

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

Б1.В.ДВ.04.02

(индекс дисциплины)

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Разработка приложений на платформе Net (Дотнет) 2

(наименование дисциплины)

по направлению подготовки

01.03.02 Прикладная математика и информатика

направленность (профиль)

Компьютерные технологии и математическое моделирование

Форма обучения: очная

Год набора: 2026

Общая трудоемкость: 4 ЗЕ

Распределение часов дисциплины по семестрам

Семестр	7	Итого
Форма контроля	зачет	
Вид занятий		
Лекции	16	16
Лабораторные		
Практические	48	48
Руководство: курсовые работы (проекты) / РГР		
Промежуточная аттестация	0,25	0,25
Контактная работа	64,25	64,25
Самостоятельная работа	79,75	79,75
Контроль		
Итого	144	144

Рабочую программу составил:

старший преподаватель института цифровых технологий, 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

01.03.02 Прикладная математика и информатика

Срок действия рабочей программы дисциплины до «31» августа 2030 г.

УТВЕРЖДЕНО

На заседании

института цифровых технологий

(протокол заседания № 1 от «05» сентября 2025 г.).

1. Цель освоения дисциплины

Цель освоения дисциплины – сформировать у обучаемых 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

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

- Архитектура компьютеров и операционные системы;
- Объектно-ориентированное программирование.
- Программирование на Net (Дотнет) 1

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

- Производственная практика (преддипломная практика).

3. Планируемые результаты обучения

Формируемые и контролируемые компетенции (код и наименование)	Индикаторы достижения компетенций (код и наименование)	Планируемые результаты обучения
ПК-5 Способен осуществлять выбор языка программирования и моделировать решение для реализации программного обеспечения	ПК-5.1 Знает принципы выбора языка программирования и моделирования решений для реализации программного обеспечения	Знать: основные особенности бизнес-ориентированных языков, их сильные и слабые стороны, а также оптимальные области применения в бизнес-среде. Уметь: применять бизнес-ориентированные языки для решения бизнес-задач, понимая требования и выбирая подходящие конструкции языка. Владеть: навыками анализа бизнес-требований и перевода их в код на бизнес-ориентированный язык с учетом преимуществ языка..
	ПК-5.2 Умеет использовать знания по выбору языка программирования и моделированию решений для реализации программного обеспечения	Знать: принципы разработки приложений, включая бизнес-ориентированное программирование, структуры данных и алгоритмы, актуальные для бизнес-приложений. Уметь: -проектировать и реализовывать приложения на бизнес-ориентированных языках, которые соответствуют определённым требованиям бизнеса, обеспечивая их функциональность и производительность. - пользоваться интегрированными средами разработки программного обеспечения. Владеть: приемами использования среды разработки бизнес-ориентированных языков, а также инструментов для отладки и тестирования информационных систем.
	ПК-5.3 Владеет навыками выбора языка программирования и моделирования решений для реализации программного	Знать: современные фреймворки, библиотеки и инструменты для бизнес-ориентированного языка, которые повышают эффективность разработки и производительность приложений. Уметь: оценивать и выбирать подходящие инструменты и технологии для конкретных бизнес-проектов, таких как базы данных, API и фреймворки. Владеть: навыками проектирования и реализации сложного программного обеспечения на современных бизнес-ориентированных платформах 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	Лек 1	Коллекции и обобщенные типы	7	2	-	-	
1	Лек 2	Коллекции и обобщенные типы	7	2	-	-	
1	СР	Коллекции и обобщенные типы	7	20	-	-	
1	Пр 1	Коллекции и обобщенные типы	7	2	-	-	Отчет по практической работе №1
1	Пр 2	Бинарное дерево поиска	7	2		-	
1	Пр 3	Бинарное дерево поиска	7	2		-	
1	Пр 4	XML и платформа Microsoft .NET Framework».	7	2		-	
1	Пр 5	Основы XML	7	2		-	
2	Лек 3	Введение в LINQ	7	2	-	-	
2	Лек 4	XML и платформа Microsoft .NET Framework	7	2	-	-	
2	СР	Введение в LINQ	7	20	-	-	
2	Пр 6	Введение в LINQ	7	2	-	-	Отчет по практической работе №2
2	Пр 7	Введение в LINQ	7	2		-	
2	Пр 8	Введение в LINQ	7	2		-	

2	Пр 9	Введение в LINQ	7	2		-	
2	Пр 10	Введение в LINQ	7	2		-	
3	Лек 5	Основы XML	7	2	-	-	
3	Пр 11	Основы XML	7	2	-	-	Отчет по практической работе №3
3	Пр 12	Основы XML	7	2		-	
3	Лек 6	Юнит тесты	7	2	-	-	
3	Пр 13	Юнит тесты	7	2	-	-	Отчет по практической работе.№4
3	Пр 14	Юнит тесты	7	2		-	
3	Пр 15	Юнит тесты	7	2		-	
3	Пр 16	Юнит тесты	7	2		-	
3	Лек 7	Типичные ошибки при разработке программного обеспечения	7	2	-	-	
3	СР	Типичные ошибки при разработке программного обеспечения	7	20	-	-	
3	Пр 17	Типичные ошибки при разработке программного обеспечения	7	2	-	-	Отчет по практической работе.№5
3	Пр 18	Типичные ошибки при разработке программного обеспечения	7	2		-	
3	Пр 19	Типичные ошибки при разработке программного обеспечения	7	2		-	
4	Лек 8	Работа с XML	7	2	-	-	

4	СР	Работа с XML	7	19,75	-	-	
4	Пр 20	Работа с XML	7	2	-	-	Отчет по практической работе.№6
4	Пр 21	Работа с XML	7	2		-	
4	Пр 22	Работа с XML	7	2		-	
4	Пр 23	Работа с XML	7	2		-	
	ПА		7	0,25		-	
	Пр 24	Зачет	7	2	-	-	Вопросы к зачету
				Итого:	180		

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.

обучаемый 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

обучаемым 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 обучаемым на практических занятиях как в результате обсуждения и анализа лекционного материала, так и с помощью решения проблемных ситуаций, задач. При этих

условиях обучаемый 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 обучаемый 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, обучаемый ликвидирует имеющиеся пробелы в знаниях, углубляет, систематизирует и упорядочивает свои знания. На экзамене обучаемый демонстрирует то, что он приобрел в процессе обучения по конкретной учебной дисциплине.

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

Необходимо ориентировать обучаемых 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
7	ПК-5	Вопросы к зачету Отчеты по практическим занятиям

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

7.2.1. Вопросы к тестовому заданию

Комплект отчетов по практическим работам

Практическая работа №1 «Коллекции и обобщенные типы».

Цель работы: Ознакомиться с основами работы с коллекциями и обобщенными типами в языке C#. Научиться создавать, модифицировать и использовать коллекции для хранения и обработки данных.

Задания к практической работе:

1. Создание обобщенного класса:

- Реализовать обобщенный класс `Container<T>`, который будет представлять собой простую коллекцию элементов.

- Класс должен содержать методы для добавления, удаления и получения элементов по индексу.

- Реализовать метод 'Count', который возвращает количество элементов в контейнере.

2. Использование стандартных коллекций:

- Создать класс 'Student', который будет содержать поля:

- Имя (string)

- Возраст (int)

- Оценка (double)

- Создать список ('List<Student>') для хранения объектов класса 'Student'.

- Реализовать методы для добавления, удаления и поиска обучаемых по имени.

3. Сортировка и фильтрация:

- Реализовать метод для сортировки списка обучаемых по оценке (по возрастанию или убыванию).

- Реализовать метод для фильтрации обучаемых по возрасту (например, вернуть всех обучаемых старше 18 лет).

4. Работа с LINQ:

- Используя LINQ, реализовать метод, который возвращает список обучаемых с оценкой выше заданного порога.

- Пример использования LINQ: 'var highScorers = students.Where(s => s.Grade > threshold).ToList();'

5. Тестирование:

- Написать тестовый класс, в котором создать несколько объектов 'Student', добавить их в список, выполнить сортировку и фильтрацию.

- Вывести результаты на консоль.

Итог:

В результате выполнения данной практической работы обучаемые должны научиться:

- Создавать и использовать обобщенные классы.

- Работать со стандартными коллекциями в C#.

- Применять LINQ для обработки данных.

- Понимать основы работы с коллекциями и их применение в реальных задачах.

Дополнительные рекомендации:

- Используйте комментарии к коду для пояснения логики реализации.

- Протестируйте все методы на различных наборах данных, чтобы убедиться в их корректности и эффективности.

Задания к практической работе №2 «Бинарное дерево поиска».

Цель работы: развить навыки анализа и обработки данных в формате XML

- Разработать тип «бинарное дерево поиска» для хранения результатов тестов, выполненных обучающимися, и их предоставления в упорядоченном виде.
- Предусмотреть возможность хранения данных любого типа, поддерживающего сравнения.
- Информация о обучающемся может содержать поля, хранящие имя обучающегося, название теста, дату его прохождения и оценку теста для данного обучающегося
 - Реализовать обобщенный стандартный интерфейс, чтобы информацию о обучающемся можно было сравнить с информацией о другом обучающемся.
- Предусмотреть следующие конструкторы «дерева»:
 - Который принимает объект стандартного обобщенного типа для сравнения двух объектов одного типа, и применял его для сравнения добавляемых в дерево элементов.
 - Который принимает объект типа стандартной обобщенной коллекции (не список!), и добавлял элементы коллекции в дерево (пример смотрите в «исходниках» конструкторов типа List<T>).

- Который принимает и коллекцию элементов для добавления в дерево, и объект с помощью, которого будет выполнять сравнения.
- ❖ Итого: если указан «компаратор», то он используется для сравнения элементов, иначе – используется «сравнение, реализованное внутри элемента».
- Созданный класс «дерево» должен реализовывать обобщённый интерфейс перечислителя. При реализации нужно применить «итератор» (реализовать с помощью yield, см. важно п.3)).
- В «дереве» реализовать метод, который возвращает объект для итерации по дереву в обратном порядке.
Например, GetReversedEnumerator()(реализовать с помощью yield, см. важно п.3))
- В качестве UI-интерфейса использовать консольное приложение с интерфейсом командной строки, WinForms или WPF-приложение. В приложении нужно:
- Применить созданное «дерево» для хранения чисел из целочисленного массива, при этом в дерево они сразу должны храниться в порядке убывания.
 1. Добавить ещё несколько чисел в дерево.
 2. Вывести все числа «дерева» в том порядке, в котором они хранятся в дереве.
- Применить созданное «дерево» для хранения результатов прохождения тестов обучающимися, при этом элементы в дереве **сразу** должны храниться в порядке убывания «оценок (баллов)» по тесту.
 1. Вывести информацию о обучающихся в том порядке, в котором они хранятся в дереве.
- Применить созданное «дерево» для хранения строк:
 1. Строки добавить из списка строк;

Вывести строки в обратном порядке, используя метод «дерева», который возвращает «перечислитель» для прохода «дерева» в обратном порядке.

Практическая работа №3 «Введение в LINQ ». 2

Цель работы: освоение языка запросов LINQ для эффективной работы с данными в дотнет приложениях.

Задания к практической работе

- Разработать тип, реализующий алгоритм Евклида для вычисления НОД двух целых чисел (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 Эвклида) для расчета НОД двух целых чисел (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.

Практическая работа №4 «XML и платформа Microsoft .NET Framework».

Цель работы: изучение и применение технологий XML в контексте разработки на платформе Microsoft .NET Framework

Задания к практической работе №4:

- Разработать класс «треугольник» для работы с треугольником, заданным на плоскости сторонами a , b , c .
- Предусмотреть возможность проверки существования треугольника, реализовать функции подсчета его площади и периметра.
- В качестве UI-интерфейса использовать консольное приложение с интерфейсом командной строки, WinForms или WPF-приложение.
- Создать unit-тесты для тестирования методов разработанного типа.

Практическая работа №5 «Основы XML».

Цель работы: освоение базовых принципов работы с форматом XML и его применении для структурирования и обмена данными .

Задания к практической работе №5:

- Разработать класс «вектор» для работы с трехмерными векторами.
- Перегрузить для класса операции, допустимые для работы с трехмерными векторами.
- Создать unit-тесты для тестирования разработанных методов.
- В качестве UI-интерфейса использовать консольное приложение с интерфейсом командной строки, WinForms или WPF-приложение.
- Разработать класс «многочлен» для работы с многочленами от одной переменной.
- Перегрузить для класса операции, допустимые для работы с многочленами.
- 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 A4 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:

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

Практическая работа №6 «Юнит-тесты» 4

Цель работы: Ознакомиться с основами написания юнит-тестов в C# с использованием библиотеки NUnit или MSTest. Научиться тестировать методы и классы, созданные в предыдущих практических работах.

Задания к практической работе:

1. Настройка окружения:
 - Установить библиотеку NUnit или MSTest через NuGet Package Manager в проекте.
 - Создать новый проект для юнит-тестов, который будет ссылаться на проект с реализацией коллекций и обобщенных типов.
2. Тестирование класса `Container<T>`:
 - Написать юнит-тесты для методов класса `Container<T>`:
 - Тест на добавление элемента.
 - Тест на удаление элемента.
 - Тест на получение элемента по индексу.
 - Тест на метод `Count`, чтобы убедиться, что он возвращает правильное количество элементов.
3. Тестирование класса `Student`:
 - Написать юнит-тесты для методов работы со списком обучаемых:
 - Тест на добавление в список.
 - Тест на удаление по имени.
 - Тест на поиск по имени (должен возвращать правильного обучаемого).
 - Тест на сортировку обучаемых по оценке (проверить порядок).
4. Тестирование фильтрации и сортировки:

- Написать юнит-тесты для методов фильтрации и сортировки:
 - Тест на фильтрацию обучаемых по возрасту (должен возвращать только старше 18 лет).
 - Тест на сортировку по оценке (проверить, что список отсортирован правильно).

5. Тестирование методов LINQ:

- Написать юнит-тесты для метода, который возвращает список обучаемых с оценкой выше заданного порога:
- Проверить, что метод возвращает правильный список обучаемых.
 - Проверить, что метод корректно обрабатывает случай, когда нет обучаемых с оценкой выше порога.

6. Проверка исключений:

- Написать тесты для проверки выбрасываемых исключений:
 - Проверить, что при попытке получить элемент по недопустимому индексу выбрасывается исключение.
 - Проверить, что при удалении несуществующего обучаемого выбрасывается соответствующее исключение.

Итог:

В результате выполнения данной практической работы обучаемые должны научиться:

- Понимать основы написания юнит-тестов.
- Использовать библиотеки NUnit или MSTest для создания и выполнения тестов.
- Проводить тестирование различных аспектов своих классов и методов.

Дополнительные рекомендации:

- Используйте атрибуты `[Test]` и `[TestFixture]` для организации тестов в NUnit или аналогичные атрибуты в MSTest.
- Применяйте принципы TDD (разработка через тестирование) при написании новых функций в будущем.
- Обязательно проверяйте результаты тестов после их написания и убедитесь, что все они проходят успешно.

Практическая работа №7 «Типичные ошибки при разработке программного обеспечения» 5

Цель работы: Ознакомиться с типичными ошибками, которые могут возникнуть при разработке программного обеспечения, и научиться их выявлять и исправлять. Научиться применять лучшие практики для предотвращения ошибок.

Задания к практической работе:

1. Анализ кода на наличие ошибок:

- Предоставить обучаемым фрагмент кода, содержащий типичные ошибки (например, ошибки логики, неправильное использование коллекций, отсутствие проверки на null и т.д.).
 - Описание проблемы.
 - Возможные последствия.
 - Способы исправления.

2. Использование исключений:

- Создать класс `'Student'` с методами для добавления и удаления обучаемых из списка.
 - Внедрить в методы обработку исключений (например, выбрасывать `'ArgumentNullException'`, если имя пустое).
- Написать тесты для проверки корректности обработки исключений.

3. Проблемы с производительностью:

- Предоставить код, который использует неэффективные алгоритмы (например, многократный перебор списка для поиска элементов).
- Попросить обучаемых оптимизировать код, предложив более эффективные решения (например, использование `'HashSet'` для быстрого поиска).

4. Проблемы с многопоточностью:

- Создать пример класса, который использует многопоточность без должной синхронизации (например, несколько потоков изменяют один и тот же список).
- Попросить обучаемых выявить потенциальные проблемы (например, состояние гонки) и предложить способы их решения (например, использование `'lock'` или других механизмов синхронизации).

5. Неправильное использование коллекций:

- Предоставить код с неправильным использованием коллекций (например, использование `'List<T>'` вместо `'Dictionary<TKey,TValue>'` для хранения пар ключ-значение).
- Попросить обучаемых переписать код с использованием более подходящих коллекций и объяснить выбор.

6. Тестирование на наличие ошибок:

- Написать юнит-тесты для методов класса `'Student'`, чтобы проверить корректность работы методов добавления и удаления имен.
- Включить тесты на проверку обработки исключений и тесты на производительность.

7. Документация и комментарии:

- Попросить обучаемых написать документацию для класса `'Container<T>'` и класса `'Student'`, включая описание методов и параметров.
- Обратить внимание на важность написания понятных комментариев в коде.

Итог:

В результате выполнения данной практической работы обучаемые должны научиться:

- Выявлять типичные ошибки в коде.
- Применять лучшие практики разработки для предотвращения ошибок.
- Понимать важность тестирования и документации в процессе разработки программного обеспечения.

Дополнительные рекомендации:

- Обсудите с группой примеры реальных ошибок из практики разработки программного обеспечения.
- Подчеркните важность постоянного обучения и улучшения навыков программирования для минимизации ошибок в будущем.

Практическая работа №8 «Работа с XML» 6

Цель работы: Ознакомиться с основами работы с XML в C#. Научиться создавать, читать, изменять и сохранять XML-документы. Понять, как использовать XML для хранения и обмена данными.

Задания к практической работе:

1. Создание XML-документа:

- Написать программу, которая создает XML-документ для хранения информации о обучаемых. Каждый обучаемый должен содержать следующие поля: 'Id', 'Name', 'Age', 'Grade'.
- Сохранить созданный XML-документ в файл 'students.xml'.

2. Чтение XML-документа:

- Написать программу, которая загружает созданный ранее файл 'students.xml' и выводит информацию о каждом обучаемом на консоль.
- Использовать класс 'XmlDocument' или LINQ to XML для чтения данных.

3. Изменение данных в XML:

- Реализовать метод, который позволяет обновлять информацию о студенте по его 'Id'. Например, изменить оценку или возраст.
- После изменения сохранить обновленный XML-документ обратно в файл.

4. Удаление данных из XML:

- Реализовать метод, который позволяет удалять имя из файла по его 'Id'.
- После удаления сохранить обновленный XML-документ.

5. Поиск данных в XML:

- Написать метод, который принимает имя и возвращает информацию о нем из файла 'students.xml'. Если элемент не найден, выводить соответствующее сообщение.
- Использовать LINQ to XML для выполнения поиска.

6. Валидация данных:

- Создать XSD-схему для вашего XML-документа, чтобы определить структуру и типы данных.
- Реализовать проверку на соответствие загружаемого XML-файла этой схеме с помощью класса 'XmlReader'.

7. Работа с атрибутами:

- Модифицировать структуру вашего XML-документа так, чтобы некоторые данные (например, 'Grade') хранились в атрибутах элементов '<Student>'.
- Обновить методы чтения и записи данных соответственно.

8. Сериализация и десериализация объектов:

- Создать класс 'Student' и использовать сериализацию для сохранения списка в формате XML.
- Реализовать десериализацию для загрузки списка из файла.

Итог:

В результате выполнения данной практической работы обучаемые должны научиться:

- Создавать и манипулировать XML-документами в C#.
- Использовать различные подходы для работы с данными (XmlDocument, LINQ to XML).
- Понимать важность валидации данных и использования схемы XSD.
- Применять сериализацию и десериализацию объектов для работы с данными.

Дополнительные рекомендации:

- Обсудите с группой примеры использования XML в реальных приложениях (например, конфигурационные файлы, обмен данными между системами).
- Поощряйте обучаемых экспериментировать с различными способами работы с XML (например, использование других библиотек).
- Подчеркните важность правильного структурирования данных при работе с форматом XML.

Критерии оценки:

Максимальное количество баллов, начисляемое за задание, выполненное на практическом занятии, равно 15. При условии выполнения менее 40 процентов от запланированного объема работы – 0 баллов, от 40 до 60 процентов – соответственно 6 балла, от 60 до 80 процентов – 10 баллов, от 80 до 100 процентов – 15 баллов.

7.3. Оценочные средства для промежуточной аттестации по итогам освоения дисциплины

7.3.1. Вопросы к промежуточной аттестации

Семестр 5

№ п/п	Вопросы к зачету
1.	Понятие инкапсуляции как одного из основных принципов объектно-ориентированного программирования.
2.	Применение свойств класса. Внутреннее представление свойств.
3.	Понятие наследования как одного из основных принципов объектно-ориентированного программирования. Примеры реализации.
4.	Понятие полиморфизма как одного из основных принципов объектно-ориентированного программирования. Примеры реализации.
5.	Принципы реализации графического интерфейса программных приложений.
6.	Проектирование базы данных с использованием MS SQL - сервера.
7.	Пространства имен ADO.NET. Типы пространства имен System.Data.
8.	Тип DataView. Возможности класса DataSet.
9.	Статические методы и методы экземпляров.
10.	Статические данные.
11.	Библиотека базовых классов .NET. Сборки. MSIL.
12.	Статические члены System.Object. Упаковка и распаковка.
13.	Базовый класс System.Array. Примеры алгоритмов обработки.
14.	Применение System.StringBuilder. Примеры использования.
15.	Ссылки класса на самого себя, перенаправление вызовов конструктора с использованием this. Определение открытого интерфейса по умолчанию. Примеры реализации.
16.	Реализация инкапсуляции при помощи традиционных методов доступа и изменения.
17.	Тип DataColumn. Первичный ключ таблицы. Тип DataRow. Тип DataTable.
18.	Использование класса DataRelation для моделирования отношений между таблицами.
19.	Статические конструкторы. Статические поля.
20.	Принципы построения приложений обработки баз данных.
21.	Базовый и производные классы. Работа с конструктором базового класса. Примеры реализации.
22.	Работа с конструктором базового класса. Примеры реализации.
23.	Запрет наследования. Применение модели включения-делегирования. Примеры реализации.
24.	Механизм позднего связывания. Примеры реализации.
25.	Абстрактные классы. Абстрактные методы.
26.	Конструкторы и деструкторы.
27.	Философия .NET.
28.	Открытые, закрытые и защищенные методы класса. Примеры реализации.
29.	Использование памяти для структурных (числовые данные, перечисления и структуры) и ссылочных (классы и интерфейсы) типов.
30.	Класс System.Object. Замещение методов System.Object.
31.	Определение пользовательских методов класса.
32.	Модификаторы уровня доступа к методам.
33.	Модификаторы для параметров методов.
34.	В связи с чем язык XML получил широкое распространение?

35.	Что представляет собой язык разметки?
36.	Какими особенностями обладает язык разметки XML?
37.	Что представляет собой платформа Microsoft .NET Framework?
38.	Каково назначение платформы Microsoft .NET Framework?
39.	Из каких компонентов состоит платформа Microsoft .NET Framework?
40.	Каковы функции среды CLR?
41.	Что представляет собой библиотека классов Microsoft .NET Framework?
42.	На основе каких стандартов строится web-служба XML в рамках платформы Microsoft .NET Framework?
43.	Какие API для реализации работы с XML разработаны в рамках платформы Microsoft .NET Framework?
44.	Что представляет собой XML-документ?
45.	Из каких разделов состоит XML-документ?
46.	Какую информацию содержит декларация XML-документа?
47.	Какая информация размещается в прологе XML-документа?
48.	Как строятся элементы XML-документа?
49.	Каковы синтаксические правила создания атрибутов XML документа?
50.	Для чего в XML-документах используются комментарии?
51.	Чем парсируемые данные в XML-документе отличаются от непарсируемых данных?
52.	Что представляют собой сущности XML-документа?
53.	Для чего в XML-документе необходимо пространство имен?
54.	Для чего необходима валидация XML-документа?
55.	Что определяет DTD-схема документа?
56.	Какие инструкции используются в DTD-схеме для описания грамматики XML-документа?
57.	Каковы достоинства и недостатки DTD-схем?
58.	Что представляет собой XML(XDR)-схема?
59.	Какие структурные элементы и характеристики XML-документа описывает XML(XDR)-схема?
60.	В чем заключается разница между простыми и комплексными типами элементов в XML(XDR)-схемах?
61.	Какие типы элементов, описываемых с помощью XML(XDR)-схем, могут содержать атрибуты?
62.	Что определяют примитивы XML(XDR)-схем?
63.	В чем заключается основная разница между DTD и XML(XDR)схемами?
64.	Для чего разработан язык XPath? Как прописываются пути в XPath?
65.	Что определяет ось выборки XPath?
66.	С какой целью в XPath используются предикаты?
67.	Чем определяется контекст выражения в XPath?
68.	Каково основное назначение технологии XSLT?
69.	Как строятся шаблоны преобразований в XSLT?
70.	Какова структура шаблона XSLT?
71.	Как с помощью языка XSLT создаются новые структурные единицы документа (элементы, атрибуты и т.п.)?
72.	Каков алгоритм преобразования XML-документа с помощью языка XSLT?

7.3.2. Критерии и нормы оценки

Семестр	Форма проведения	Критерии и нормы оценки
---------	------------------	-------------------------

	промежуточной аттестации		
7	Зачет	«зачтено»	оценка «зачтено» выставляется обучающемуся, если он исчерпывающе и грамотно дал ответ на 1 вопрос, а на другой только тезисные высказывания или допустил небольшие неточности при ответе на вопросы экзаменационного билета и дал краткие ответы на дополнительные вопросы;
		«не зачтено»	оценка «не зачтено» выставляется обучающемуся, если он не дал ответ на вопросы экзаменационного билета или в ответе содержались фундаментальные ошибки

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 FREEDOM COLLECTION (Полнотекстовая коллекция электронных журналов Elsevier B.V.) <https://www.sciencedirect.com/>
- 2 Nano Database <http://nano.nature.com/>
- 3 Springer Materials <http://materials.springer.com/>
- 4 Springer Nature Protocols and Methods <https://experiments.springernature.com/sources/springer-protocols>
- 5 zbMath <https://zbmath.org/>
- 6 Springer Nature (Полнотекстовая коллекция журналов) <https://www.springernature.com/gp/products>
- 7 Springer eBooks (Полнотекстовая коллекция электронных книг издательства Springer Nature) <https://link.springer.com/>
- 8 ORBIT INTELLIGENCE (Патентная база компании QUESTEL) <http://www.orbit.com/>
- 9 CSD-ENTERPRISE (База данных компании CAMBRIDGE CRYST ALLOGRAPHIC DATA CENTER) <https://www.ccdc.cam.ac.uk/structures/>

8.4. Перечень программного обеспечения

№ п/п	Наименование ПО	Реквизиты договора (дата, номер, срок действия)
1	WinPro 10 RUS Upgrd OLP NL Acdmc	Договор № 757 от 04.07.2018, срок действия - бессрочно; Контракт № 1653 от 14.12.2018, срок действия – бессрочно
2	Office Stdandard 2013 Russian OLP NL AcademicEdition	Контракт № 690 от 19.05.2015, срок действия - бессрочно)

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	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 курсового проектирования (выполнения курсовых работ). Учебная аудитория для проведения групповых и индивидуальных консультаций. Учебная аудитория для проведения занятий текущего контроля и промежуточной аттестации. (УЛК-402).	Системные блоки (HP ProDesk), мониторы (Samsung), коммутатор (D-Link), столы ученические, столы компьютерные, стулья, доска аудиторная, экран.
2	Помещение для самостоятельной работы обучающихся (УЛК-105).	Стол, стулья, стеллажи (в т.ч. выставочные) с книгами, компьютеры,

№ 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
		мобильные рабочие места.
3	Помещение для самостоятельной работы обучающихся (УЛК-406).	Столы компьютерные, стулья, микрокомпьютеры raspberry pi 32 bit.