индивидуальных заданий, использующих многопоточность и синхронизацию потоков..	

\* заполняется, если отдельные разделы дисциплины изучаются с помощью онлайн-курса. В колонке Примечание необходимо указать название онлайн-курса или ЭУМК. В других случаях в ячейки ставятся прочерки.

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

№ п/п	Наименование темы (раздела) дисциплины					
		Лекции	Лабораторные	Самостоятельная работа	Контроль	Всего
1	Философия .NET	1		1	1	2
2	История создания языка C#. Структура приложения на языке C#.	1		1	1	3
3	Система типов в C#	1		1	1	3
4	Алфавит и лексемы языка.	1		1	1	3
5	Переменные, операции, выражения.	1	0,5	2	1	4,5
6	Операторы.	1	0,5	2	1	4,5
7	Классы.	2	1	2	1	6
8	Функции.	2	1,5	2	1	6,5
9	Массивы.	1	2,5	2	1	6,5
10	Символы и строки.	1	3	2	1	7

11	Операции класса.	1	1	2	1	5
12	Отношения между классами	1	1	2	2	6
13	Наследование	2	3	2	2	9
14	Исключения.	1	1	2	2	6
15	Интерфейсы	2	3	2	2	9
16	Контейнерные классы.	1	1	2	2	6
17	Делегаты	1	1	2	2	6
18	События	1	1	2	2	6
19	Универсальность	2	2	2	2	8
20	Сериализация	1	3	2	2	8
21	Паттерны программирования	2	3	2	2	9
22	Многопоточность	2	3	2	2	9
23	Небезопасный код	1		2	2	5
24	Сборка мусора	2		2	2	6
	Итого:	32	32	44	36	144

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

Работа с конспектами лекций, выполнение практических заданий для самостоятельной работы, тестов, выполнение лабораторных работ по дисциплине, использование рекомендованной литературы и методических материалов, в том числе находящихся в личном кабинете. 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.	Дэвис А. Асинхронное программирование в C# 5.0 [Электронный ресурс] : . — Электрон. дан. — Москва : ДМК Пресс, 2013. — 120 с. — Режим доступа: <a href="http://e.lanbook.com/books/element.php?pl1_id=9132">http://e.lanbook.com/books/element.php?pl1_id=9132</a> —
2.	Романенко, В. В. Объектно-ориентированное программирование : учебное пособие / В.В. Романенко ; Министерство образования и науки Российской Федерации ; Томский Государственный Университет Систем Управления и Радиоэлектроники (ТУСУР) .— Томск : Томский государственный университет систем управления и радиоэлектроники, 2014 .— 475 с. : ил. — Библиоар.: с. 442. — <a href="http://biblioclub.ru/">http://biblioclub.ru/</a> .— URL: <a href="http://biblioclub.ru/index.php?page=book&amp;id=480517">http://biblioclub.ru/index.php?page=book&amp;id=480517</a>

б) дополнительная литература:

№ п/п	Источник
1.	Либерти, Д. Программирование на C# / Д. Либерти ; Пер.с англ. С. Иноземцева .— 2-е изд. — СанктПетербург;Москва : Символ, 2003 .— 684 с
2.	Шилд Г. Полный справочник по C# / Г. Шилд . - Москва : Издательский дом «Вильямс», 2004 . —752 с.
3.	Рихтер Дж. CLR via C#. Программирование на платформе Microsoft .NET Framework 2.0 на языке C# / Дж. Рихтер . - Москва: Издательство «Русская редакция»; Санкт-Петербург.: Питер, 2007.
4.	Структуры и алгоритмы обработки данных : учебно-методическое пособие / Воронеж. гос. ун-т; сост.: Г.Э.Вошинская [и др.] .— Воронеж : ИПЦ ВГУ, 2012 .— 28 с.
5.	Троелсен, Э. C# и платформа .NET 3.0/ Э.Троелсен. - Санкт-Петербург : Питер, 2003 .— 795 с.
6.	Агапов В.П. Основы программирования на языке C# : учебное пособие / В.П. Агапов. - Москва, 2012. — Режим доступа: <a href="http://elibrary.ru">http://elibrary.ru</a>
7.	Снетков В.М. Практикум прикладного программирования на C# в среде VS.NET 2008 / В.М.Снетков Москва, 2010. — Режим доступа: <a href="http://elibrary.ru">http://elibrary.ru</a>
8.	Курченкова Т.В. Основы программирования на языке C# (учебно-методическое пособие для вузов) / Т.В. Курченкова, О.Ю. Лавлинская // Воронеж: Издательский дом ВГУ, 2021. — 82 с.



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

№ п/п	Ресурс
1.	Курс: Разработка приложений на С#. – Режим доступа: <a href="https://edu.vsu.ru/course/view.php?id=19794">https://edu.vsu.ru/course/view.php?id=19794</a>
2.	Электронный каталог Научной библиотеки Воронежского государственного университета. – Режим доступа: <a href="http://www.lib.vsu.ru/">http://www.lib.vsu.ru/</a>
3.	ЭБС «Издательство Лань». – Режим доступа: <a href="http://e.lanbook.com/">http://e.lanbook.com/</a>

\* Вначале указываются ЭБС, с которыми имеются договора у ВГУ, затем открытые электронно-образовательные ресурсы, онлайн-курсы, ЭУМК

**16. Перечень учебно-методического обеспечения для самостоятельной работы** (учебно-методические рекомендации, пособия, задачки, методические указания по выполнению практических (контрольных), курсовых работ и др.)

№ п/п	Источник
1.	Г.Э. Вошинская, Е.М. Лещенко Структуры и алгоритмы обработки данных. Часть 1. - Образовательный портал ВГУ: <a href="https://lib.vsu.ru/elib/texts/method/vsu/m19-59.pdf">https://lib.vsu.ru/elib/texts/method/vsu/m19-59.pdf</a> Режим доступа: личный кабинет студента
2.	Г.Э. Вошинская, Е.М. Лещенко Структуры и алгоритмы обработки данных. Часть 2. - Образовательный портал ВГУ: <a href="https://lib.vsu.ru/elib/texts/method/vsu/m19-209.pdf">https://lib.vsu.ru/elib/texts/method/vsu/m19-209.pdf</a> Режим доступа: личный кабинет студента

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

Дисциплина реализуется с применением электронного обучения и дистанционных образовательных технологий. Для организации занятий рекомендован онлайн-курс «Разработка приложений на С#», размещенный на платформе Электронного университета ВГУ (LMS moodle), а также Интернет-ресурсы, приведенные в п.15в.

**18. Материально-техническое обеспечение дисциплины:** (при использовании лабораторного оборудования указывать полный перечень, при большом количестве оборудования можно вынести данный раздел в приложение к рабочей программе)

Требования к аудиториям для проведения лекционных и практических занятий: наличие доски и средств письма на ней, оснащение проекционной техникой и компьютером.

Требования к аудиторному оборудованию для проведения лабораторных занятий: наличие компьютерных классов с современной компьютерной техникой и установленным программным обеспечением: ОС Windows, Microsoft Visual Studio 2010, Microsoft Visual Studio 2015, Microsoft Visual Studio 2019.

---

## 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.	Философия .NET	ПК-6	ПК-6.1	тесты, лабораторные работы
2.	История создания языка C#. Структура приложения на языке C#.			
3.	Система типов в C#			
4.	Алфавит и лексемы языка.			
5.	Переменные, операции, выражения.			
6.	Операторы.			
7.	Классы.			
8.	Функции.			
9.	Массивы.			
10.	Символы и строки.			
11.	Операции класса.			
12.	Отношения между классами			
13.	Наследование			
14.	Исключения.			
15.	Интерфейсы			
16.	Контейнерные классы.			
17.	Делегаты			
18.	События			
19.	Универсальность			
20.	Сериализация			
21.	Паттерны программирования			
22.	Многопоточность			
23.	Небезопасный код			
24.	Сборка мусора			
Промежуточная аттестация форма контроля – экзамен				<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.

#### 1. Тестовые задания

Курс «Программирование на C#»: <https://edu.vsu.ru/course/view.php?id=13150>

**Вопросы с одним верным вариантом ответа**

Критерий оценивания	Шкала оценок
Верный ответ	1 балл
Неверный ответ	0 баллов

**Вопросы с двумя одновременно верными вариантами ответа**

Критерий оценивания	Шкала оценок
Каждый верный ответ	+ 0,5 балла
Каждый неверный ответ	- 0,5 балла

Минимальная оценка – 0 баллов (в случае, если в сумме получается отрицательное число), максимальная – 1 балл (все верные ответы выбраны, все неверные не выбраны).

1. Какие из объявлений методов конфликтуют

```
class Overload
{
    public void Ovl() { //1
    }
    public int Ovl(int a) { return 0; //2
    }
    public int Ovl(int a, int b) { return 0; //3
    }
    public double Ovl(double a, double b) { return 0; //4
    }
    private void Ovl(int x) { //5
    }
    public void Ovl(ref int x) { //6
    }
}
```

Выберите один или несколько ответов:

- a. 1
- b. 2**
- c. 3
- d. 4
- e. 5**
- f. 6
- g. никакие

2. Какие из объявлений методов конфликтуют

```
class Overload
{ public void Ovl() { //1
  }
  public int Ovl(int a, int b) { return 0; //2
  }
  public void Ovl(out int x) { x = 1; //3
  }
  public double Ovl(double a, double b){ return 0; //4
  }
  public float Ovl(ref int x){ return 0; //5
  }
  private void Ovl(int x){ //6
  }
}
```

Выберите один или несколько ответов:

- a. 1
- b. 2
- c. 3**
- d. 4
- e. 5**
- f. 6
- g. никакие

3. Какие из объявлений методов конфликтуют

```
class Overload
{ public void Ovl(double x, params double[] a) { //1
  }
  public int Ovl(int y = 0) { return 0; //2
  }
  public void Ovl(int x, int y = 1) { //3
  }
  public double Ovl(double a, double b) { return 0; //4
  }
  public float Ovl(ref int x) { return 0; //5
  }
  private void Ovl(int x, ref int y) { //6
  }
}
```

Выберите один или несколько ответов:

- a. 1
- b. 2
- c. 3
- d. 4
- e. 5
- f. 6
- g. никакие**

1. Как работает приведение базовых и дочерних классов в C#?

- a. **Дочерние классы могут быть приведены к родительским, но не наоборот.**
- b. Родительские классы могут быть приведены к дочерним, но не наоборот.
- c. Родительские классы и дочерние могут быть приведены друг к другу.
- d. Приведение типов между дочерними и родительскими классами невозможно.

2. Из каких ключевых слов может состоять конструкция обработки исключений в C#?

- a. **try**
- b. get
- c. set
- d. **finally**
- e. **catch**
- f. handle

3. Что такое delegate в C#?

- a. Ключевое слово, которым обозначают динамическую переменную.
- b. **Тип данных, представляющий ссылку на метод.**
- c. Модификатор доступа классов.
- d. Ключевое слово, которым обозначают наследование.

4. Что такое Generic методы?

- a. Методы, не имеющие реализации в базовом классе.
- b. **Методы, зависящие от неопределенного типа.**
- c. Методы, не возвращающие значений.
- d. Методы, помеченные ключевым словом generic.

### Вопросы с кратким текстовым ответом

Критерий оценивания	Шкала оценок
Должен быть сформулирован ответ из указанных вариантов (один или несколько) или аналогичные по сути ответы с альтернативными терминами и определениями	2 балла
Неверный ответ	0 баллов

```
class Base
{
    public virtual void Who()
    {
        Console.WriteLine("Base");
    }
}
class Derived1 : Base
{
    public void Show()
    {
        Console.WriteLine("Derived");
    }
}
class Derived2 : Base
{
    public void Who()
    {
        Console.WriteLine("Derived2");
    }
}
```

1.

Что появится в результате выполнения следующего кода:

```
Base baseOb = new Base();
Derived1 dOb1 = new Derived1();
Derived2 dOb2 = new Derived2();
Base baseRef;
baseRef = baseOb; baseRef.Who();
baseRef = dOb1; baseRef.Who();
baseRef = dOb2; baseRef.Who();
```

Если в результате возникнет ошибка, в качестве ответа указывать слово Error

а. BaseBaseBase

```

class Base
{
    public virtual void Who()
    {
        Console.WriteLine("Base");
    }
}
class Derived1 : Base
{
    public void Show()
    {
        Console.WriteLine("Derived");
    }
}
class Derived2 : Base
{
    public override void Who()
    {
        Console.WriteLine("Derived2");
    }
}
}

```

Что появится в результате выполнения следующего кода:

```

Base baseOb = new Base();
Derived1 dOb1 = new Derived1();
Derived2 dOb2 = new Derived2();
Base baseRef;
baseRef = baseOb; baseRef.Who();
baseRef = dOb1;   baseRef.Who();
baseRef = dOb2;   baseRef.Who();

```

Если в результате возникнет ошибка, в качестве ответа указывать слово Error

**b.** BaseBaseDerived2

```

class Base
{
    public virtual void Who()
    {
        Console.WriteLine("Base");
    }
}
class Derived1 : Base
{
    public override void Who()
    {
        Console.WriteLine("Derived");
    }
}
class Derived2 : Base
{
    public override void Who()
    {
        Console.WriteLine("Derived2");
    }
}
}

```

3.

Что появится в результате выполнения следующего кода:

```

Base baseOb = new Base();
Derived1 dOb1 = new Derived1();
Derived2 dOb2 = new Derived2();
Base baseRef;
baseRef = baseOb; baseRef.Who();
baseRef = dOb1;   baseRef.Who();
baseRef = dOb2;   baseRef.Who();

```

Если в результате возникнет ошибка, в качестве ответа указывать слово Error  
с. BaseDerivedDerived2

### **Описание технологии проведения**

Текущая аттестация проводится на занятии одновременно во всей учебной группе в виде теста в электронной образовательной среде «Электронный университет ВГУ», адрес курса — <https://edu.vsu.ru/course/view.php?id=13150>, адрес теста текущей аттестации — <https://edu.vsu.ru/mod/quiz/view.php?id=1079972>. Тест составляется из материалов ФОСа, формируется системой автоматически путём добавления случайных вопросов, количество которых соответствует имеющимся образцам билетов. Большая часть вопросов проверяется автоматически, проверки преподавателем с ручным оцениванием требуют только отдельные вопросы, представленные в форме эссе. Ограничение по времени на каждую попытку — 1 час. Количество попыток – 1.

#### ***Требования к выполнению заданий (или шкалы и критерии оценивания)***

- проведение эксперимента, составление его описания и формулировка выводов, планирование отдельных стадий при наличии общего плана работы;
- обработка полученных результатов исследований с использованием стандартных методик;
- применение при обработке данных стандартного и оригинального программного обеспечения;
- оформление результатов лабораторных заданий в соответствии с действующими технологическими требованиями.

### **20.2 Промежуточная аттестация**

Промежуточная аттестация по дисциплине осуществляется с помощью следующих оценочных средств:

- Собеседование по вопросам к экзамену.

#### **Перечень вопросов к экзамену:**

1. Структура проекта Visual C#.
2. Структура программы.
3. Основные понятия ООП.
4. Алфавит. Ключевые слова. Идентификаторы.
5. Типы данных. (Типы-значения, типы-ссылки)
6. Литералы.
7. Форматирование вывода.
8. Переменные. Область действия и время существования переменных.
9. Преобразование и приведение типов данных.
10. Операции (арифметические, логические, сравнения, поразрядные, строковые)
11. Выражения, операции, операнды. Приоритет операций, ассоциативность.
12. Операция присваивания
13. Операторы языка. if-else, switch-case, do, for, foreach, while, break, continue, goto, return)
14. Пространства имен. Область видимости имен.
15. Массивы. Одномерные, многомерные, ступенчатые. Инициализация массивов.
16. Строки. Класс System.String. Методы работы со строками. Сравнение строк.
17. Понятие класса. Модификаторы полей, функций. Объявление класса. Оператор доступа к членам класса.



18. Понятие класса. Поля, методы, методы, сжатые до выражений. (7-6)
19. Понятие класса. Конструкторы, деструкторы, инициализация полей.
20. Управление доступом к членам класса.
21. Способы передачи аргументов в методы.
22. Использование переменного числа аргументов, именованные аргументы  
необязательные аргументы.
23. Перегрузка методов, конструкторов, ключевое слово `this`.
24. Инициализаторы объектов.
25. Статические члены класса. Статические классы.
26. Перегрузка операторов, логических операторов, `true`, `false`, операторов  
преобразования.
27. Индексаторы
28. Свойства. Методы доступа `get`, `set`.
29. Автореализуемые свойства.
30. Наследование. Доступ к членам класса, организация защищенного доступа.  
Конструкторы и наследование. Ключевое слово `sealed`.
31. Наследование и сокрытие имен.
32. Виртуальные методы и их переопределение.
33. Динамическая диспетчеризация.
34. Абстрактные методы и классы.
35. Класс `object` как универсальный тип данных.
36. Интерфейсы. Реализация интерфейсов, наследование. Выбор между  
интерфейсом и абстрактным классом
37. Интерфейсные ссылки, применение. Интерфейсные свойства,  
индексаторы. Использование явной реализации.
38. Встроенные интерфейсы `IEnumerable`, `ICloneable`, `IComparer`, `IComparable`.
39. Структуры. Объявление, члены структуры.
40. Перечисления. Объявление, инициализация, указание базового типа  
перечисления, применение перечислений.
41. Обобщения (универсальные шаблоны).
42. Обобщенные типы. Основные категории (свободный, связанный  
сконструированный, открытый, закрытый)
43. Обобщения. Ограниченные типы (`where`). Ограничения на базовый класс,  
на интерфейс, на конструктор, ссылочного типа и типа значения.
44. Правило применения нескольких ограничений. Связь между двумя  
параметрами типа с помощью ограничения.
45. Обобщенные структуры, методы (примеры)
46. Применение ограничений в обобщенных методах
47. Обработка исключительных ситуаций (`throw`, `try-catch`, `try-finally`, `try-catch-  
finally`)
48. Класс `Exception`. Пример использования производных классов исключений.
49. Потoki. Иерархия классов потоков `C#`. Назначение классов потоков.  
Основные методы.
50. Работа с файлами. Открытие и закрытие. Оператор `using`
51. Переадресация стандартных потоков.
52. Класс `File`. Основные методы
53. Директивы препроцессора (`#define`, `#if`, `#endif`, `#else`, `#elif`, `#undef`, `#error`,  
`#warning`, `#line`, `#region`, `#pragma`).
54. Делегаты.
55. Групповая адресация.
56. Ковариантность и контравариантность. Определения, примеры
57. Анонимные методы. Замыкания.
58. Лямбда-выражения.
59. События.

- 60. Паттерны проектирования. MVC, MVP
- 61. Порождающие паттерны
- 62. Структурные паттерны
- 63. Поведенческие паттерны
- 64. Динамическая идентификация типов. Операторы is, as, typeof
- 65. Рефлексия. Примеры, применение
- 66. Атрибуты
- 67. Сериализация, десериализация. Бинарная, XML, JSON сериализация.
- 68. Коллекции. ArrayList, List, Queue, Stack
- 69. Встроенные интерфейсы IEnumerable и IEnumerator
- 70. Сборка мусора. Класс System.GC. Финализируемые объекты. Создание деструкторов. Интерфейс IDisposable
- 71. Небезопасный код. Указатели, выделение памяти, фиксация, сравнение указателей.

**Критерии аттестации по итогам освоения дисциплины:**

отлично	Грамотно написаны и сданы задачи лабораторного практикума. Свободное владение материалом. Умение обосновать высказанное утверждение.
хорошо	Грамотно написаны и сданы задачи лабораторного практикума. Хорошее знание теоретического материала.
удовлетворительно	Сданы задачи лабораторного практикума. Недостаточное знание теоретического материала.
неудовлетворительно	Не сданы задачи лабораторного практикума, незнание теоретического материала.