

Минобрнауки России  
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»  
(ФГБОУ ВО «ВГУ»)**

**УТВЕРЖДАЮ**

Заведующий кафедрой  
программирования и информационных технологий



*проф. Махортов С.Д.*

10.03.2021 г.

**РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ**  
Б1.В.ДВ.04.01 Современные технологии программирования (Си)

**1. Код и наименование направления подготовки/специальности:**

02.03.01 Математика и компьютерные науки

**2. Профиль подготовки/специализация:**

Распределённые системы и искусственный интеллект

**3. Квалификация (степень) выпускника:**

Бакалавриат

**4. Форма обучения:**

Очная

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

Кафедра программирования и информационных технологий

**6. Составители программы:**

Хлебостроев В.Г., канд. физ.-мат. наук, доцент

**7. Рекомендована:**

НМС ФКН протокол № 5 от 10.03.2021 г.

**8. Учебный год: 2021-2022 и 2022-2023, семестры: 2, 3, 4**

## 9. Цели и задачи учебной дисциплины:

### Цель учебной дисциплины<sup>^</sup>

освоение студентами основных принципов современного объектно-ориентированного программирования на базе языков программирования C++ и C#.

### Основные задачи дисциплины:

- изучение синтаксиса и семантики языков программирования C++ и C#;
  - изучение принципов объектно-ориентированного программирования;
  - знакомство с наиболее распространенными алгоритмами и структурами данных;
- освоение технологии разработки программ в инструментальной среде Visual Studio.

## 10. Место учебной дисциплины в структуре ООП:

Является дисциплиной по выбору профессионального цикла; для ее успешного освоения необходимо знание основ информатики в пределах программы среднего общего образования. Данная дисциплина является предшествующей для специальных дисциплин. Б1.О.16 Компьютерная геометрия и геометрическое моделирование, Б1.В.13 Параллельное программирование, Б1.О.24 Основы компьютерных наук.

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

Код и название компетенции	Код и название индикатора компетенции	Знания, умения, навыки
ПК-1. Способен демонстрировать базовые знания математических и естественных наук, основ программирования и информационных технологий.	ПК-1.1. Обладает базовыми знаниями, полученными в области математических и (или) естественных наук, программирования и информационных технологий	Знать: основные концепции в области математических и естественных наук, принципы объектно-ориентированного программирования. Уметь: формулировать постановку задачи, находить и обосновывать алгоритм ее решения. Иметь: практический опыт научно-исследовательской деятельности в математике и информатике.
	ПК-1.2. Умеет находить, формулировать и решать стандартные задачи в собственной научно-исследовательской деятельности в математике и информатике	
	ПК-1.3. Имеет практический опыт научно-исследовательской деятельности в математике и информатике	
ПК-4. Способен использовать современные методы разработки и реализации конкретных алгоритмов математических	ПК-4.1. Знает современные методы разработки и реализации алгоритмов математических моделей на базе языков и пакетов прикладных программ моделирования	Знать: современные методы разработки и реализации алгоритмов математических моделей. Уметь: разрабатывать и реализовывать алгоритмы математических моделей на
	ПК-4.2. Умеет разрабатывать и реализовывать алгоритмы математических	

моделей на базе языков программирования и пакетов прикладных программ моделирования.	моделей на базе языков и пакетов прикладных программ моделирования	базе языков и пакетов прикладных программ. Иметь: практический опыт разработки и реализации алгоритмов.
	ПК-4.3. Имеет практический опыт разработки и реализации алгоритмов на базе языков и пакетов прикладных программ моделирования	
ПК-5. Способен участвовать в разработке технической документации программных продуктов и программных комплексов	ПК-5.1. Знает основные стандарты, нормы и правила разработки технической документации программных продуктов и программных комплексов	Знать: основные стандарты, нормы и правила разработки технической документации программных продуктов. Уметь: использовать их при подготовке технической документации программных продуктов. Иметь: практический опыт подготовки технической документации.
	ПК-5.2. Умеет использовать их при подготовке технической документации программных продуктов	
	ПК-5.3. Имеет практический опыт подготовки технической документации	
ПК-5.2	Умеет использовать их при подготовке технической документации программных продуктов	
ПК-5.3	Имеет практический опыт подготовки технической документации	

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

**Форма промежуточной аттестации:** Зачет с оценкой, Экзамен

**13. Трудоемкость по видам учебной работы**

Вид учебной работы	Трудоемкость			
	Всего	По семестрам		
		Семестр 2	Семестр 3	Семестр 4
Аудиторные занятия, в том числе:	128	64	32	32
лекции	64	32	16	16
практические занятия	32	16	16	
лабораторные занятия	32	16		16
Самостоятельная работа	196	116	40	40
Форма промежуточной аттестации (зачет – 4 час. / экзамен – 36 час.)	36			36
<b>Всего</b>	<b>360</b>	<b>180</b>	<b>72</b>	<b>108</b>

### 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	Пользовательские типы данных	Типы данных, задаваемые программистом, Структурный и перечислимый типы данных, способы использования переменных этих типов. Сходство и различие между массивами и структурами.	Онлайн-курс «Современные технологии программирования»*
1.2	Объектно-ориентированная парадигма программирования. Принцип инкапсуляции.	Объединение (инкапсуляция) в одной сущности данных и способов их обработки. Классы, как типы, инкапсулирующие данные и функции. Синтаксис описания классов, члены класса, уровни их доступа. Экземпляры классов (объекты), их объявление и инициализация. Конструкторы и деструкторы. Статические члены класса.	Онлайн-курс «Современные технологии программирования»
1.3	Перегрузка операций в классах.	Понятие перегрузки операций. Правила перегрузки для унарных и бинарных операция. Ограничения на перегрузку операций.	Онлайн-курс «Современные технологии программирования»
1.4	Динамические структуры данных: списки, стеки, очереди	Статические и динамические структуры данных, списки. Основные операции над списками. Реализация списков в виде классов с рекурсивным определением. Стеки и очереди, как динамические структуры со специальной дисциплиной доступа.	Онлайн-курс «Современные технологии программирования»
1.5	Принцип наследования классов. Уровни наследования.	Создание новых классов на основе существующих, принцип наследования. Изменение уровней доступа, членов класса при наследовании	Онлайн-курс «Современные технологии программирования»
1.6	Виртуальные методы и принцип полиморфизма	Перекрытие членов функций класса при наследовании, виртуальные и чисто виртуальные функции. Абстрактные классы. Полиморфное поведение экземпляров классов, условия его реализации.	Онлайн-курс «Современные технологии программирования»
1.7	Исключения, обработка исключений.	Исключения, как способ обработки программных ошибок. Синтаксис исключений. Генерация	Онлайн-курс «Современные технологии программирования»

		исключений. Стандартные исключения.	
1.8	Стандартная библиотека C++. Библиотека шаблонов STL.	Стандарты языка C++. Назначение и состав стандартной библиотеки C++. Пространство имен std и заголовочные файлы. Понятие шаблона и специализации, шаблоны функций и классов. Стандартная библиотека шаблонов STL.	Онлайн-курс «Современные технологии программирования»
1.9	Стандартная библиотека C++. Классы работы со строками.	C-строки и объекты класса string. Функции класса string. Создание и обработка строк средствами класса string.	Онлайн-курс «Современные технологии программирования»
1.10	Стандартная библиотека C++. Классы потоков ввода-вывода.	Понятие потока, потоки ввода и вывода. Буферизация потоков. Иерархия классов ввода-вывода. Консольные и файловые потоки. Форматирование ввода-вывода, манипуляторы.	Онлайн-курс «Современные технологии программирования»
1.11	Стандартная библиотека C++. Шаблонные классы коллекций.	Понятие коллекции. Контейнеры и итераторы. Типы контейнеров, контейнерные адаптеры. Функции-члены для работы с итераторами. Диапазонный цикл for. Алгоритмы библиотеки STL. Шаблоны контейнеров. Инициализация контейнеров и общие операции над ними.	Онлайн-курс «Современные технологии программирования»
1.12	Стандартная библиотека C++. Шаблонные классы array<> и vector <>.	Контейнеры статических и динамических массивов. Возможности шаблонных классов array<> и vector <>.	Онлайн-курс «Современные технологии программирования»
1.13	Стандартная библиотека C++. Шаблонные классы list<> и set <>.	Контейнеры списков и множеств. Возможности шаблонных классов list<> и set <>.	Онлайн-курс «Современные технологии программирования»
1.14	Язык программирования C# и платформа .Net	Архитектура платформы .Net, подсистема разработки и выполнения программ .Net Framework. Общеязыковая среда выполнения CLS и библиотека типов CTS. Пространства имен. Сборки, исполняемые сборки и библиотеки классов. Консольные и оконные приложения на языке C#, их структура.	Онлайн-курс «Создание приложений на языке C#»**
1.15	Основы языка C#: типы данных и операторы.	Типы данных в языке C#, Типы-значения и ссылочные типы. Иерархия типов. Операции и операторы языка C#.	Онлайн-курс «Создание приложений на языке C#»

1.16	Классы в языке C#, члены классов. Реализация принципа инкапсуляции.	Объявление и определение классов. Члены-данные и члены-функции. Члены-выражения. Принцип инкапсуляции, уровни видимости членов классов.	Онлайн-курс «Создание приложений на языке C#»
1.17	Интерфейсы, как средство объявления функциональности класса.	Интерфейсы, их назначение. Синтаксис описания и члены интерфейсов. Реализация (наследование) интерфейсов. Множественное наследование интерфейсов. Принцип неизменности интерфейсов.	Онлайн-курс «Создание приложений на языке C#»
1.18	Исключения, классы исключений. Поточный ввод-вывод. Классы потокового ввода-вывода.	Синтаксис конструкции обработки исключений. Иерархия классов исключений. Иерархия классов потокового ввода-вывода. Консольные и файловые потоки. Средства форматирования.	Онлайн-курс «Создание приложений на языке C#»
1.19	Массивы в языке C#, класс Array.	Объявление, инициализация и обработка массивов. Абстрактный базовый класс Array, свойства и статические методы класса.	Онлайн-курс «Создание приложений на языке C#»
1.20	Обработка строковых данных, классы String и StringBuilder.	Строки, как объекты. Классы String и StringBuilder, различие в поведении экземпляров этих классов. Объявление, инициализация и обработка строк.	Онлайн-курс «Создание приложений на языке C#»
1.21	Наследование и классов.	Отношение наследования, иерархия наследования, уровни наследования. Конструкторы классов-наследников. Абстрактные классы. Проблема приведения типов.	Онлайн-курс «Создание приложений на языке C#»
1.22	Полиморфизм, способы реализации полиморфного поведения объектов.	Принцип полиморфизма, реализации полиморфного поведения объектов в программах на языке C#. Виртуальные методы и их перекрытие в классах-наследниках.	Онлайн-курс «Создание приложений на языке C#»
1.23	Отношения между классами:	Различные виды отношений между классами и объектами: наследование, ассоциация (агрегация и композиция). Графическое изображение отношений на UML-диаграммах.	Онлайн-курс «Создание приложений на языке C#»
1.24	События, как средство взаимодействия между объектами разных классов.	Понятие события, событийное управление выполнением приложения. События, как члены-данные класса. Делегаты, объявление событий. Обработчики событий, подписка на события.	Онлайн-курс «Создание приложений на языке C#»

1.25	Классы коллекций. Обобщения, обобщенные (типизированные) коллекции.	Коллекции, проблема перечисления элементов коллекций. Нумераторы и итераторы. Универсальные и типизированные коллекции.	Онлайн-курс «Создание приложений на языке C#»
1.26	Стандартные классы коллекций из библиотеки классов.	Реализация основных структур данных в виде классов коллекций. Коллекции ArrayList, List<>, Set<>, HashTable<>, Dictionary<>.	Онлайн-курс «Создание приложений на языке C#»
1.27	Коллекции, как хранилища данных. Технология LINQ.	Язык запросов для выбора подмножеств элементов коллекций. Выражения запросов и методы запросов. Фильтрация данных и проецирование.	Онлайн-курс «Создание приложений на языке C#»
<b>2. Практические занятия</b>			
2.1	Объектно-ориентированная парадигма программирования. Принцип инкапсуляции.	Примеры объявления и определения классов. Разделение членов класса по уровням видимости. Объявление и инициализация объектов класса и указателей на объекты. Работа с объектами класса. Статические члены класса.	Онлайн-курс «Современные технологии программирования»
2.2	Динамические структуры данных: списки, стеки, очереди	Реализация классов, представляющих основные динамические структуры данных. Определение функций, выполняющих операции над динамическими структурами данных.	Онлайн-курс «Современные технологии программирования»
2.3	Принцип наследования классов. Уровни наследования.	Определение цепочки наследования из родительского, дочернего и внучатого классов. Анализ различных вариантов уровня наследования.	Онлайн-курс «Современные технологии программирования»
2.4	Исключения, обработка исключений	Рассмотрение различных способов контроля и обработки программных ошибок. Преимущества использования механизма исключений.	Онлайн-курс «Современные технологии программирования»
2.5	Стандартная библиотека C++. Классы работы со строками.	Анализ способов обработки строк, предоставляемых классом string. Присваивание значений строкам, способы модификации строк (добавление, удаление, замена подстрок и символов). Методы поиска и выделения подстрок и символов, способы доступа к элементам строки.	Онлайн-курс «Современные технологии программирования»

2.6	Стандартная библиотека C++. Классы потоков ввода- вывода.	Консольный ввод и проблема локализации. Различные способы форматирования при консольном выводе. Манипуляторы, их использование в потоках вывода. Файловый ввод и вывод, обработка исключений, генерируемых функциями классов файловых потоков.	Онлайн-курс «Современные технологии программирования»
2.7	Стандартная библиотека C++. Шаблонные классы коллекций.	Анализ примеров определения шаблона функции и шаблона класса. Примеры использования шаблонных классов и алгоритмов библиотеки STL.	Онлайн-курс «Современные технологии программирования»
2.8	Стандартная библиотека C++. Шаблонные классы list<> и set <>.	Примеры использования шаблонных классов list<> и set <> библиотеки STL.	Онлайн-курс «Современные технологии программирования»
2.9	Язык программирования C# и платформа .Net	Консольные и оконные приложения, их особенности и архитектура. Пример создания оконного приложения в среде программирования Visual Studio.	Онлайн-курс «Создание приложений на языке C#»
2.10	Основы языка C#: типы данных и операторы.	Система типов, типы-значения и ссылочные типы. Операции и операторы языка C#.	Онлайн-курс «Создание приложений на языке C#»
2.11	Классы в языке C#, члены классов. Реализация принципа инкапсуляции.	Примеры объявления и определения классов. Пространства имен. Разделение членов класса по уровням видимости. Объявление и инициализация объектов класса. Статические члены класса.	Онлайн-курс «Создание приложений на языке C#»
2.12	Интерфейсы, как средство объявления функциональности класса.	Примеры объявления и реализации интерфейсов. Члены интерфейсов. Наследование интерфейсов.	Онлайн-курс «Создание приложений на языке C#»
2.13	Исключения, классы исключений. Поточный ввод-вывод. Классы потокового ввода-вывода.	Байтовые потоки и классы-адаптеры. Текстовые и двоичные потоки. . Файловый ввод и вывод, обработка исключений, генерируемых методами классов файловых потоков.	Онлайн-курс «Создание приложений на языке C#»
2.14	Обработка строковых данных, классы String и StringBuilder.	Примеры обработки строковых данных с использованием средств классов String и StringBuilder. Различие в свойствах экземпляров этих классов	Онлайн-курс «Создание приложений на языке C#»
2.15	Отношения между классами:	Примеры отношений агрегации и композиции между объектами разных классов Графическое изображение отношений на UML-диаграммах.	Онлайн-курс «Создание приложений на языке C#»



2.16	Классы коллекций. Обобщения, обобщенные (типизированные) коллекции.	Интерфейсы IEnumerable и IEnumerator, использование IEnumerable вместо имени класса при объявлении объекта. Примеры определения нумератора и итераторов.	Онлайн-курс «Создание приложений на языке C#»
<b>3.Лабораторные занятия</b>			
3.1	Пользовательские типы данных	Разработка приложения, в котором используются переменные структурного и перечислимого типов	Онлайн-курс «Современные технологии программирования»
3.2	Объектно-ориентированная парадигма программирования. Принцип инкапсуляции.	Разработка приложения, содержащего объявление и определение некоторого класса и оперирующего объектами этого класса.	Онлайн-курс «Современные технологии программирования»
3.3	Динамические структуры данных: списки, стеки, очереди	Разработка приложения, содержащего определение одного из классов: списка, стека или очереди. Приложение должно демонстрировать работу с динамическими структурами данных.	Онлайн-курс «Современные технологии программирования»
3.4	Принцип наследования классов. Уровни наследования.	Разработка приложения, содержащего определения родительского и дочернего (дочерних) классов.	Онлайн-курс «Современные технологии программирования»
3.5	Виртуальные методы и принцип полиморфизма	Разработка приложения, демонстрирующего полиморфное поведение объектов	Онлайн-курс «Современные технологии программирования»
3.6	Стандартная библиотека C++. Классы работы со строками.	Разработка приложения, демонстрирующего различные способы создания и обработки строк с использованием функций класса string	Онлайн-курс «Современные технологии программирования»
3.7	Стандартная библиотека C++. Классы потоков ввода- вывода.	Разработка приложения, демонстрирующего использование файлов для ввода и вывода данных.	Онлайн-курс «Современные технологии программирования»
3.8	Стандартная библиотека C++. Шаблонные классы array<> и vector <>.	Разработка приложения, демонстрирующего обработку массивов с помощью классов-специализаций шаблонов array<> и vector <>.	Онлайн-курс «Современные технологии программирования»
3.9	Классы в языке C#, члены классов. Реализация принципа инкапсуляции.	Разработка приложения, содержащего объявление и определение некоторого класса и оперирующего объектами этого класса.	Онлайн-курс «Создание приложений на языке C#»

3.10	Массивы в языке С#, класс Array.	Разработка приложения, демонстрирующего обработку массивов с помощью статических методов класса Array.	Онлайн-курс «Создание приложений на языке С#»
3.11	Обработка строковых данных, классы String и StringBuilder.	Разработка приложения, демонстрирующего различные способы создания и обработки строк с использованием методов классов String и StringBuilder.	Онлайн-курс «Создание приложений на языке С#»
3.12	Наследование классов.	Разработка приложения, содержащего определения родительского и дочернего (дочерних) классов.	Онлайн-курс «Создание приложений на языке С#»
3.13	Полиморфизм, способы реализации полиморфного поведения объектов.	Разработка приложения, демонстрирующего полиморфное поведение объектов.	Онлайн-курс «Создание приложений на языке С#»
3.14	События, как средство взаимодействия между объектами разных классов.	Разработка приложения, содержащего определения двух классов, объекты которых взаимодействуют с использованием механизма событий.	Онлайн-курс «Создание приложений на языке С#»
3.15	Стандартные классы коллекций из библиотеки классов.	Разработка приложения, демонстрирующего работу с экземпляром (экземплярами) одного из стандартных классов-коллекций.	Онлайн-курс «Создание приложений на языке С#»
3.16	Коллекции, как хранилища данных. Технология LINQ.	Разработка приложения, демонстрирующего использование LINQ-запросов для извлечения подмножеств элементов из коллекций,	Онлайн-курс «Создание приложений на языке С#»

\* Онлайн-курс «Современные технологии программирования» размещен на портале «Электронный университет ВГУ» по адресу <https://edu.vsu.ru/course/view.php?id=4091>

\*\* Онлайн-курс «Создание приложений на языке С#» размещен на портале «Электронный университет ВГУ» по адресу <https://edu.vsu.ru/course/view.php?id=4091>

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

№ п/п	Наименование темы (раздела)	Виды занятий (часов)				
		Лекции	Практические	Лабораторные.	Самостоятельная работа	Всего
1.1	Пользовательские типы данных	2	–	2	6	10
1.2	Объектно-ориентированная парадигма	4	2	2	10	18

	программирования. Принцип инкапсуляции.					
1.3	Перегрузка операций в классах.	2	–	–	6	8
1.4	Динамические структуры данных: списки, стеки, очереди	4	2	2	12	20
1.5	Принцип наследования классов. Уровни наследования.	2	2	2	8	14
1.6	Виртуальные методы и принцип полиморфизма	2	–	2	10	14
1.7	Исключения, обработка исключений.	2	2	–	6	10
1.8	Стандартная библиотека C++. Библиотека шаблонов STL.	2	–	–	8	10
1.9	Стандартная библиотека C++. Классы работы со строками.	2	2	2	12	18
1.10	Стандартная библиотека C++. Классы потоков ввода-вывода.	4	2	2	10	18
1.11	Стандартная библиотека C++. Шаблонные классы коллекций.	2	2	–	8	12
1.12	Стандартная библиотека C++. Шаблонные классы array<> и vector <>.	2	–	2	10	14
1.13	Стандартная библиотека C++. Шаблонные классы list<> и set <>.	2	2	–	10	14
1.14	Язык программирования C# и платформа .Net	2	2	–	4	8
1.15	Основы языка C#: типы данных и операторы.	2	2	–	4	8
1.16	Классы в языке C#, члены классов. Реализация принципа инкапсуляции.	2	2	2	6	12

1.17	Интерфейсы, как средство объявления функциональности класса.	2	2	–	6	10
1.18	Исключения, классы исключений. Поточковый ввод-вывод. Классы потокового ввода-вывода.	2	2	–	6	10
1.19	Массивы в языке C#, класс Array.	2	–	2	4	10
1.20	Обработка строковых данных, классы String и StringBuilder.	2	2	2	6	10
1.21	Наследование классов.	2	–	2	6	10
1.22	Полиморфизм, способы реализации полиморфного поведения объектов.	2	–	2	6	10
1.23	Отношения между классами:	2	2	–	6	10
1.24	События, как средство взаимодействия между объектами разных классов.	2	–	2	8	12
1.25	Классы коллекций. Обобщения, обобщенные (типизированные) коллекции.	2	2	–	6	10
1.26	Стандартные классы коллекций из библиотеки классов.	4	–	2	6	12
1.27	Коллекции, как хранилища данных. Технология LINQ.	4	–	2	6	12
Экзамен		–	–	–	–	36
<b>Всего</b>		<b>64</b>	<b>32</b>	<b>32</b>	<b>196</b>	<b>360</b>

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

Работа с конспектами лекций и презентационными материалами; выполнение практических заданий, анализ образцов приложений; выполнение лабораторных заданий; подготовка к текущим аттестациям.

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

№ п/п	Источник
1	<i>Тюкачев Н.А. С# Основы программирования : учебное пособие / Н.А. Тюкачев, В.Г. Хлебостроев.— 2-е изд., испр. и доп. – СПб. : Издательство «Лань», 2017 .— 212 с.</i>
2	<i>Махортов С.Д. Программирование на С# : учебное пособие / С.Д. Махортов ; Воронеж. гос. ун-т .— Воронеж : ЛОП ВГУ, 2006 .— 51 с</i>
3	<i>Прата С. Язык программирования С++ : лекции и упражнения / Стивен Прата ; пер. с англ. Д.Я. Иваненко [и др.]; под ред. Ю.Н. Артеменко .— 5-е изд. — М. ; СПб ; Киев : Вильямс, 2007 .— 1181 с. : ил .— Парал. тит. л. англ.</i>
4	<i>Березин Б.И. Начальный курс С и С++ / Б.И. Березин, С.Б. Березин .— М. : Диалог-МИФИ, 2007 .— 288 с. : ил., табл. — Библиогр.: С. 285 .— ISBN 5-86404-075-4</i>

**б) дополнительная литература:**

№ п/п	Источник
5	<i>Языки программирования и методы трансляции : учебное пособие для студентов вузов, обучающихся по направлению подготовки "Прикладная математика и информатика" / С.З. Свердлов .— СПб [и др.] : Питер, 2007 .— 637 с.</i>

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

№ п/п	Источник
6	<i>В.А. Буллиг Основы программирования на С#. (<a href="https://intuit.ru/studies/courses/2247/18/info">https://intuit.ru/studies/courses/2247/18/info</a>)</i>
7	<i><a href="http://www.lib.vsu.ru">www.lib.vsu.ru</a> - ЗНБ ВГУ</i>

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

№ п/п	Источник
8	<i>Руководство по языку программирования С++. – Интернет-ресурс.. (<a href="https://metanit.com/cpp/tutorial/">https://metanit.com/cpp/tutorial/</a>)</i>
9	<i>Полное руководство по языку программирования С# 9.0 и платформе .NET 5. – Интернет-ресурс. (<a href="https://metanit.com/sharp/tutorial/">https://metanit.com/sharp/tutorial/</a>)</i>

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

Дистанционные образовательные технологии (ДОТ), электронное обучение (ЭО), смешанное обучение.

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

Мультимедийная лекционная аудитория, компьютерный класс, соответствующее программное обеспечение

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

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

№ п/п	Разделы дисциплины (модули)	Код компетенции	Код индикатора	Оценочные средства для текущей аттестации
1	1.2 Объектно-ориентированная парадигма программирования. Классы, члены классов. Принцип инкапсуляции. 1.4 Статические и динамические структуры данных: 1.5 Принцип наследования классов. 1.6 Виртуальные методы классов и принцип полиморфизма 1.7 Исключения, обработка исключений 1.8 Обобщенное программирование. Понятие шаблона и специализации, 1.10 Понятие потока ввода-вывода. Буферизация потоков. Иерархия классов ввода-вывода. 1.14 Архитектура платформы .Net, подсистема разработки и выполнения программ .Net Framework. 1.17 Интерфейсы, их назначение. 1.24 Понятие события, событийное управление выполнением приложения. 1.25 Коллекции, перечисление элементов коллекций. Нумераторы и итераторы.	ПК-1	ПК-1.1	Вопросы к текущим аттестациям размещены в онлайн-курсах «Современные технологии программирования» и «Создание приложений на языке C#»
2	1.1 Структурный и перечислимый типы данных, способы использования переменных этих типов. 1.4 Основные операции над списками. Реализация списков в виде классов с рекурсивным определением. 1.5 Создание новых классов на основе существующих, принцип наследования. 1.9 Создание и обработка строк средствами класса string. 1.11 Алгоритмы библиотеки STL. 1.19 Абстрактный базовый класс Array, свойства и статические методы класса. 1.20 Обработка строковых данных, классы String и StringBuilder. 1.27 Язык запросов для выбора подмножеств элементов коллекций.	ПК-4	ПК-4.1	Вопросы к текущим аттестациям размещены в онлайн-курсах «Современные технологии программирования» и «Создание приложений на языке C#»
3	1.2 Синтаксис описания классов, члены класса, уровни их доступа.	ПК-5	ПК-5.1	Вопросы к текущим аттестациям размещены в онлайн-курсах

№ п/п	Разделы дисциплины (модули)	Код компетенции	Код индикатора	Оценочные средства для текущей аттестации
	<p>Экземпляры классов (объекты), их объявление и инициализация.</p> <p>1.7 Исключения, как способ обработки программных ошибок. Синтаксис исключений.</p> <p>1.8 Стандарты языка C++.</p> <p>Назначение и состав стандартной библиотеки C++.</p> <p>1.14 Консольные и оконные приложения на языке C#, их структура.</p> <p>1.17 Синтаксис описания и члены интерфейсов. Реализация (наследование) интерфейсов.</p> <p>1.23 Графическое изображение отношений между классами и объектами на UML-диаграммах.</p>			«Современные технологии программирования» и «Создание приложений на языке C#»
4	<p>1.2 Синтаксис описания классов, члены класса, уровни их доступа. Экземпляры классов (объекты), их объявление и инициализация.</p> <p>1.7 Исключения, как способ обработки программных ошибок. Синтаксис исключений.</p> <p>1.8 Стандарты языка C++.</p> <p>Назначение и состав стандартной библиотеки C++.</p> <p>1.14 Консольные и оконные приложения на языке C#, их структура.</p> <p>1.17 Синтаксис описания и члены интерфейсов. Реализация (наследование) интерфейсов.</p> <p>1.23 Графическое изображение отношений между классами и объектами на UML-диаграммах.</p>	ПК-5	ПК-5.2	Вопросы к текущим аттестациям размещены в онлайн-курсах «Современные технологии программирования» и «Создание приложений на языке C#»
5	<p>1.2 Синтаксис описания классов, члены класса, уровни их доступа. Экземпляры классов (объекты), их объявление и инициализация.</p> <p>1.7 Исключения, как способ обработки программных ошибок. Синтаксис исключений.</p> <p>1.8 Стандарты языка C++.</p> <p>Назначение и состав стандартной библиотеки C++.</p> <p>1.14 Консольные и оконные приложения на языке C#, их структура.</p> <p>1.17 Синтаксис описания и члены интерфейсов. Реализация (наследование) интерфейсов.</p> <p>1.23 Графическое изображение отношений между классами и объектами на UML-диаграммах.</p>	ПК-5	ПК-5.3	Вопросы к текущим аттестациям размещены в онлайн-курсах «Современные технологии программирования» и «Создание приложений на языке C#»
6	1.1 Структурный и перечислимый типы данных, способы	ПК-4	ПК-4.2	Вопросы к текущим аттестациям размещены в онлайн-курсах

№ п/п	Разделы дисциплины (модули)	Код компетенции	Код индикатора	Оценочные средства для текущей аттестации
	<p>использования переменных этих типов.</p> <p>1.4 Основные операции над списками. Реализация списков в виде классов с рекурсивным определением.</p> <p>1.5 Создание новых классов на основе существующих, принцип наследования.</p> <p>1.9 Создание и обработка строк средствами класса string.</p> <p>1.11 Алгоритмы библиотеки STL.</p> <p>1.19 Абстрактный базовый класс Array, свойства и статические методы класса.</p> <p>1.20 Обработка строковых данных, классы String и StringBuilder.</p> <p>1.27 Язык запросов для выбора подмножеств элементов коллекций.</p>			<p>«Современные технологии программирования» и «Создание приложений на языке C#»</p>
7	<p>1.1 Структурный и перечислимый типы данных, способы использования переменных этих типов.</p> <p>1.4 Основные операции над списками. Реализация списков в виде классов с рекурсивным определением.</p> <p>1.5 Создание новых классов на основе существующих, принцип наследования.</p> <p>1.9 Создание и обработка строк средствами класса string.</p> <p>1.11 Алгоритмы библиотеки STL.</p> <p>1.19 Абстрактный базовый класс Array, свойства и статические методы класса.</p> <p>1.20 Обработка строковых данных, классы String и StringBuilder.</p> <p>1.27 Язык запросов для выбора подмножеств элементов коллекций.</p>	ПК-4	ПК-4.3	<p>Вопросы к текущим аттестациям размещены в онлайн-курсах «Современные технологии программирования» и «Создание приложений на языке C#»</p>
8	<p>1.2 Объектно-ориентированная парадигма программирования. Классы, члены классов. Принцип инкапсуляции.</p> <p>1.4 Статические и динамические структуры данных:</p> <p>1.5 Принцип наследования классов.</p> <p>1.6 Виртуальные методы классов и принцип полиморфизма</p> <p>1.7 Исключения, обработка исключений</p>	ПК-1	ПК-1.2	<p>Вопросы к текущим аттестациям размещены в онлайн-курсах «Современные технологии программирования» и «Создание приложений на языке C#»</p>



№ п/п	Разделы дисциплины (модули)	Код компетенции	Код индикатора	Оценочные средства для текущей аттестации
	<p>1.8 Обобщенное программирование. Понятие шаблона и специализации,</p> <p>1.10 Понятие потока ввода-вывода. Буферизация потоков.</p> <p>Иерархия классов ввода-вывода.</p> <p>1.14 Архитектура платформы .Net, подсистема разработки и выполнения программ .Net Framework.</p> <p>1.17 Интерфейсы, их назначение.</p> <p>1.24 Понятие события, событийное управление выполнением приложения.</p> <p>1.25 Коллекции, перечисление элементов коллекций.</p> <p>Нумераторы и итераторы.</p>			
9	<p>1.2 Объектно-ориентированная парадигма программирования. Классы, члены классов. Принцип инкапсуляции.</p> <p>1.4 Статические и динамические структуры данных:</p> <p>1.5 Принцип наследования классов.</p> <p>1.6 Виртуальные методы классов и принцип полиморфизма</p> <p>1.7 Исключения, обработка исключений</p> <p>1.8 Обобщенное программирование. Понятие шаблона и специализации,</p> <p>1.10 Понятие потока ввода-вывода. Буферизация потоков.</p> <p>Иерархия классов ввода-вывода.</p> <p>1.14 Архитектура платформы .Net, подсистема разработки и выполнения программ .Net Framework.</p> <p>1.17 Интерфейсы, их назначение.</p> <p>1.24 Понятие события, событийное управление выполнением приложения.</p> <p>1.25 Коллекции, перечисление элементов коллекций.</p> <p>Нумераторы и итераторы.</p>	ПК-1	ПК-1.3	Вопросы к текущим аттестациям размещены в онлайн-курсах «Современные технологии программирования» и «Создание приложений на языке C#»
<p>Промежуточная аттестация Форма контроля - Зачет с оценкой, Экзамен</p>				Вопросы к зачетам и экзамену размещены в онлайн-курсах «Современные технологии программирования» и «Создание приложений на языке C#»

## 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: рубежные аттестации лабораторные работы. Перечень заданий для лабораторных

работ соответствует темам занятий. Решение каждого задания должно быть доведено до компьютерной реализации

Перечень заданий для лабораторных работ размещен на портале «Электронный университет ВГУ»:

- по адресу <https://edu.vsu.ru/course/view.php?id=4091> для 2-го семестра;
- по адресу <https://edu.vsu.ru/course/view.php?id=4091> для 3 и 4-го семестров.

В течение 2-го семестра студент обязан сделать 5 заданий, в течение 3-го семестра – 5 заданий, в течение 4-го семестра – 4 задания. Студент, не сделавший нужное число заданий к зачету не допускается.

За каждое задание выставляется оценка по 50-бальной шкале. За отчет в промежуточной аттестации выставляется средний бал из полученных в течение месяца. Если к месячному отчету не сдано необходимое количество заданий на оценку более 25 баллов, то студент к отчету автоматически не допускается (оценка 0 баллов). При сдаче задачи не в срок оценка снижается на 5-10 баллов.

Студентам, сдавшим вовремя промежуточные аттестации на оценку более 30 баллов, в конце семестра зачет может быть выставлен автоматом.

## 20.2 Промежуточная аттестация

№ п/п	Перечень вопросов к зачету 2-го семестра
1.	Типы данных, задаваемые пользователями. Структурный тип данных, объявление структур.
2.	Структурные переменные, их отличие от массивов. Объявление и инициализация структурных переменных.
3.	Размер структурных переменных, прямой доступ к их полям.
4.	Перечисления, как пользовательский тип данных. Объявление перечислений и переменных перечислимого типа.
5.	Класс как тип данных. Поля и методы класса. Принцип инкапсуляции. Статус видимости элементов класса, управление видимостью.
6.	Конструкторы и деструкторы, создание объектов и инициализация полей.
7.	Создание и использование экземпляров класса (объектов).
8.	Методы доступа к закрытым полям классов.
9.	Статические члены класса, способ доступа к этим членам.
10.	Различие между объявлением классов с помощью ключевого слова <code>class</code> и ключевого слова <code>struct</code> .
11.	Создание классов в среде Visual Studio. Заголовочные файлы и файлы описания. Разрешение области видимости метода .
12.	Списки, их создание и модификация.
13.	Понятие дисциплины доступа. Прямой, последовательный и специальный доступ к структурам данных.
14.	Стек, как структура данных с дисциплиной доступа LIFO. Алгоритмы, реализации операций над стеком.
15.	Очередь, как структура данных с дисциплиной доступа FIFO. Алгоритмы, реализации операций над очередью.
16.	Принцип наследования. Наследование полей и методов. Уровень доступа <code>protected</code> (защищенный).
17.	Уровни наследования. Изменение уровней доступа полей и методов при наследовании.
18.	Перегрузка операций. Ограничения на выполнение перегрузки операций.
19.	Перегрузка унарных и бинарных операций.
20.	Строковый тип данных в языке C++. Стандартный класс <code>string</code> , строки, как объекты этого класса. Конструкторы класса <code>string</code> .
21.	Стандартный класс <code>string</code> . Присваивание значений строкам, способы модификации строк (добавление, удаление, замена подстрок и символов).
22.	Стандартный класс <code>string</code> . Методы поиска и выделения подстрок и символов, способы доступа к элементам строки.
23.	Файловые потоки. Создание файловых потоков, режимы открытия файлов.
24.	Текстовые файлы, операции чтения и записи для текстовых файлов.

25.	Бинарные файлы. Прямой доступ к записям с помощью указателей текущей позиции. Операции чтения и записи для бинарных файлов.
26.	Варианты изменения методов в классах-наследниках: перегрузка, переопределение, скрытие.
27.	Проблема вызова переопределенных и скрытых методов. Статическое и динамическое связывание.
28.	Реализация динамического связывания, виртуальные методы. Полиморфизм, как принцип объектно-ориентированного программирования.
29.	Понятие обобщенного программирования. Шаблоны и специализации шаблонов. Шаблоны функций и шаблоны классов.
30.	Библиотека стандартных шаблонов, основные компоненты STL.
31.	Контейнеры, категории контейнеров, методы контейнеров.
32.	Итераторы, как обобщение указателей. Категории итераторов, их иерархия.
33.	Объявление итераторов, операции над итераторами.
34.	Контейнеры <code>array&lt;&gt;</code> и <code>vector&lt;&gt;</code> , их сравнение, методы этих контейнеров.
35.	Контейнеры <code>list&lt;&gt;</code> и <code>set&lt;&gt;</code> , методы этих контейнеров.
36.	Алгоритмы библиотеки STL как дополнительные способы обработки векторов.

№ п/п	Перечень вопросов к зачету 3-го семестра
1.	Язык C# и платформа .NET. Система типов CTS и типы данных в языке C#. Типы-значения и ссылочные типы.
2.	Операции в языке C#. Характеристики операций (арность, направление ассоциативности, уровень приоритета). Примеры вычисления значений выражений.
3.	Оператор выражения в C#. Правила вычисления значений выражений. Примеры вычисления значений выражений.
4.	Условный оператор и оператор выбора. Примеры их использования.
5.	Операторы цикла, их синтаксис и семантика. Примеры их использования.
6.	Консольный вывод, методы вывода. Форматирование строк при выводе. Консольный ввод, методы ввода. Способы преобразования типа при вводе.
7.	Объявление и инициализация одномерных массивов. Объектные свойства массивов. Класс <code>System.Array</code> , примеры использования его свойств и методов.
8.	Объявление и инициализация двумерных массивов. Ломаные массивы. Класс <code>System.Array</code> , примеры использования его свойств и методов.
9.	Строки в языке C#. Класс <code>System.String</code> и его методы. Примеры использования методов класса <code>System.String</code> .
10.	Класс <code>StringBuilder</code> и его методы. Область применения класса <code>StringBuilder</code> .
11.	Типы структура ( <code>struct</code> ) и перечисление ( <code>enum</code> ). Синтаксис объявления этих типов и их переменных. Примеры использования структур и перечислений.
12.	Классы в языке C#. Объявление и описание класса. Сходство и различие между классами и структурами. Класс <code>Object</code> , его методы.
13.	Уровни доступа, модификаторы доступа. Уровни принадлежности членов класса, статические члены. Примеры объявления классов.
14.	Члены-данные класса. Объявление полей, констант и событий. Принцип инкапсуляции. Примеры объявления констант, полей и событий.
15.	Члены-функции класса. Объявление методов класса. Параметры методов. Передача параметров по ссылке и по значению.
16.	Конструктор и деструктор класса, их назначение. Способы инициализации членов-данных.
17.	Свойства класса, их описание и использование. Автоматически создаваемые свойства, свойства-выражения. Примеры определения свойств.
18.	Индексаторы, их определение и использование. Примеры определения и использования индексаторов.
19.	Перегрузка операций для объектов класса. Правила выполнения перегрузки операций. Синтаксис объявления членов-функций, перегружающих операции. Примеры перегрузки операций.
20.	Отношение между классами «клиент-поставщик». Способы размещения определения класса-поставщика относительно определения класса-клиента. Примеры отношения «клиент-поставщик».

21.	Отношение наследования, синтаксис объявления класса-потомка. Создание экземпляров класса-потомка, взаимодействие конструкторов родителя и потомка. Абстрактные классы. Примеры отношения «родитель-потомок».
-----	--

№ п/п	Перечень вопросов к экзамену
1.	Классы в языке C#. Объявление и описание класса. Сходство и различие между классами и структурами. Класс Object, его методы.
2.	Отношение наследования, синтаксис объявления класса-потомка. Создание экземпляров класса-потомка, взаимодействие конструкторов родителя и потомка. Абстрактные классы. Примеры отношения «родитель-потомок».
3.	Три способа изменения методов класса-родителя в классе-потомке. Статическое и динамическое связывание. Полиморфизм, механизмы реализации полиморфизма. Примеры полиморфного поведения объектов.
4.	Интерфейсы, объявления интерфейсов. Члены интерфейсов. Реализация интерфейсов. Принцип неизменности интерфейсов. Системные интерфейсы .Net Framework. Примеры определения и реализации интерфейсов.
5.	Обобщения, как способ параметризации типов. Обобщенные классы, способ их объявления. Ограничения для типов-параметров. Создание экземпляров обобщенных классов. Примеры объявления обобщенных классов.
6.	Обобщенные интерфейсы. Определение и реализация обобщенных интерфейсов. Примеры определения обобщенных интерфейсов. Обобщенные интерфейсы пространства имен System.Collections.Generic.
7.	. Понятие коллекции. Интерфейс IEnumerable и его реализация. Нумераторы и итераторы. Примеры их определения.
8.	Классы обобщенных коллекций. Делегаты, их объявление и применение в качестве параметров методов обобщенных коллекций. Примеры использования делегатов в качестве параметров.
9.	Списки, методы класса обобщенной коллекции List<T>. Свойства и методы класса List<T>. Примеры использования класса List<T>.
10.	Класс обобщенной коллекций LinkedList<T>. Интерфейсы, реализуемые этим классом. Свойства и методы класса, примеры его использования.
11.	Классы обобщенных коллекций Stack<T> и Queue<T>. Методы и свойства этих классов. Примеры использования этих классов.
12.	Множества. Класс обобщенной коллекции Hashtable<T>. Интерфейсы, реализуемые этим классом. Свойства и методы класса, примеры его использования.
13.	Словари. Класс обобщенной коллекции Dictionary<TK, TV>. Интерфейсы, реализуемые этим классом. Свойства и методы класса, примеры его использования.
14.	Коллекции, как хранилища данных. Проблема поиска и выбора информации из коллекций. Технология LINQ и средства построения запросов.
15.	Технология LINQ и средства построения запросов. Понятие последовательности. Выражения запросов и правила их построения. Отложенное выполнение выражений запросов.
16.	Стандартные операторы (методы) запросов. Понятие методов расширения. Создание и использование методов расширения.
17.	Анонимные методы и лямбда-выражения, их использование в качестве аргументов при вызовах стандартных операторов запросов.
18.	Операция фильтрации (метод Where), способы описания условия (предиката) фильтрации.
19.	Операции упорядочения (метод OrderBy и др.), способы сортировки, многоуровневая сортировка.
20.	Операция проецирования (метод Select) и анонимные типы.
21.	Операция группировки, группы, как элементы выходной последовательности метода GroupBy
22.	Методы преобразования последовательностей в коллекции, немедленное выполнение запросов.
23.	События, как средство взаимодействия объектов. Объявление события как члена класса. Генерация события, способы генерации. Метод Invoke. Оператор условного null

24.	Подписка на событие, обработчики событий. Создание цепочки обработчиков. Добавление обработчиков в цепочку и удаление из цепочки.
25.	Виды программных ошибок. Исключения, базовая синтаксическая структура для обработки исключений. Выполнение базовой структуры обработки исключений.
26.	Виды стандартных исключений. Генерация программных исключений. Вложенные структуры обработки исключений.
27.	Входные и выходные потоки данных. Организация работы с файлами в C#. Байтовые потоки. Класс Stream и его наследники. Файловые байтовые потоки. Последовательный и прямой доступ к файлам.
28.	Символьные файловые потоки. Создание символьных файловых потоков. Классы StreamReader и StreamWriter. Запись и чтение символьных данных. Возможные типы исключений при работе с символьными потоками.
29.	Двоичные файловые потоки, их создание, особенности применения. Классы BinaryReader и BinaryWriter. Запись и чтение данных различного типа. Возможные типы исключений при работе с двоичными потоками.