

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

УТВЕРЖДАЮ

Заведующий кафедрой программного обеспечения  
и администрирования информационных систем



Артемов М. А.

31.08.2019 г.

**РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ**  
Б1.Б.22 Язык программирования C#

**1. Шифр и наименование направления подготовки:**

02.03.03 Математическое обеспечение и администрирование информационных систем

**2. Профиль подготовки:** Информационные системы и базы данных

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

**4. Форма обучения:** очная

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

Программного обеспечения и администрирования информационных систем

**6. Составители программы:** Вошинская Г.Э. ст. преп.

**7. Рекомендована:** НМС факультета ПММ протокол № 10 от 15.06.2019 г.

**8. Учебный год:** 2019/2020

**Семестр(ы):** 3,4

**9. Цели и задачи учебной дисциплины:** развитие навыков объектно-ориентированного и компонентно-ориентированного программирования, знакомство с платформой .NET, практическое применение методов объектно-ориентированного анализа, изучение технологии разработки сложных систем, формирование навыков самостоятельного изучения и использования современных программных средств.

**10. Место учебной дисциплины в структуре ООП:** профессиональный цикл

(базовая часть);

требования к входным знаниям: программирование, структуры и алгоритмы обработки данных. Дисциплина является предшествующей для следующих:

- операционные системы и оболочки;
- Java;
- комбинаторные алгоритмы;
- программирование с использованием технологий MS.NET;
- использование фреймворков при разработке приложений;
- распределенные системы;
- технология разработки программного обеспечения;
- проектирование информационных систем;
- компьютерная графика;
- проектирование пользовательских интерфейсов;
- сервис-ориентированные архитектуры;
- разработка многопоточных приложений;
- шаблоны проектирования;
- Web-программирование;
- написание курсовых и дипломных проектов.

#### 11. Компетенции обучающегося, формируемые в результате освоения дисциплины:

Компетенция		Планируемые результаты обучения
Код	Название	
ОПК-7	способность использовать знания основных концептуальных положений функционального, логического, объектно-ориентированного и визуального направлений программирования, методов, способов и средств разработки программ в рамках этих направлений	Знать: основные этапы компьютерного решения задач; понятие алгоритма и структуры управления; структуры данных; основные требования методологии объектноориентированного программирования как технологической основы разработки качественных программных компонентов. Уметь: применять требования методологии объектноориентированного программирования при разработке программ; разрабатывать и записывать на языке программирования алгоритмы решения классических задач информатики; реализовывать технологию проектирования сверху вниз; выбрать оптимальную структуру для представления данных Владеть: навыками проектирования, реализации эффективных программ
ОПК-8	способность использовать знания методов проектирования и производства программного продукта, принципов построения, структуры и приемов работы с инструментальными средствами, поддерживающими создание программного обеспечения	Знать: основные этапы компьютерного решения задач; понятие алгоритма и структуры управления; структуры данных; основные требования методологии объектноориентированного программирования как технологической основы разработки качественных программных компонентов. Уметь: применять требования методологии объектноориентированного программирования при разработке программ; разрабатывать и записывать на языке программирования алгоритмы решения классических задач информатики; реализовывать технологию проектирования сверху вниз; выбрать оптимальную структуру для представления данных Владеть: навыками проектирования, реализации

		эффективных программ
ОПК-11	готовность использовать навыки выбора, проектирования, реализации, оценки качества и анализа эффективности программного обеспечения для решения задач в различных предметных областях	Знать: основные этапы компьютерного решения задач; понятие алгоритма и структуры управления; структуры данных; основные требования методологии объектноориентированного программирования как технологической основы разработки качественных программных компонентов. Уметь: применять требования методологии объектноориентированного программирования при разработке программ; разрабатывать и записывать на языке программирования алгоритмы решения классических задач информатики; реализовывать технологию проектирования сверху вниз; выбрать оптимальную структуру для представления данных Владеть: навыками проектирования, реализации эффективных программ

**12. Объем дисциплины в зачетных единицах/часах в соответствии с учебным планом — 5/180.**

**13. Виды учебной работы**

Вид учебной работы	Трудоемкость (часы)		
	Всего	По семестрам	
		3	4
Аудиторные занятия	64	32	32
в том числе:			
лекции	32	16	16
лабораторные			
практические	32	16	16
Самостоятельная работа	44	22	22
Итого	108	54	54
Промежуточная аттестация(экзамен)	72	36	36
<b>Итого:</b>	<b>180</b>	<b>90</b>	<b>90</b>

**13.1. Содержание разделов дисциплины**

№ п/п	Наименование раздела дисциплины	Содержание раздела дисциплины
<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(Solution), проект(project), пространство имен(namespace), сборка(assembly). Консольный проект.
1.3	Система типов в C#	Типы-значения(value), 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 IComparable, ICloneable, IEnumerable.
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 языка.	Алфавит языка C#. Имена, ключевые слова, 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, IEnumerable.
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	Многопоточность	Класс Thread. Средства синхронизации потоков.

### 13.2. Разделы дисциплины и виды занятий

№ п/п	Наименование раздела дисциплины	Виды занятий (часов)				
		Лекции	Лабораторные	Практические	Самостоятельная работа	Всего
1	Философия .NET	1			0	1
2	История создания языка C#. Структура приложения на языке C#.	1			2	3
3	Система типов в C#	1			2	3
4	Алфавит и лексемы языка.	1			2	3
5	Переменные, операции, выражения.	1		2	2	5
6	Операторы.	1		2	2	5
7	Классы.	2		2	2	6
8	Функции.	2		2	2	6
9	Массивы.	1		1	2	4
10	Символы и строки.	1		1	2	4
11	Операции класса.	1		1	2	4
12	Отношения между классами	1		1	2	4
13	Наследование	2		2	2	6
14	Исключения.	1		2	2	5
15	Интерфейсы	2		2	2	6
16	Контейнерные классы.	1		2	2	5
17	Делегаты	1		4	2	7
18	События	1		2	2	5
19	Универсальность	2		2	2	6
20	Сериализация	1		2	2	5
21	Паттерны программирования	2			2	4

22	Многопоточность	2		2	2	6
23	Небезопасный код	1			1	2
24	Сборка мусора	2			1	3
	Итого:	32		32	44	108

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

Работа с конспектами лекций, выполнение практических заданий для самостоятельной работы, выполнение лабораторных работ по дисциплине Б1.В.ОД.3 Практикум на ЭВМ по С#, использование рекомендованной литературы и методических материалов, в том числе находящихся в личном кабинете. Выполнение контрольных работ. Курсовая работа эссе по программированию.

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

а) основная литература:

№ п/п	Источник
1	Дэвис А. Асинхронное программирование в С# 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> —

б) дополнительная литература:

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

в) информационные электронно-образовательные ресурсы:

№ п/п	Источник
9	Электронный каталог Научной библиотеки Воронежского государственного университета. – <a href="http://www.lib.vsu.ru/">http://www.lib.vsu.ru/</a>
10	ЭБС «Издательство Лань» <a href="http://e.lanbook.com/">http://e.lanbook.com/</a>

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

№ п/п	Источник
1	Г.Э.Вощинская ,Структуры и алгоритмы обработки данных - Образовательный портал ВГУ: <a href="https://edu.vsu.ru/course/view.php?id=2797">https://edu.vsu.ru/course/view.php?id=2797</a> Режим доступа: личный кабинет студента
2	Воронина И.Е., Огаркова Н.В. Курсовая работа по программированию– Образовательный портал ВГУ: <a href="https://edu.vsu.ru/course/view.php?id=2797">https://edu.vsu.ru/course/view.php?id=2797</a> Режим доступа: личный кабинет студента

**17. Информационные технологии, используемые для реализации учебной дисциплины, включая программное обеспечение и информационно-справочные системы (при необходимости)**

ОС Windows, Microsoft Visual Studio 2010, Microsoft Visual Studio 2015.

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

Аудитория с проектором, доска

**19. Фонд оценочных средств:**

**19.1. Перечень компетенций с указанием этапов формирования и планируемых результатов обучения**

Код и содержание компетенции (или ее части)	Планируемые результаты обучения (показатели достижения заданного уровня освоения компетенции посредством формирования знаний, умений, навыков)	Этапы формирования компетенции (разделы (темы) дисциплины или модуля и их наименование)	ФОС* (средства оценивания)
ОПК-7 — способность использовать знания основных концептуальных положений функционального, логического, объектноориентированного и визуального направлений программирования, методов, способов и средств разработки программ в рамках этих направлений.	Знать: основные этапы компьютерного решения задач; понятие алгоритма и структуры управления; структуры данных; основные требования методологии объектноориентированного программирования как		Выполнение индивидуальных заданий лабораторного практикума.
	технологической основы разработки качественных программных компонентов; понятие статических и динамических данных; примеры базовых структур данных; идеи, лежащие в основе объектно-ориентированного программирования, реализацию взаимодействия классов.		

	<p>Уметь:</p> <p>применять требования методологии объектноориентированного программирования при разработке программ;</p> <p>разрабатывать и записывать на языке программирования алгоритмы решения классических задач информатики; реализовывать технологию проектирования сверху вниз; применять средства поддержки объектно-ориентированного программирования в языке программирования;</p> <p>выбрать оптимальную структуру для представления данных</p>		Контрольные работы №1-4
	<p>Владеть: навыками проектирования, реализации программ</p>		Контрольные работы №1-4
<p>ОПК-8</p> <p>способность программировать приложения и создавать программные прототипы решения прикладных задач</p>	<p>Знать: основные этапы компьютерного решения задач;</p> <p>понятие алгоритма и структуры управления; структуры данных; основные требования методологии объектноориентированного программирования как технологической основы разработки качественных программных компонентов; примеры базовых структур данных;</p> <p>идеи, лежащие в основе процедурного программирования, реализацию вызова процедур в языках с блочной структурой, рекурсию.</p>		Курсовая работа



	<p>Уметь:</p> <p>применять требования методологии объектноориентированного программирования при разработке программ; разрабатывать и записывать на языке программирования алгоритмы решения классических задач информатики;</p> <p>применять средства поддержки объектно-ориентированного программирования в языке программирования;</p> <p>выбрать оптимальную структуру для представления данных</p>		Контрольные работы №1-4
	<p>Владеть: навыками проектирования, реализации программ</p>		Контрольные работы №1-2
ОПК-11 готовность использовать навыки выбора, проектирования, реализации, оценки качества и анализа эффективности программного обеспечения для решения задач в различных предметных областях.	<p>Знать: основные этапы компьютерного решения задач;</p> <p>понятие алгоритма и структуры управления; структуры данных; основные требования методологии объектноориентированного программирования как технологической основы разработки качественных программных компонентов.</p>		Контрольные работы №1-2
	<p>Уметь: применять требования методологии объектноориентированного программирования при разработке программ; разрабатывать и записывать на языке программирования алгоритмы решения классических задач информатики; реализовывать технологию проектирования сверху вниз; выбрать оптимальную структуру для представления данных</p>		Контрольные работы №1-2
	<p>Владеть: навыками проектирования, реализации эффективных программ</p>		Контрольные работы №1-2
<b>Промежуточная аттестация</b>			Комплект КИМ №1-2

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

### Пример:

Для оценивания результатов обучения на экзамене/зачете используются следующие показатели:

- 1) владение навыками реализации программ;
- 2) знание методологии объектно-ориентированного программирования и применение ее на практике;
- 3) знание и умение использовать при решении задач базовые структуры данных;
- 4) умение выбирать и применять при реализации оптимальную структуру данных;
- 5) умение разрабатывать оптимальный алгоритм решения задачи и выполнять его реализацию; 4) знание теоретического материала

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

Соотношение показателей, критериев и шкалы оценивания результатов обучения.

Критерии оценивания компетенций	Уровень сформированности компетенций	Шкала оценок
Отличное знание теоретического материала, правильное и эффективное решение задачи, правильные ответы на тестовые вопросы. Должны быть выполнены ВСЕ лабораторные работы по дисциплине <i>Практикум на ЭВМ по С#</i>	<i>Повышенный уровень</i>	<i>Отлично</i>
Хорошее знание теоретического материала, в целом правильное решение задачи. НО: допускает незначительные ошибки в решении задачи. Неправильный ответ на тест. Должны быть выполнены ВСЕ лабораторные работы по дисциплине <i>Практикум на ЭВМ по С#</i>	<i>Базовый уровень</i>	<i>Хорошо</i>
Решение задачи не доведено до конца или недостаточное знание теоретического материала, ошибки в тестах Неоптимальное решение задачи и недостаточное владение теоретическим материалом. ИЛИ: выполнены все показатели базового уровня, но по дисциплине <i>Практикум на ЭВМ по программированию</i> не зачтены более одной задачи.	<i>Пороговый уровень</i>	<i>Удовлетворительно</i>
Задача не решена или серьезные пробелы в знании теоретического материала (с незнанием могут быть связаны и грубые ошибки в ответе на тестовые вопросы), по дисциплине <i>Практикум на ЭВМ по программированию</i> зачтены все задачи.	–	<i>Неудовлетворительно</i>

## 19.3 Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующие этапы формирования компетенций в процессе освоения образовательной программы

### 19.3.1 Перечень вопросов к экзамену:

1. Структура программы.
2. Область действия переменных.
3. Развернутые и ссылочные типы.
4. Операторы языка.

5. Массивы.
6. Строки string StringBuilder. Операции над строками. Форматирование строк.
7. Методы. Доступ к методам. Параметры методов. Перегрузка методов.
8. Понятие класса. Конструкторы. Методы. Поля. Свойства. Индексаторы.
9. Инкапсуляция. Наследование. Полиморфизм.
10. Абстрактные классы.
11. Структуры.
12. Перечисления.
13. Интерфейсы.
14. Множественное наследование.
15. Встроенные интерфейсы IEnumerable, ICloneable, IComparer, Comparable.
16. Сериализация.
17. Делегаты. События.
18. Универсальные классы, структуры, интерфейсы, делегаты. Ограничения.
19. Лямбда-выражения.
20. Сериализация. Бинарная сериализация. XML-сериализация. Десериализация.
21. Интерфейсы. Две стратегии реализации интерфейсов. Преобразование к классу интерфейса. Проблемы множественного наследования и их решение. Выбор между интерфейсами и классами. Встроенные интерфейсы Comparable, IComparer, ICloneable, IEnumerable.
22. Универсальность. Синтаксис универсального класса. Класс с универсальными методами. Синтаксис ограничений. Универсальные интерфейсы. Универсальные делегаты.
23. Делегаты и события. Использование делегатов. Передача делегатов в методы. Механизм обратного вызова. Стандартный классы EventHandler и EventArgs.
24. Паттерны проектирования: итератор, наблюдатель, фабрика, абстрактная фабрика, MVC.
25. Поток и синхронизация. Класс Thread. Создание потока. Передача параметров в поток. Синхронизация потоков: lock, Monitor, Semaphore, Mutex, EventWaitHandle.
26. Обработка исключений. Корректность и устойчивость программ. Классы Debug и Trace. Схема обработки исключений. Создание объектов Exception. Захват исключения.
27. Лямбда-выражения. Замыкания. Лямбда-операторы. Операции над выражениями. Мемоизация. Каррирование. Анонимная рекурсия.
28. Небезопасный код. Совмещение управляемого и неуправляемого кода. Разрешение небезопасного кода. Указатели, выделение памяти, фиксация, сравнение указателей.
29. Сборка мусора. Время жизни объектов. Поколения объектов. Параллельная и фоновая сборка мусора. Класс System.GC. Финализируемые объекты. Высвобождаемые объекты.

### **19.3.2 Перечень практических заданий**

Иллюстрируется на примере КИМ1 и КИМ2 **19.3.4**

#### **Тестовые задания**

Иллюстрируется на примере КИМ1 и КИМ2

#### **ПРИМЕРЫ КИМ 1**

Вопросы	. Бинарная сериализация и десериализация.  . Схема обработки исключений. Создание объектов сключения.	Exception. Захват
Задача	<p>. Опишите функцию для вычисления значения суммы ряда</p> $y = 1 + \frac{x^2}{2!} + \frac{x^4}{4!} + \frac{x^6}{6!} + \dots + \frac{x^{2n}}{2n!} + \dots$ <p>. с точностью <math>\epsilon</math>, используя цикл</p> <p>. for,</p> <p>. while, .</p> <p>do.</p>	
Тест	<p>Что выведет программа?</p> <pre>namespace контрольн_раб {     class Program     { static void Main(string[] args)</pre>	
	<pre>    {         A a = new A();         Console.WriteLine(a.ToString());     } } class A { string str;   public A()   { str = "это класс А";   } } }</pre>	

### КИМ2

Вопросы	<ol style="list-style-type: none"> <li>1. Паттерны проектирования. Шаблон наблюдатель.</li> <li>2. Класс Thread. Создание потока. Передача параметров в поток.</li> </ol>
Задача	<p>Опишите универсальный класс <b>матрица</b> с возможностью заполнения из файла и поэлементного изменения и с инициализацией события.</p> <p>Определите классы, обрабатывающие событие и тестирующий класс.</p>

Тест

```
Что выведет программа? public
class A
{
    static A()
    {
        Console.WriteLine("static A::A()");
    }
    private static int InitX()
    {
        Console.WriteLine("A.InitX()");
    }
    return 1;
}
    private static int InitY()
    {
        Console.WriteLine("A.InitY()");
    }
    return 2;
}
    private static int InitA()
    {
        Console.WriteLine("A.InitA()");
    }
    return 3;
}
    private static int InitB()
    {
        Console.WriteLine("A.InitB()");
    }
    return 4; }
```

### 19.3.4 Перечень заданий для контрольных работ

Иллюстрируется на примерах заданий для контрольных работ 1-2

#### Примеры заданий для контрольной работы № 1

##### Вариант № 1

1. Дан текстовый файл. Строки состоят из слов, разделённых одним или несколькими пробелами. Слова — произвольные последовательности символов. Слова не переносятся с одной строки на другую. Переписать в отдельный файл слова, являющиеся правильной записью целого десятичного числа
2. Приведите пример описания и использования делегата.

##### Вариант № 2

1. Дан текстовый файл. Строки состоят из слов, разделённых одним или несколькими пробелами. Слова — произвольные последовательности символов. Слова не переносятся с одной строки на другую. Создать новый файл, удалив все строки максимальной длины.
2. Как описываются перегруженные методы?

##### Критерии оценки:

- оценка «отлично» выставляется студенту, если правильно решена обе задачи (возможно с небольшими недочётами);
- оценка «хорошо» выставляется студенту, если первая задача решена правильно (возможно с небольшими недочётами); при решении третьей задачи правильно составлен алгоритм, а в реализации допущены ошибки;

- оценка «удовлетворительно» выставляется студенту, если 1-я задача решена верно, а для двух других задач (2-й и 3-й) правильно составлены алгоритмы решения, но при реализации допущены ошибки;
- оценка «неудовлетворительно» выставляется студенту, если не выполняются вышеуказанные критерии оценки.

## Примеры заданий для контрольной работы № 2

### Вариант № 1

1. Опишите универсальный класс **матрица** с возможностью заполнения из файла и поэлементного изменения и с инициализацией события. Определите классы, обрабатывающие событие и тестирующий класс.
2. Каков результат выполнения операторов:  
`...string str="программа"; str[2]='о'; Console.WriteLine(str);...`

### Вариант № 2

1. Опишите универсальный класс **двусвязный список**. Методы: вставка в начало, вставка в конец, поиск, сериализация, десериализация. Событие, если количество элементов превысит заданное значение или список станет пустым.
2. Приведите описание и создание матрицы A размером 3x4, если обращение к отдельному элементу A[i][j].

### Критерии оценки:

- оценка «отлично» выставляется студенту, если правильно решены две задачи (возможно с небольшими недочетами);
- оценка «хорошо» выставляется студенту, если одна задача решена правильно (возможно с небольшими недочетами); при решении второй задачи правильно составлен алгоритм, а в реализации допущены ошибки;
- оценка «удовлетворительно» выставляется студенту, если при решении обеих задач правильно составлен алгоритм, а в реализации допущены ошибки;
- оценка «неудовлетворительно» выставляется студенту, если не выполняются вышеуказанные критерии оценки.

## 19.4. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Оценка знаний, умений и навыков, характеризующая этапы формирования компетенций в рамках изучения дисциплины осуществляется в ходе текущей и промежуточной аттестаций.

Текущая аттестация проводится в соответствии с Положением о текущей аттестации обучающихся по программам высшего образования Воронежского государственного университета. Текущая аттестация проводится в форме контрольных работ.

Промежуточная аттестация проводится в соответствии с Положением о промежуточной аттестации обучающихся по программам высшего образования.

Контрольно-измерительные материалы промежуточной аттестации включают в себя практические задания, позволяющие оценить степень сформированности умений и навыков. При оценивании используются количественные шкалы оценок. Критерии оценивания приведены выше.