ФГАОУ ВО «КФУ им. В.И. Вернадского»

Физико-технический институт (структурное подразделение)

|  |
| --- |
|  |

Кафедра компьютерной инженерии и моделирования

Усеинов Ильяс Эльнур оглы

отчет по практической работе №1  
по дисциплине **«ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ»**

09.03.04 "Программная инженерия"

Оценка -   
"удовлетворительно"  
"хорошо","отлично"



Симферополь, 2024

**Практическая работа №1.   
Тема: Использование программных конструкций C#**

**Цель работы:** Научиться создавать простейшие консольные и WPF приложения на языке C# в среде Visual Studio, изучить возможность создания самодокументируемых приложений.

Научиться преобразовывать различные типы данных в C#, познакомиться с типом данных Decimal, научиться грамотно использовать циклы для итерационных вычислений с контролем погрешности, обрабатывать события нажатия клавиш, научиться использовать классы String, StringBuilder, научиться создавать самодокументируемые XML справочные файлы.

**Описание ключевых понятий:**   
**CLR** -общеязыковая исполнительная среда CLR (Common Language Runtime, виртуальная машина), обеспечивает выполнение сборки и является основным компонентом .NET Framework. Под Виртуальной Машиной понимают абстракцию операционной системы высокого уровня, которая обеспечивает выполнение (управляемого) программного кода. Управляемый код – программный код, который при своём выполнении способен использовать службы, предоставляемые CLR. Сама CLR состоит из двух главных компонентов: ядра (mscore.dll) и библиотеки базовых классов FCL (mscorlib.dll) .   
**FCL** - .NET Framework class library - коллекция классов, интерфейсов и типов, которые предоставляют базовые функции для приложений .NET.  
**CLS** - общеязыковые спецификации CLS –набор соглашений и стандартов, которые должны поддерживаться всеми языками программирования для взаимодействия в рамках платформы .NET.

**Решение (solution)-** контейнер для одного или нескольких связанных проектов, а также сведения о сборке, параметры окна Visual Studio и любые другие файлы, которые не связаны с определенным проектом..

**Проект (project)** -  все файлы, которые будут скомпилированы в исполняемую программу, библиотеку или веб-сайт. Эти файлы могут включать исходный код, значки, изображения, файлы данных и многое другое.   
**namespace** - область, которая организует код в логические группы и предотвращает конфликты имен между различными компонентами и библиотеками.

**Сборка (assembly)** — основной компонент приложений .NET, который содержит скомпилированный код и метаданные. Сборка может быть библиотекой или исполняемым файлом.  
**MSIL (Microsoft Intermediate Language, IL)** — промежуточный язык, в который компилируются программы на .NET перед их выполнением CLR. MSIL является платформо-независимым.

**Управляемый код** — код, который выполняется под управлением CLR, обеспечивающей такие функции, как сборка мусора, безопасность и межъязыковое взаимодействие.

**WPF** - **Windows Presentation Foundation** — аналог WinForms, система для построения клиентских приложений [Windows](https://ru.wikipedia.org/wiki/Windows) с визуально привлекательными возможностями взаимодействия с пользователем, графическая подсистема в составе .NET Framework (начиная с версии 3.0), использующая язык XAML. В основе WPF лежит векторная система визуализации, основанная на DirectX.

**Дизассемблер -** транслятор, преобразующий машинный код, объектный файл или библиотечные модули в текст программы на языке ассемблера.

**Windows.Forms и WPF проекты** — два типа проектов для создания графических пользовательских интерфейсов в Windows. Windows.Forms — это более старый фреймворк, а WPF (Windows Presentation Foundation) предоставляет более современные и мощные средства для создания интерфейсов.

**Массивы** — коллекции элементов одного типа, которые хранятся в памяти подряд.

**Перечисления (enums)** — тип, который представляет собой набор именованных констант.

**Структуры (struct)** — тип данных, который может содержать поля и методы, но является типом-значением. В отличие от классов, структуры не поддерживают наследование.

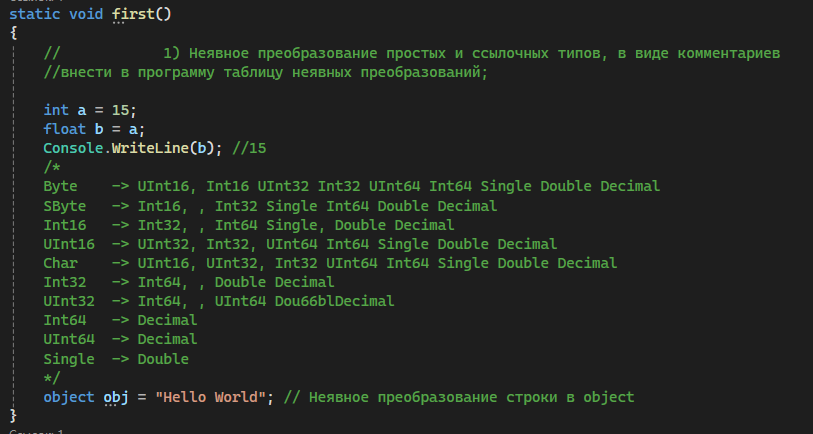
**Упаковка (boxing**) - преобразование объекта значимого типа (например, типа int) к типу object. При упаковке общеязыковая среда CLR обертывает значение в объект типа System.Object и сохраняет его в управляемой куче (хипе).

**Перед выполнением лабораторной работы изучена следующая литература:**

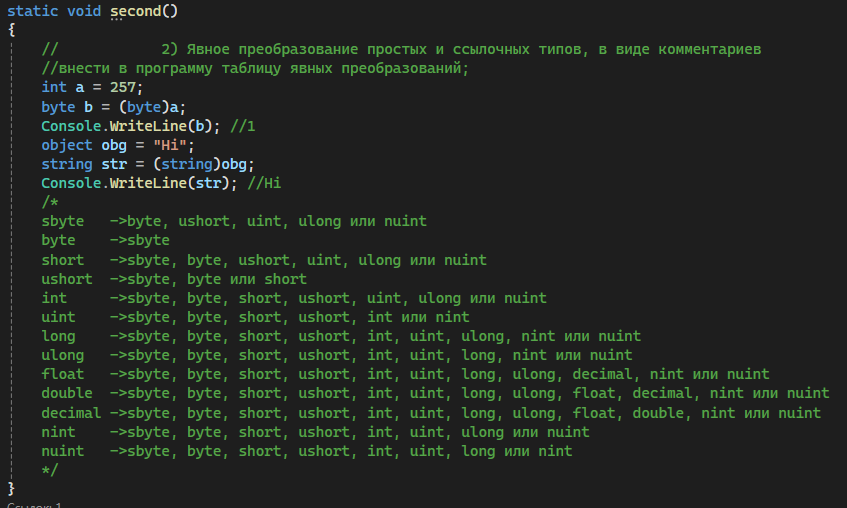
1. Прослушана лекция.
2. Материалы сайта metanit.com.
3. Получены начальные сведения о спецификации языка C#.

**Выполнены 3 задания, описанных в методических указания к выполнению лабораторных работ.**

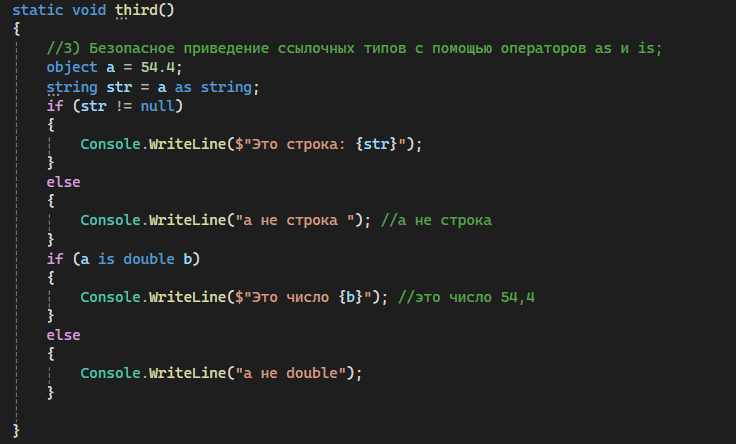
Задание 1.  
1) Неявное преобразование простых и ссылочных типов, в виде комментариев внести в программу таблицу неявных преобразований;



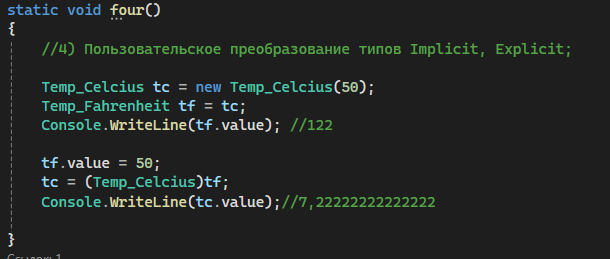
2) Явное преобразование простых и ссылочных типов, в виде комментариев внести в программу таблицу явных преобразований;

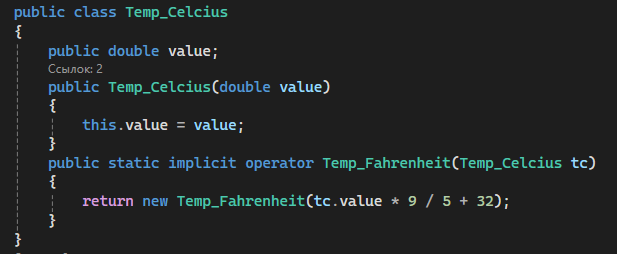


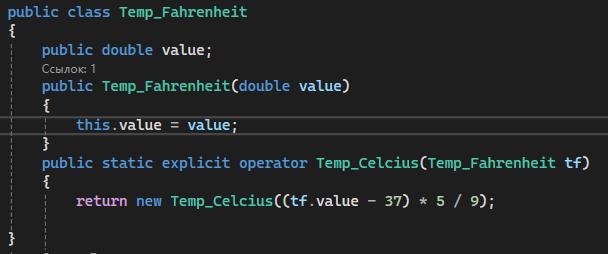
3) Безопасное приведение ссылочных типов с помощью операторов as и is;



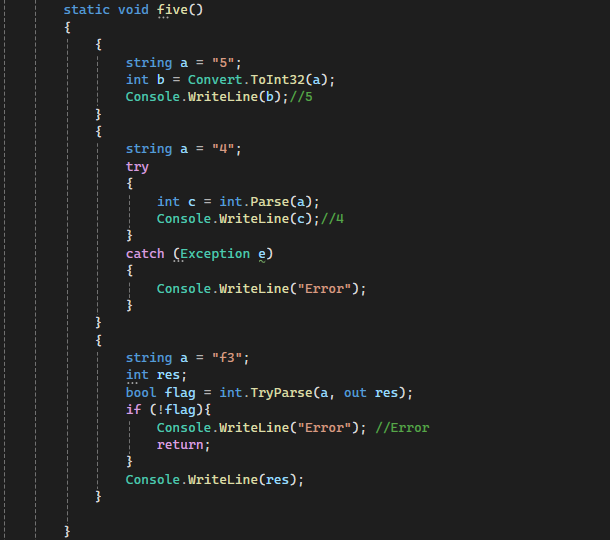
4) Пользовательское преобразование типов Implicit, Explicit;



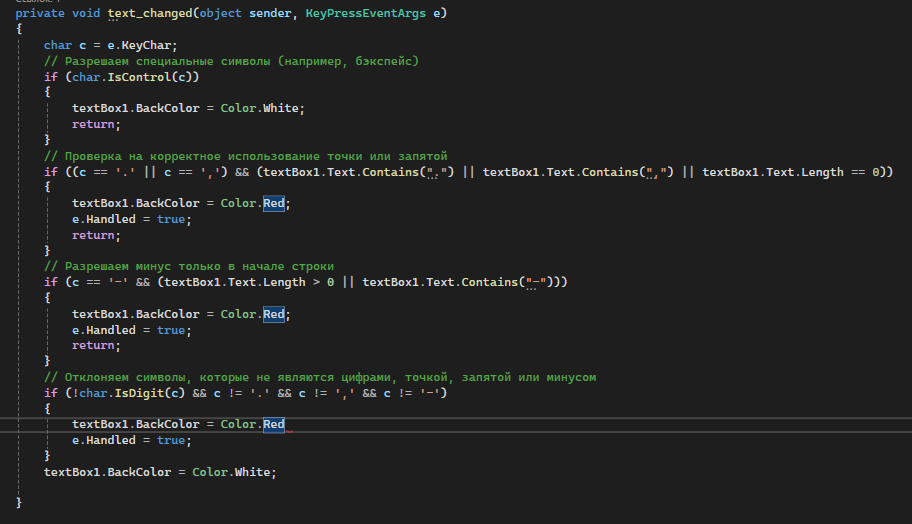


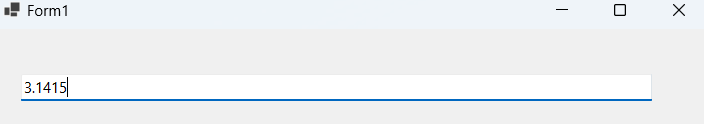


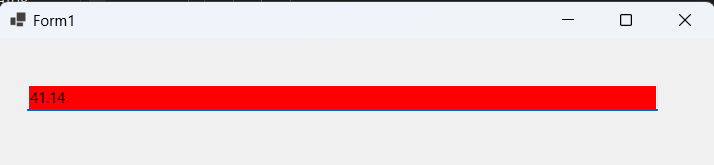
5) Преобразование с помощью класса Convert и преобразование строки в число с помощью методов Parse, TryParse.



6) Написать программу, позволяющую ввод в текстовое поле TextBox только символов, задающих правильный формат вещественного числа со знаком.

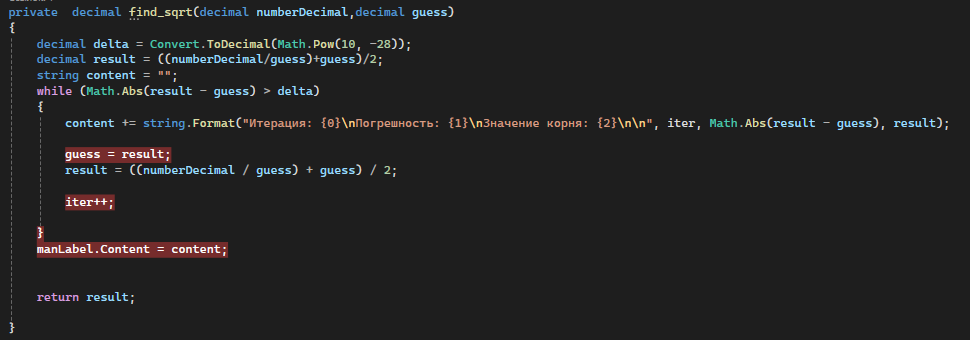
****

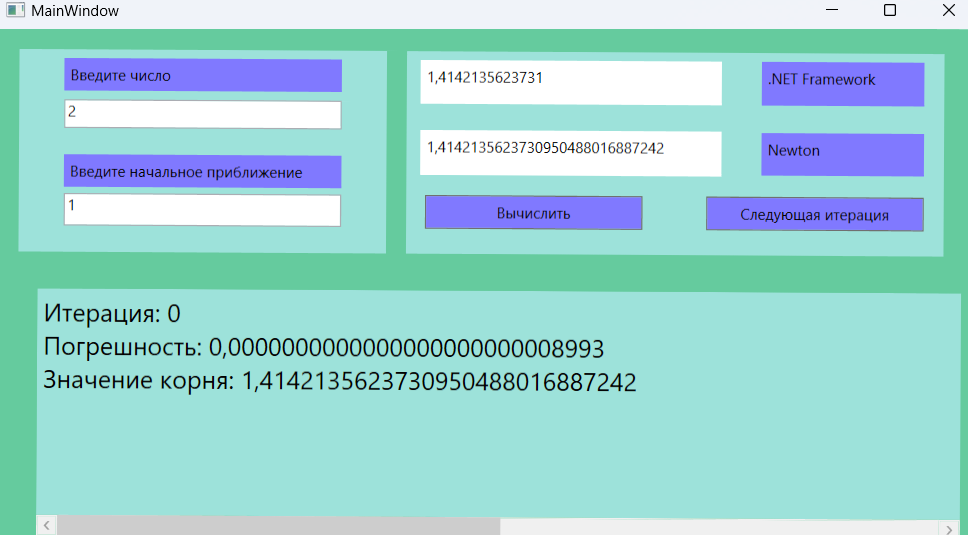
****

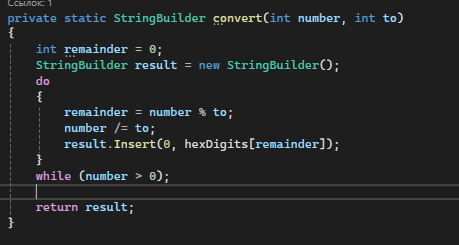
Попытка обойти условия, добавив еще одну точку  
****

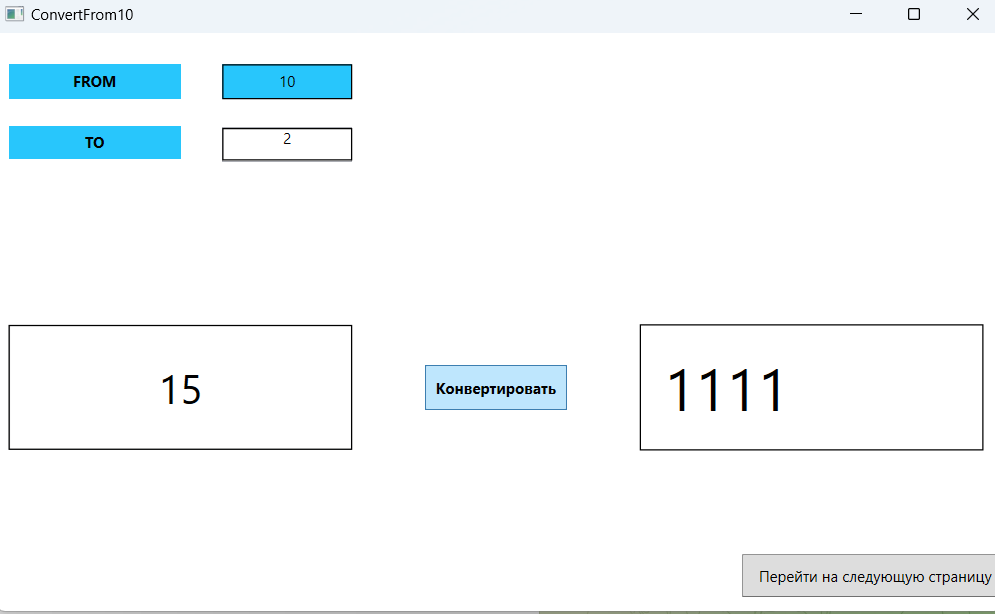
Задание 2: Вычислить квадратный корень с контролем точности

Скриншот кода и интерфейса приведены ниже

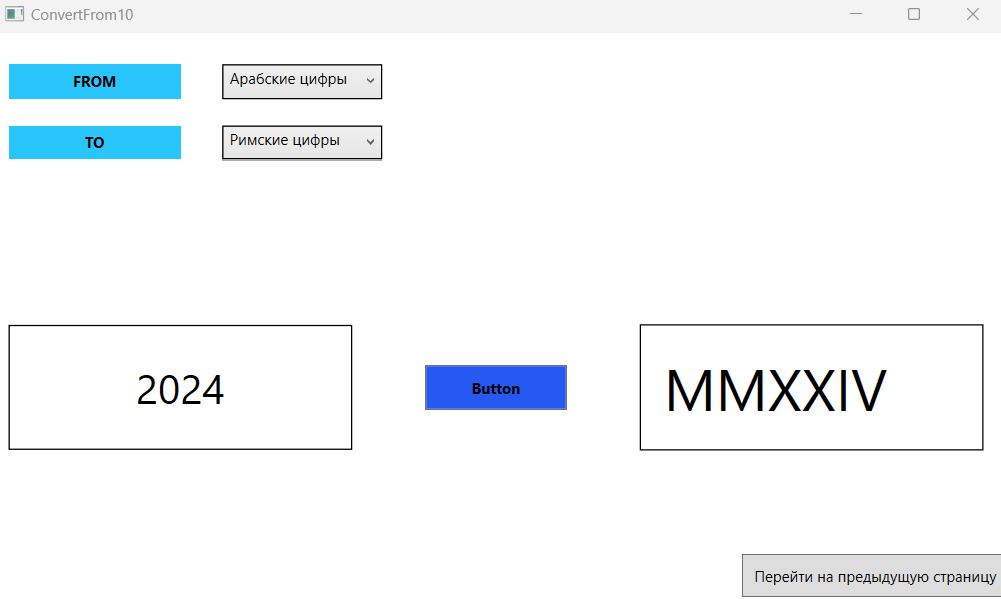


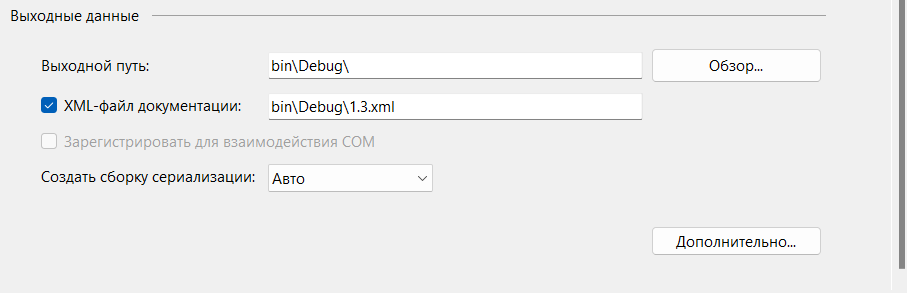
  
Задание 3: Конвертация целочисленных данных в бинарные.





Программа, реализующая перевод десятичного целого числа из арабской системы исчисления в римскую и обратно.

  
Задание 4: Создание документации для приложения

  
Текст файла документации приведен ниже

**Представлены 4 проекта, реализованных в Visual Studio Community 2022.  
Проекты представлены преподавателю в электронной форме, продемонстрирована их работоспособность, разъяснены детали программного кода.**

**Вопросы, заданные преподавателем:**

* 1. Может ли быть потеряна точность в неявных преобразованиях из int, uint, long, или ulong в float и из long или ulong в double.
  2. Как осуществить неявное преобразование в тип char.
  3. Как осуществить неявные преобразования между типами с плавающей запятой и типом decimal?
  4. Как контролируется точность при вычислении квадратного корня методом Ньютона?
  5. Зачем в работе используется класс StringBuilder?