

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Тольяттинский государственный университет»

Б1.В.ДВ.02.02  
(индекс дисциплины)

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ**

Программирование на .Net (ДотНет) 1

(наименование дисциплины)

по направлению подготовки (специальности)  
09.03.03 Прикладная информатика

направленность (профиль)/специализация  
Цифровая трансформация бизнеса

Форма обучения: очная

Год набора: 2022

Общая трудоемкость: 5 ЗЕ

**Распределение часов дисциплины по семестрам**

Семестр	4	Итого
Форма контроля	экзамен	
Вид занятий		
Лекции	32	32
Лабораторные		
Практические	32	32
Промежуточная аттестация	0,35	0,35
Контактная работа	64,35	64,35
Самостоятельная работа	80	80
Контроль	35,65	35,65
<b>Итого</b>	<b>180</b>	<b>180</b>

Рабочую программу составил(и):

доцент, к.п.н., Ерофеева Е.А.

*(должность, ученое звание, степень, Фамилия И.О.)*

ст. преподаватель Дружинкин В.В.

*(должность, ученое звание, степень, Фамилия И.О.)*

Рецензирование рабочей программы дисциплины:



Отсутствует



Рецензент

*(должность, ученое звание, степень, Фамилия И.О.)*

Рабочая программа дисциплины составлена на основании ФГОС ВО и учебного плана направления подготовки (специальности)

09.03.03 Прикладная информатика

**Срок действия рабочей программы дисциплины до «31» августа 2026 г.**

УТВЕРЖДЕНО

На заседании кафедры  
«Прикладная математика и информатика»

(протокол заседания № 2 от «15» сентября 2021 г.).

## 1. Цель освоения дисциплины

Цель освоения дисциплины – сформировать у студентов практические навыки по разработке программных приложений на платформе .Net для решения прикладных задач с применением современных методов и технологий программирования, обучить работе с научно-технической литературой и технической документацией по разработке и тестированию приложений.

## 2. Место дисциплины в структуре ОПОП ВО

Данная дисциплина относится к Блоку Б1 «Дисциплины (модули)» (вариативная часть, дисциплины по выбору).

Дисциплины, учебные курсы, на освоении которых базируется данная дисциплина:

- Программирование на языках высокого уровня;
- Алгоритмы и структуры данных;
- Архитектура компьютеров и операционные системы;
- Объектно-ориентированное программирование.

Дисциплины, учебные курсы, для которых необходимы знания, умения, навыки, приобретаемые в результате изучения данной дисциплины:

- Разработка приложений на платформе Java 2;
- производственная практика;
- курсовое проектирование;
- подготовка выпускной квалификационной работы.

## 3. Планируемые результаты обучения

Формируемые и контролируемые компетенции (код и наименование)	Индикаторы достижения компетенций (код и наименование)	Планируемые результаты обучения
ПК-3- способен разрабатывать и адаптировать прикладное программное обеспечение	ПК-3.1- знает современные технологии разработки и адаптации прикладного программного обеспечения, их достоинства и недостатки	Знать: основные понятия, классификации и архитектуры информационных систем; базовые библиотеки современных объектно- ориентированных платформ программирования
	ПК-3.2 - умеет разрабатывать, адаптировать компоненты прикладного программного обеспечения	Уметь: пользоваться интегрированными средами разработки программного обеспечения
	ПК-3.3 - владеет навыками разработки прикладного программного обеспечения на современных языках программирования, методами адаптации прикладного программного обеспечения	Владеть: навыками проектирования и реализации сложного программного обеспечения на современных объектно- ориентированных платформ программирования

#### 4. Структура и содержание дисциплины

Модуль (раздел)	Вид учебной работы	Наименование тем занятий (учебной работы)	Семестр	Объем, ч.	Баллы	Интерактив, ч.	Формы текущего контроля (наименование оценочного средства)
1	Лекция 1	Введение в .NET Framework	4	4		-	тест
1	Практическое занятие 1	Введение в .NET Framework	4	4		-	
1	Лекция 2	Основные программные конструкции C#	4	4		-	
1	Практическое занятие 2	Основные программные конструкции C#	4	4		-	
2	Лекция 3	Объявление и вызов методов в C#	4	4		-	
2	Практическое занятие 3	Объявление и вызов методов в C#	4	4		-	
2	Лекция 4	Создание новых типов в C#	4	4		-	
2	Практическое занятие 4	Создание новых типов в C#	4	4		-	
2	Лекция 5	Инкапсуляция данных и методов. Перегрузка операций	4	4		-	
2	Практическое занятие 5	Инкапсуляция данных и методов. Перегрузка операций	4	4		-	
2	Лекция 6	Наследование. Интерфейсы и абстрактные классы	4	4		-	
2	Практическое занятие 6	Наследование. Интерфейсы и абстрактные классы	4	4		-	
2	Лекция 7	Обработка исключений	4	2		-	
2	Практическое занятие 7	Обработка исключений	4	2		-	
2	Лекция 8	Управление ресурсами в .NET. Сборка мусора Framework	4	2		-	

Модуль (раздел)	Вид учебной работы	Наименование тем занятий (учебной работы)	Семестр	Объем, ч.	Баллы	Интерактив, ч.	Формы текущего контроля (наиме- нование оценочного средства)
2	Практическое занятие 8	Управление ресурсами в .NET. Сборка мусора Framework	4	2		-	
3	Лекция 9	Работа с файловой системой Framework	4	2		-	
3	Практическое занятие 9	Работа с файловой системой Framework	4	2		-	
3	Лекция 10	Делегаты и события	4	2		-	
3	Практическое занятие 10	Делегаты и события	4	2		-	
<b>Итого:</b>				<b>64</b>			

## **5. Образовательные технологии**

В рамках изучения дисциплины «Программирование на .Net (ДотНет) 1» предусмотрено использование следующих образовательных технологий:

- технология традиционного обучения: лекции и практические работы, самостоятельная работа;
- технология проектного обучения: реализация и защита отчетов по практическим работам.

## **6. Методические указания по освоению дисциплины**

### **6.1. Рекомендации по подготовке к лекционным занятиям**

Изучение дисциплины требует систематического и последовательного накопления знаний, следовательно, пропуски отдельных тем не позволяют глубоко освоить предмет.

В ходе лекционных занятий рекомендуется конспектировать учебный материал, обращая внимание на категории, формулировки, раскрывающие содержание тех или иных явлений и процессов, научные выводы и практические рекомендации, положительный опыт в ораторском искусстве. Желательно оставить в рабочих конспектах поля, на которых делать пометки из рекомендованной литературы, дополняющие материал прослушанной лекции, а также подчеркивающие особую важность тех или иных теоретических положений. Задавать преподавателю уточняющие вопросы с целью уяснения теоретических положений, разрешения спорных ситуаций.

В ходе подготовки к лекциям изучить основную литературу, ознакомиться с дополнительной литературой, публикациями в Интернет-источниках, периодических изданиях. При этом учесть рекомендации преподавателя и требования учебной программы. Дорабатывать свой конспект лекции, делая в нем соответствующие записи из литературы, рекомендованной преподавателем и предусмотренной учебной программой.

Студент может дополнить список использованной литературы современными источниками, не представленными в списке рекомендованной литературы, и в дальнейшем использовать собственные подготовленные учебные материалы при написании курсовых и выпускных квалификационных работ.

### **6.2. Рекомендации по подготовке к практическим занятиям**

Студентам следует:

- до очередного практического занятия по рекомендованным литературным источникам проработать теоретический материал, соответствующей темы занятия;
- при подготовке к практическим занятиям следует обязательно использовать не только лекции, учебную литературу, но и другие источники;
- в начале занятий задать преподавателю вопросы по материалу, вызвавшему затруднения в его понимании и освоении при решении задач, заданных для самостоятельного решения;
- на занятии доводить каждую задачу до окончательного решения, демонстрировать понимание проведенных расчетов (анализов, ситуаций), в случае затруднений обращаться к преподавателю.

Для того чтобы практические занятия приносили максимальную пользу, необходимо помнить, что упражнение и решение задач проводятся по рассмотренному на лекциях материалу и связаны, как правило, с детальным разбором отдельных вопросов лекционного курса. Следует подчеркнуть, что только после усвоения лекционного материала с определенной точки зрения (а именно с той, с которой он излагается на лекциях) он будет закрепляться студентом на практических занятиях как в результате обсуждения и анализа лекционного материала, так и с помощью решения проблемных ситуаций, задач. При этих условиях студент не только хорошо усвоит материал, но и научится применять его на практике, а также получит дополнительный стимул (и это очень важно) для активной проработки лекции.

При самостоятельном решении задач нужно обосновывать каждый этап решения, исходя из теоретических положений курса. Если студент видит несколько путей решения проблемы (задачи), то нужно сравнить их и выбрать самый рациональный. Полезно до начала вычислений составить краткий план решения проблемы (задачи). Решение проблемных задач или примеров следует излагать подробно, вычисления располагать в строгом порядке, отделяя вспомогательные вычисления от основных. Решения при необходимости нужно сопровождать комментариями, схемами, чертежами и рисунками.

Следует помнить, что решение каждой учебной задачи должно доводиться до окончательного логического ответа, которого требует условие, и по возможности с выводом. Полученный ответ следует проверить способами, вытекающими из существа данной задачи. Полезно также (если возможно) решать несколькими способами и сравнить полученные результаты. Решение задач данного типа нужно продолжать до приобретения твердых навыков в их решении.

### **6.3. Рекомендации по подготовке к экзамену**

Подготовка к экзамену способствует закреплению, углублению и обобщению знаний, получаемых, в процессе обучения, а также применению их к решению практических задач. Готовясь к экзамену, студент ликвидирует имеющиеся пробелы в знаниях, углубляет, систематизирует и упорядочивает свои знания. На экзамене студент демонстрирует то, что он приобрел в процессе обучения по конкретной учебной дисциплине.

На консультации перед экзаменом студенты должны быть ознакомлены с основными требованиями и получить ответы на возникающие в процессе подготовки вопросы.

Необходимо ориентировать студентов на систематическую подготовку к занятиям в течение семестра, что позволит использовать время экзаменационной сессии для систематизации знаний.

## 7. Оценочные средства

### 7.1. Паспорт оценочных средств

Семестр	Код контролируемой компетенции (или ее части)	Наименование оценочного средства
4	ПК-3	Вопросы для собеседования по модулю 1
		Комплект отчетов по практическим работам 1-2
4	ПК-3	Вопросы для собеседования по модулю 2
		Комплект отчетов по практическим работам 3-10
4	ПК-3	Вопросы для собеседования по модулю 3
4	ПК-3	Комплект отчетов по практическим работам 11-18

### 7.2. Типовые задания или иные материалы, необходимые для текущего контроля

#### 7.2.1. Вопросы для собеседования по модулю (примеры вопросов)

(наименование оценочного средства)

##### Типовые примеры заданий

##### Модуль 1. Введение в платформы Java

1. Поясните технологию установки и настройки JDK (JAVA\_HOME, PATH, CLASSPATH).
2. Дайте общую характеристику утилитам и средствам JDK.
3. Охарактеризуйте основные средства (Basic Tools) JDK: javac, java, jar, appletviewer, jdb.
4. Опишите работу с утилитами javac, java.
5. Опишите общую технологию работы с утилитой jar.
6. Подробно расскажите как можно создать, модифицировать jar-файл.
7. Что такое точка входа (Entry Point)? Как можно задать entry point? Как запустить программу из jar-файла?
8. Что такое технология и платформа java?
9. JVM, области памяти JVM.
10. Сборка мусора, острова изоляции.
11. Загрузка классов.

##### Критерии оценки:

- оценка «зачтено» выставляется студенту, если продемонстрированы всесторонние, систематизированные, глубокие знания по основным принципам функционирования платформ Java;
- оценка «не зачтено» выставляется студенту, если продемонстрированы фрагментарные, несистематизированные знания по основным принципам функционирования платформ Java, допускаются неточности и ошибки причинно-следственных связей.

##### Модуль 2. Базовые технологии Java SE

1. Классы: определение, объявление, модификаторы доступа.
2. Члены класса: определение, модификаторы доступа, объявление.
3. Конструкторы классов.
4. Создание экземпляров классов. Безымянные классы.
5. Абстрактные методы и классы.



6. Окончательные методы и классы.
7. Статические члены класса.
8. Наследование. Отношения “Is-A” и “Has-A”.
9. Перегрузка и переопределение методов.
10. Вложенные классы.
11. Назначение класса `Object`.
12. Назначение класса `Class`.
13. Контракт метода `equals()`.
14. Контракт метода `hashCode()`.
15. Контракт метода `clone()`.
16. Переопределение метода `toString()`.

#### **Критерии оценки:**

- оценка «зачтено» выставляется студенту, если продемонстрированы всесторонние, систематизированные, глубокие знания по базовым технологиям Java SE;
- оценка «не зачтено» выставляется студенту, если продемонстрированы фрагментарные, несистематизированные знания по базовым технологиям Java SE, допускаются неточности и ошибки причинно-следственных связей.

### **Модуль 3. Продвинутое технологии Java SE**

1. Раскройте понятия многозадачность (multitask), многопоточность (concurrency), параллельность (parallel).
2. Раскройте понятие процесса и потока.
3. Создание потока через `Thread`.
4. Создание потока через `Runnable`.
5. Какие данные предоставляются совместно (shared) и несовместно (non-shared).
6. Атомарные операции.
7. Ключевое слово `volatile`.
8. Остановка потока.
9. Прерывание остановки.
10. Потоки демоны.
11. Мьютексы, семафоры, мониторы. Применение в java.
12. Реализация критических секций.
13. Deadlock.

#### **Критерии оценки:**

- оценка «зачтено» выставляется студенту, если продемонстрированы всесторонние, систематизированные, глубокие знания по продвинутым технологиям Java SE;
- оценка «не зачтено» выставляется студенту, если продемонстрированы фрагментарные, несистематизированные знания по продвинутым технологиям Java SE, допускаются неточности и ошибки причинно-следственных связей.

### **7.2.2. Комплект отчетов по практическим работам (примеры)**

#### **Практические занятия «Введение в .NET».**

### **Задания к практическим занятиям:**

- Разработать консольное приложение для чтения небольшого набора входных данных, форматирования этих данных удобными для чтения образом и отображения отформатированных результатов. Данные представляют собой текст, содержащий пары чисел, представляющие x и y- координаты местоположения объекта. Каждая строка текста содержит один набор координат. Следующий пример демонстрирует типичный набор данных.
- 23.8976,12.3218  
25.76,11.9463  
24.8293,12.2
- В отформатированном виде данные могут выглядеть следующим образом:
- X: 23,8976 Y: 12,3218  
X: 25,76 Y: 11,9463  
X: 24,8293 Y: 12,2
- Перенаправить ввод для запуска приложения с помощью текстового файла, содержащего данные, которые нужно отформатировать.
- Реализовать простое консольное (или WPF) приложение обеспечивающее описанные выше функциональные возможности. Предусмотреть возможность введения данных пользователем путем предоставления поля.
- Перенаправить ввод для запуска приложения с помощью данных из файла. Создать дополнительные тестовые данные и использовать их в качестве входных данных для тестирования приложения.
- Использовать Visual Studio 2010 (или 2012) отладчик для пошагового выполнения кода и изучения работы приложения.
- Добавить XML комментарии к разработанному приложению и создать для него документацию, используя инструмент Sandcastle.

### **Практические занятия «Основные программные конструкции C#».**

#### **Задания к практическим занятиям:**

- Разработать тип, позволяющий вычислять корень n-ой степени из числа методом Ньютона с заданной точностью.
- Предусмотрев в нем возможность сравнения полученного результата со значением, рассчитываемым с помощью метода Math.Pow библиотеки классов .NET Framework.
- Обязательно должны быть «приглашения для ввода», и отображение результата, так чтобы пользователю было понятно, что отображается.
  - Без очепяток!
- В качестве UI-интерфейса использовать консольное приложение с интерфейсом командной строки, WinForms или WPF-приложение.
- Разработать тип, позволяющий конвертировать неотрицательное десятичное значение целого числа в строку, содержащую двоичное представление этого значения.
- Нужно реализовать два варианта:
  - Использовать стандартные классы и методы для конвертирования.
  - Реализовать свой алгоритм конвертирования.
- Обязательно должны быть «приглашения для ввода», и отображение результата, так чтобы пользователю было понятно, что отображается.
  - Без очепяток!
- В качестве UI-интерфейса использовать консольное приложение с интерфейсом командной строки, WinForms или WPF-приложение.

### **Практические занятия «Объявление и вызов методов в C#».**

#### **Задания к практическим занятиям:**

- Разработать тип, реализующий алгоритм Евклида для вычисления НОД двух целых чисел ([http://en.wikipedia.org/wiki/Euclidean\\_algorithm](http://en.wikipedia.org/wiki/Euclidean_algorithm)).
- Добавить к разработанному типу дополнительную функциональность в виде перегруженных методов вычисления НОД для трех, четырех или пяти целых чисел.
- Добавить к разработанному типу метод, реализующий алгоритм Стейна (бинарный алгоритм Евклида) для расчета НОД двух целых чисел ([http://en.wikipedia.org/wiki/Binary\\_GCD\\_algorithm](http://en.wikipedia.org/wiki/Binary_GCD_algorithm)).
- Сделать «обертку для метода», которая принимает выходной параметр, содержащий значение времени, необходимое для выполнения расчетов.
- Сделать «метод-обёртку» для метода реализующего алгоритм Евклида вычисления НОД двух параметров, добавив аналогичную функциональность.
- Создать unit-тесты для тестирования разработанных методов.
- В качестве UI-интерфейса использовать консольное приложение с интерфейсом командной строки, WinForms или WPF-приложение.

### **Содержание отчета**

- I. Титульный лист.
- II. Название и цель работы.
- III. Результаты выполнения работы
- IV. Заключение
- V. Приложения

### **Требования к оформлению**

Отчёт по практическому занятию выполняется на страницах формата А4 в электронном виде.

При оформлении отчёта используется сквозная нумерация страниц, считая титульный лист первой страницей. Номер страницы на титульном листе не ставится. Номера страницы ставятся по центру сверху.

При оформлении отчёта соблюдать следующие требования:

- Для заголовков: полужирный шрифт, 14 пт, центрированный.
- Для основного текста: нежирный шрифт, 14 пт, выравнивание по ширине.
- Во всех случаях тип шрифта – Times New Roman, отступ абзаца 1.25 см, полуторный междустрочный интервал.
- Поля: левое – 2 см, правое, верхнее и нижнее – 1 см.

### **Процедура оценивания**

Оценка выполненного практического занятия проводится по следующим критериям:

2. Наличие всей существенной информации по работе
3. Точность и полнота предоставляемых сведений
4. Непротиворечивость приводимой информации
5. Правильность интерпретаций и выводов, которые сделаны по результатам работы
6. Степень достижения студентом поставленной цели
7. Обоснованность применяемого решения
8. Грамотность (содержательная) используемых формулировок

**Критерии оценки** за отчеты по практическим занятиям:

- оценка «зачтено» ставится студенту, который продемонстрировал результаты выполнения практического занятия, соответствующие поставленным задачам, и предоставил отчет, оформленный должным образом и содержащий краткое описание полученных результатов;
- оценка «не зачтено» ставится студенту, который не продемонстрировал результаты выполнения практического занятия или не представил по ней отчет или представленный отчет не соответствует требованиям по оформлению.

**7.3. Оценочные средства для промежуточной аттестации по итогам освоения дисциплины****7.3.1. Вопросы к промежуточной аттестации**

Семестр \_\_\_\_\_ 6 \_\_\_\_\_

№ п/п	Вопросы к экзамену
1.	Объектно-ориентированная парадигма программирования.
2.	Технологии CLR, CTS, CLS.
3.	Метаданные. Работа с пространствами имен.
4.	Понятие класса в C#.
5.	Объявления метода Main(). Обработка параметров командной строки.
6.	Создание объектов: конструкторы.
7.	Использование памяти для структурных (числовые данные, перечисления и структуры) и ссылочных (классы и интерфейсы) типов.
8.	Класс System.Object. Замещение методов System.Object.
9.	Определение пользовательских методов класса.
10.	Модификаторы уровня доступа к методам.
11.	Модификаторы для параметров методов.
12.	Правила объявления и определения размеров массива. Одномерные и многомерные массивы.
13.	Базовый класс System.String. Управляющие последовательности и вывод служебных символов.
14.	Перечисления в C#. Базовый класс System.Enum. Примеры использования.
15.	Определение структур в C#.
16.	Преобразование структурного типа в ссылочный: упаковка и распаковка. Пользовательские пространства имен.
17.	Формальное определение класса как пользовательского типа данных. Элементы класса. Примеры реализации.
18.	Понятие инкапсуляции как одного из основных принципов объектно-ориентированного программирования.
19.	Применение свойств класса. Внутреннее представление свойств.
20.	Понятие наследования как одного из основных принципов объектно-ориентированного программирования. Примеры реализации.
21.	Понятие полиморфизма как одного из основных принципов объектно-ориентированного программирования. Примеры реализации.
22.	Принципы реализации графического интерфейса программных приложений.
23.	Проектирование базы данных с использованием MS SQL - сервера.

№ п/п	Вопросы к экзамену
24.	Пространства имен ADO.NET. Типы пространства имен System.Data.
25.	Тип DataView. Возможности класса DataSet.
26.	Статические методы и методы экземпляров.
27.	Статические данные.
28.	Библиотека базовых классов .NET. Сборки. MSIL.
29.	Статические члены System.Object. Упаковка и распаковка.
30.	Базовый класс System.Array. Примеры алгоритмов обработки.
31.	Применение System.StringBuilder. Примеры использования.
32.	Ссылки класса на самого себя, перенаправление вызовов конструктора с использованием this. Определение открытого интерфейса по умолчанию. Примеры реализации.
33.	Реализация инкапсуляции при помощи традиционных методов доступа и изменения.
34.	Тип DataColumn. Первичный ключ таблицы. Тип DataRow. Тип DataTable.
35.	Использование класса DataRelation для моделирования отношений между таблицами.
36.	Статические конструкторы. Статические поля.
37.	Принципы построения приложений обработки баз данных.
38.	Базовый и производные классы. Работа с конструктором базового класса. Примеры реализации.
39.	Работа с конструктором базового класса. Примеры реализации.
40.	Запрет наследования. Применение модели включения-делегирования. Примеры реализации.
41.	Механизм позднего связывания. Примеры реализации.
42.	Абстрактные классы. Абстрактные методы.
43.	Конструкторы и деструкторы.
44.	Философия .NET.
45.	Открытые, закрытые и защищенные методы класса. Примеры реализации.
46.	Использование памяти для структурных (числовые данные, перечисления и структуры) и ссылочных (классы и интерфейсы) типов.
47.	Класс System.Object. Замещение методов System.Object.
48.	Определение пользовательских методов класса.
49.	Модификаторы уровня доступа к методам.
50.	Модификаторы для параметров методов.
51.	Понятие инкапсуляции как одного из основных принципов объектно-ориентированного программирования.
52.	Применение свойств класса. Внутреннее представление свойств.
53.	Понятие наследования как одного из основных принципов объектно-ориентированного программирования. Примеры реализации.
54.	Понятие полиморфизма как одного из основных принципов объектно-ориентированного программирования. Примеры реализации.
55.	Принципы реализации графического интерфейса программных приложений.
56.	Проектирование базы данных с использованием MS SQL - сервера.
57.	Пространства имен ADO.NET. Типы пространства имен System.Data.
58.	Тип DataView. Возможности класса DataSet.
59.	Статические методы и методы экземпляров.
60.	Статические данные.
61.	Библиотека базовых классов .NET. Сборки. MSIL.
62.	Статические члены System.Object. Упаковка и распаковка.
63.	Базовый класс System.Array. Примеры алгоритмов обработки.
64.	Применение System.StringBuilder. Примеры использования.
65.	Ссылки класса на самого себя, перенаправление вызовов конструктора с использова-

№ п/п	Вопросы к экзамену
	нием this. Определение открытого интерфейса по умолчанию. Примеры реализации.
66.	Реализация инкапсуляции при помощи традиционных методов доступа и изменения.
67.	Тип DataColumn. Первичный ключ таблицы. Тип DataRow. Тип DataTable.
68.	Использование класса DataRelation для моделирования отношений между таблицами.
69.	Статические конструкторы. Статические поля.
70.	Принципы построения приложений обработки баз данных.
71.	Базовый и производные классы. Работа с конструктором базового класса. Примеры реализации.
72.	Работа с конструктором базового класса. Примеры реализации.
73.	Запрет наследования. Применение модели включения-делегирования. Примеры реализации.
74.	Механизм позднего связывания. Примеры реализации.
75.	Абстрактные классы. Абстрактные методы.
76.	Конструкторы и деструкторы.
77.	Философия .NET.
78.	Открытые, закрытые и защищенные методы класса. Примеры реализации.
79.	Использование памяти для структурных (числовые данные, перечисления и структуры) и ссылочных (классы и интерфейсы) типов.
80.	Класс System.Object. Замещение методов System.Object.
81.	Определение пользовательских методов класса.
82.	Модификаторы уровня доступа к методам.
83.	Модификаторы для параметров методов.
84.	В связи с чем язык XML получил широкое распространение?
85.	Что представляет собой язык разметки?
86.	Какими особенностями обладает язык разметки XML?
87.	Что представляет собой платформа Microsoft .NET Framework?
88.	Каково назначение платформы Microsoft .NET Framework?
89.	Из каких компонентов состоит платформа Microsoft .NET Framework?
90.	Каковы функции среды CLR?
91.	Что представляет собой библиотека классов Microsoft .NET Framework?
92.	На основе каких стандартов строится web-служба XML в рамках платформы Microsoft .NET Framework?
93.	Какие API для реализации работы с XML разработаны в рамках платформы Microsoft .NET Framework?
94.	Что представляет собой XML-документ?
95.	Из каких разделов состоит XML-документ?
96.	Какую информацию содержит декларация XML-документа?
97.	Какая информация размещается в прологе XML-документа?
98.	Как строятся элементы XML-документа?
99.	Каковы синтаксические правила создания атрибутов XML документа?
100.	Для чего в XML-документах используются комментарии?
101.	Чем парсируемые данные в XML-документе отличаются от непарсируемых данных?
102.	Что представляют собой сущности XML-документа?
103.	Для чего в XML-документе необходимо пространство имен?
104.	Для чего необходима валидация XML-документа?
105.	Что определяет DTD-схема документа?
106.	Какие инструкции используются в DTD-схеме для описания грамматики XML-документа?
107.	Каковы достоинства и недостатки DTD-схем?

<b>№ п/п</b>	<b>Вопросы к экзамену</b>
108.	Что представляет собой XML(XDR)-схема?
109.	Какие структурные элементы и характеристики XML-документа описывает XML(XDR)-схема?
110.	В чем заключается разница между простыми и комплексными типами элементов в XML(XDR)-схемах?
111.	Какие типы элементов, описываемых с помощью XML(XDR)-схем, могут содержать атрибуты?
112.	Что определяют примитивы XML(XDR)-схем?
113.	В чем заключается основная разница между DTD и XML(XDR)схемами?
114.	Для чего разработан язык XPath? Как прописываются пути в XPath?
115.	Что определяет ось выборки XPath?
116.	С какой целью в XPath используются предикаты?
117.	Чем определяется контекст выражения в XPath?
118.	Каково основное назначение технологии XSLT?
119.	Как строятся шаблоны преобразований в XSLT?
120.	Какова структура шаблона XSLT?
121.	Как с помощью языка XSLT создаются новые структурные единицы документа (элементы, атрибуты и т.п.)?
122.	Каков алгоритм преобразования XML-документа с помощью языка XSLT?

### 7.3.2. Критерии и нормы оценки

<b>Семестр</b>	<b>Форма проведения промежуточной аттестации</b>	<b>Критерии и нормы оценки</b>	
6	Экзамен	«зачтено»	
		«не зачтено»	
		«отлично»	студенты, набравшие 80 и более баллов по накопительному рейтингу
		«хорошо»	студенты, набравшие более 60 но менее 80 баллов по накопительному рейтингу
		«удовлетворительно»	студенты, набравшие более 40 менее 60 баллов по накопительному рейтингу
		«неудовлетворительно»	студенты, набравшие менее 40 баллов по накопительному рейтингу

## 8. Учебно-методическое и информационное обеспечение дисциплины

### 8.1. Обязательная литература

№ п/п	Авторы, составители	Заглавие (заголовок)	Тип (учебник, учебное пособие, учебно-методическое пособие, практикум, др.)	Год издания	Количество в научной библиотеке / Наименование ЭБС
1	Гуриков С.Р.	Введение в программирование на языке Visual C#: учеб. пособие / С.Р. Гуриков. — М. : ФОРУМ : ИНФРА-М, 2017. — 447 с. — (Высшее образование: Бакалавриат).	учебное пособие	2017	ЭБС Znanium
2	Зыков С. В.	Основы современного программирования [Электронный ресурс] : Разработка гетерогенных систем в Интернет-ориентированной среде : учеб. курс / С. В. Зыков. - Саратов : Ай Пи Эр Медиа, 2017. - 484 с. : ил. - ISBN 978-5-9908055-9-0.	учебное пособие	2017	ЭБС "IPRbooks"
3	Каймин В.А.	Информатика: Учебник / В. А. Каймин - 6-е изд. – Москва : НИЦ ИНФРА-М, 2015. - 285 с.: 60х90 1/16. - (Высшее образование: Бакалавриат)ISBN 978-5-16-010876-6	учебник	2015	ЭБС Znanium
	Снетков В. М.	Практикум прикладного программирования на C# в среде VS.NET 2008 [Электронный ресурс] / В. М. Снетков. - 2-е изд., испр. - Москва : ИНТУИТ, 2016. - 1691 с.	практикум	2016	ЭБС "IPRbooks"
	Тюкачев Н. А.	C#. Программирование 2D и 3D векторной графики [Электронный ресурс] : учеб. пособие / Н. А. Тюкачев, В. Г. Хлебостроев. - Изд. 2-е, испр. и доп. - Санкт-Петербург : Лань, 2017. - 320 с. : ил. - (Учебники для вузов. Специальная литература). - ISBN 978-5-8114-2568-6.	учебное пособие	2017	ЭБС "Лань"



## 8.2. Дополнительная литература

№ п/п	Авторы, составители	Заглавие (заголовок)	Тип (учебник, учебное пособие, учебно-методическое пособие, практикум, др.)	Год издания	Количество в научной библиотеке / Наименование ЭБС
1	Костюкова Н. И.	Комбинаторные алгоритмы для программистов [Электронный ресурс] : [учеб. пособие] / Н. И. Костюкова. - 2-е изд., испр. - Москва : ИНТУИТ, 2016. - 217 с. : ил. - (Основы информационных технологий). - ISBN 978-5-9556-0069-7.	учебное пособие	2016	ЭБС "IPRbooks"
2	Кирнос В. Н.	Информатика II. Основы алгоритмизации и программирования на языке C++ [Электронный ресурс] : учебно-методическое пособие / В. Н. Кирнос. - Томск : Эль Контент, 2013. - 160 с. : ил.	учебно-методическое пособие	2013	ЭБС "IPRbooks"
3	Туральчук К. А.	Параллельное программирование с помощью языка C# [Электронный ресурс] : [учеб. пособие] / К. А. Туральчук. - 2-е изд., испр. - Москва : ИНТУИТ, 2016. - 189 с. : ил. - (Основы информационных технологий).	практикум	2016	ЭБС "IPRbooks"
4	Хорев П. Б.	Объектно-ориентированное программирование с примерами на C# [Электронный ресурс] : учеб. пособие / П. Б. Хорев. - Москва : Форум : ИНФРА-М, 2016. - 200 с. - (Высшее образование. Бакалавриат). - ISBN 978-5-00091-144-0.	учебное пособие	2016	ЭБС Znanium

### 8.3. Перечень профессиональных баз данных и информационных справочных систем

1. Бесплатный для студентов, аспирантов, школьников и преподавателей доступ к полным лицензионным версиям инструментов Microsoft для разработки и дизайна на <http://www.dreamspark.ru/> WebofScience[Электронный ресурс] : мультидисциплинарная реферативная база данных. – Philadelphia: ClarivateAnalytics, 2016– . – Режим доступа : apps.webofknowledge.com. – Загл. с экрана. – Яз. рус.,англ.
2. Scopus[Электронный ресурс] : реферативная база данных. – Netherlands: Elsevier, 2004– . – Режим доступа : scopus.com. – Загл. С экрана.– Яз. рус., англ.
3. Elibrary[Электронный ресурс] : научная электронная библиотека. – Москва : НЭБ, 2000– . – Режим доступа : elibrary.ru. – Загл. с экрана. – Яз. рус., англ.

### 8.4. Перечень программного обеспечения

№ п/п	Наименование ПО	Реквизиты договора (дата, номер, срок действия)
1	DreamSpark в составе: Microsoft Visio; Microsoft Visual Studio; Microsoft Access; Microsoft Project	До 01.07.2020. Продлевается каждые 3 года

### 8.5. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине

№ п/п	Наименование оборудованных учебных кабинетов, лабораторий, мастерских и др. объектов для проведения практических и лабораторных занятий, помещений для самостоятельной работы обучающихся (номер аудитории)	Перечень основного оборудования
1	Учебная аудитория для проведения занятий лекционного типа. Учебная аудитория для проведения занятий семинарского типа. Учебная аудитория для курсового проектирования (выполнения курсовых работ). Учебная аудитория для проведения групповых и индивидуальных консультаций. Учебная аудитория для проведения занятий текущего контроля и промежуточной аттестации.	Стол ученический двух- местный (моноблок), доска аудиторная 3-х секционная (меловая), стол преподавательский, стул, проектор Acer
2	Компьютерный класс. Учебная аудитория для проведения занятий лекционного типа. Учебная аудитория для проведения занятий семинарского типа. Учебная аудитория для проведения лабораторных работ. Учебная аудитория для	Стол ученический, стол преподавательский, стул, доска аудиторная (маркерная), компьютер с выходом в

№ п/п	<b>Наименование оборудованных учебных кабинетов, лабораторий, мастерских и др. объектов для проведения практических и лабораторных занятий, помещений для самостоятельной работы обучающихся (номер аудитории)</b>	<b>Перечень основного оборудования</b>
	курсового проектирования (выполнения курсовых работ). Учебная аудитория для проведения групповых и индивидуальных консультаций Учебная аудитория для проведения занятий текущего контроля и промежуточной аттестации.	сеть Интернет.
3	Компьютерный класс. Помещение для самостоятельной работы. Учебная аудитория для проведения занятий семинарского типа. Учебная аудитория для курсового проектирования (выполнения курсовых работ). Учебная аудитория для проведения групповых и индивидуальных консультаций. Учебная аудитория для проведения занятий текущего контроля и промежуточной аттестации.	Стол ученический, стул, компьютер с выходом в сеть интернет.

