Министерство образования и науки Российской Федерации

ФГАОУ ВО «КФУ им. В.И. Вернадского»

физико-технический институт (структурное подразделение)

|  |
| --- |
|  |

Кафедра компьютерной инженерии и моделирования

Андрияшин Алексей

&&

Докторов Станислав

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

Направление подготовки:

09.03.01 "Информатика и вычислительная техника",

Оценка -   
""



Симферополь, 2023

**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:**   
интегрированная среда разработки IDE (Integrated Development Envirionment) Visual Studio, общеязыковая исполнительная среда CLR, библиотека классов FCL, общеязыковые спецификации CLS, решение (solution), проект (project), пространство имен (namespace), сборка (assembly), MSIL (Microsoft Intermediate Language, IL), управляемый код, двухэтапная компиляция, дизассемблер, обозреватель решений (Solution Explorer), Windows.Forms и WPF проекты, класс, статический тип, динамический тип, встроенные типы, типы-значения, ссылочные типы, фундаментальные типы: логический, символьный, целый, с плавающей точкой, void, указатели, ссылки, массивы, перечисления, структуры, классы, преобразования типов: упаковка Boxing, неявное преобразование, явное преобразование, класс Convert.

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

1. Презентация лектора курса: «Основы Net Framework» и «Использование  
   конструкций языка С#».
2. Прослушаны видеолекции 1-3 на BBB.
3. Прочитаны 1-3 лекции из облака mail ru.
4. Просмотрены практические примеры из интернета.
5. Получены начальные сведения о спецификации языка C#.
6. Изучен разделы 1-2 и частично разделы 3-4 сайта Metanit.com.

**Выполнены 3 задания, описанных в методических указания к выполнению лабораторных работ.**Задание 1: Приведение и преобразование типов

Скриншот кода и интерфейса приведены ниже  
Задание 2: Вычислить квадратный корень с контролем точности

Скриншот кода и интерфейса приведены ниже  
Задание 3: Конвертация целочисленных данных в бинарные.

Скриншот кода и интерфейса приведены ниже  
Задание 4: Создание документации для приложения  
Текст файла документации приведен ниже

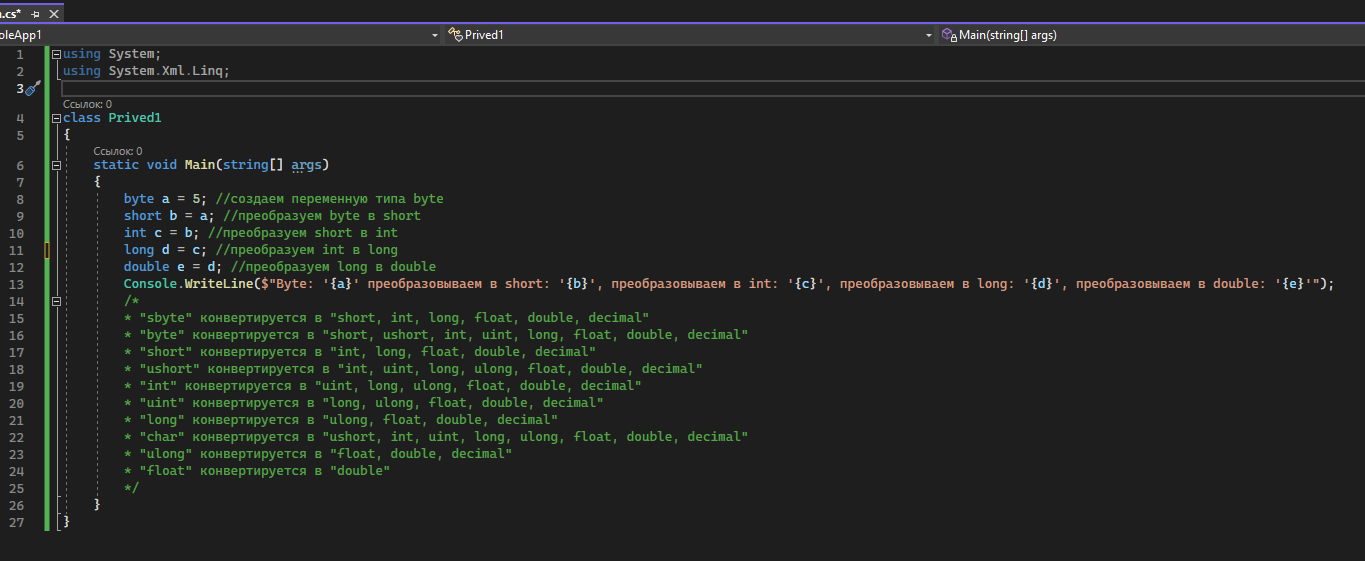
Выполнено бонусное задание в первом задании.

**Задание 1**

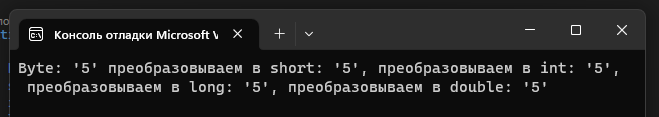
**Приведение и преобразование типов**

1-А) Неявное преобразование простых типов:

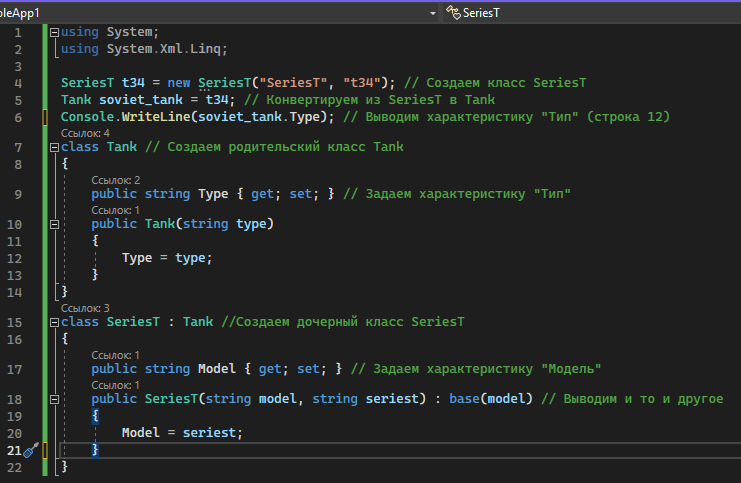
Скриншот кода приведен ниже



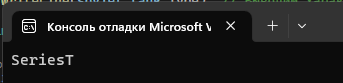
Результат:



1-Б) Неявное преобразование ссылочных типов:

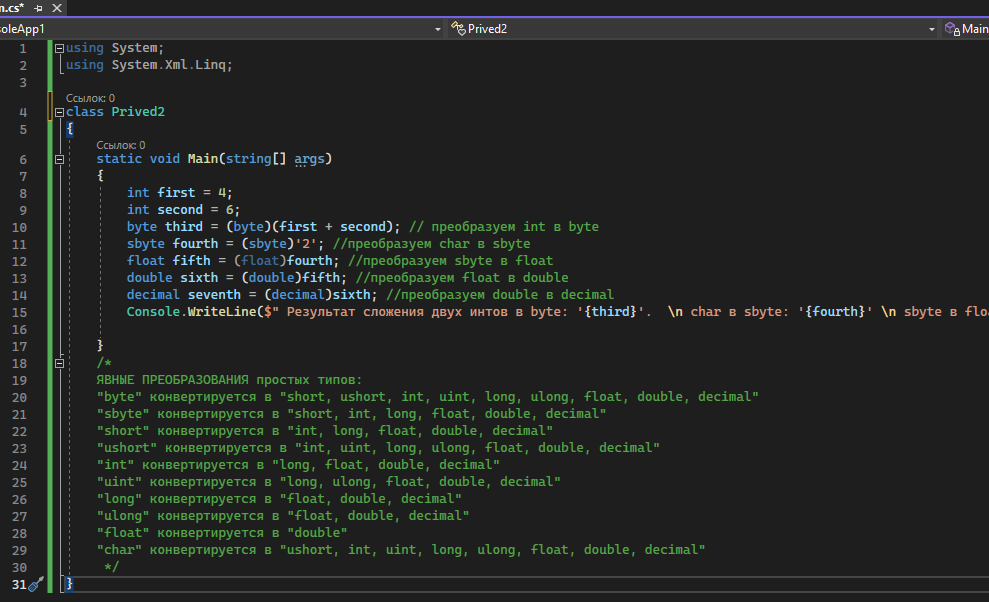


Результатом, конечно, выведется «Тип» танка:

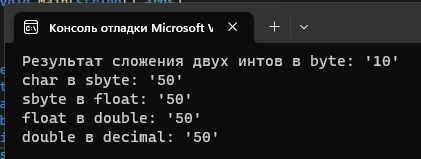


2-А) Явное преобразование простых типов:

Скриншот кода приведен ниже

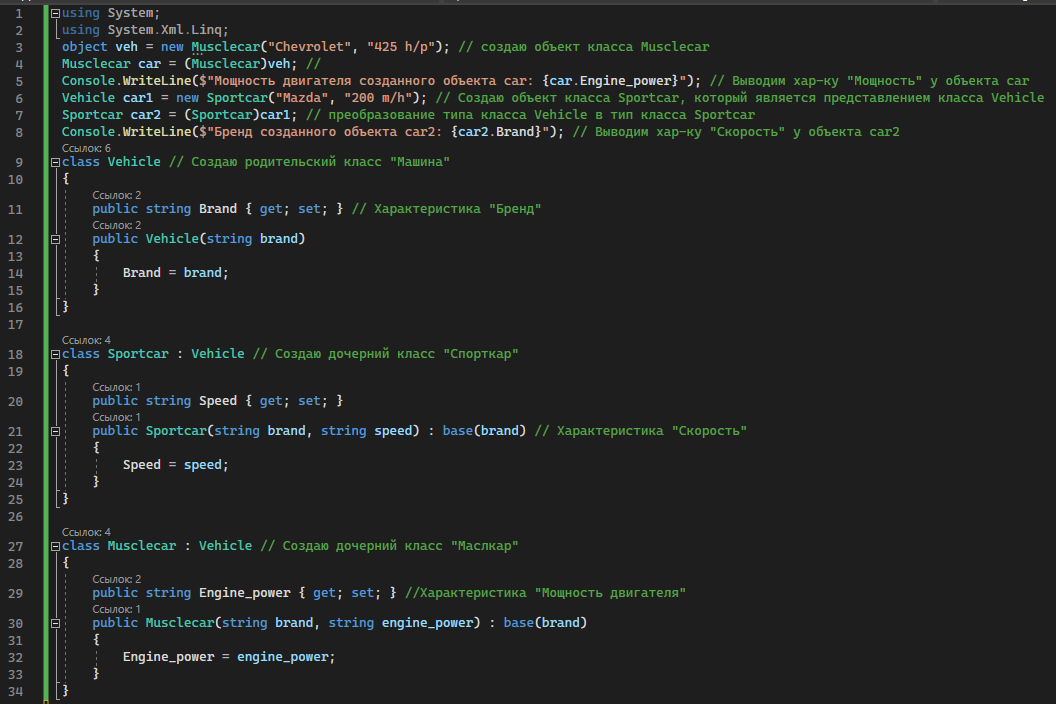


Результат:

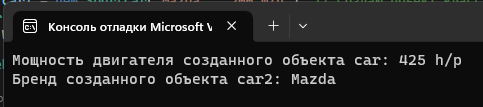


2-Б) Явное преобразование ссылочных типов:

Скриншот кода и интерфейса приведены ниже

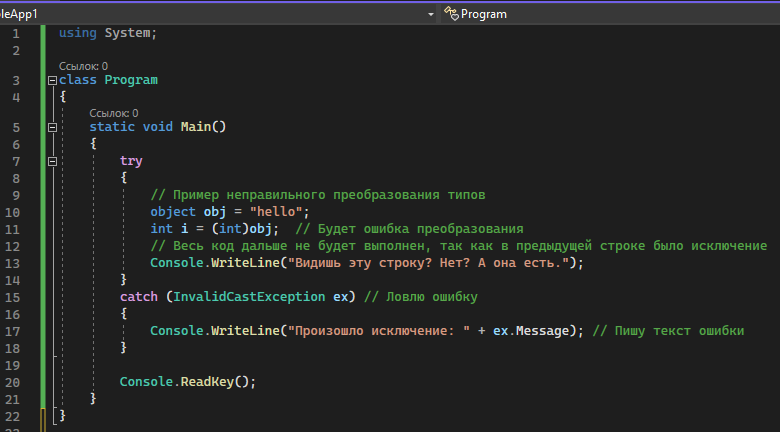


Результат:

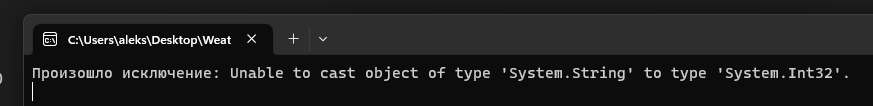


3) Вызвать и обработать исключение преобразования типов;

Скриншот кода приведен ниже

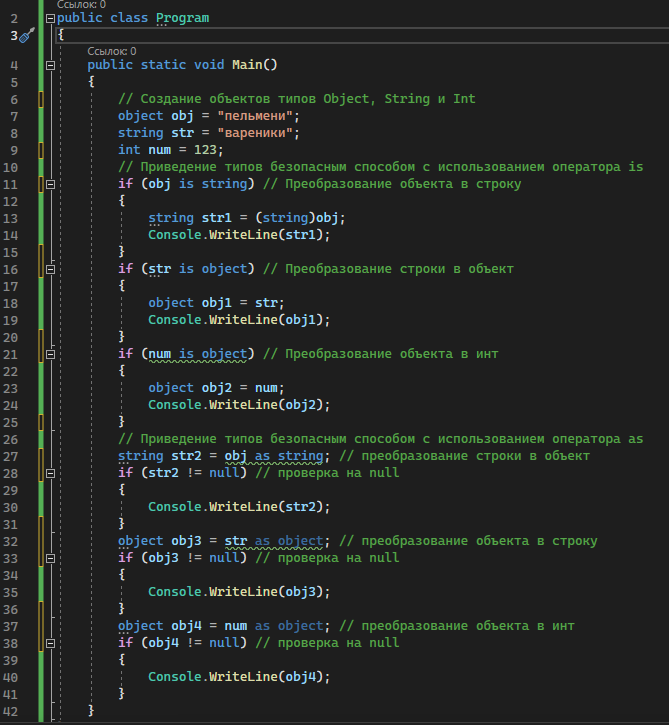


В результате будет выведен текст ошибки:

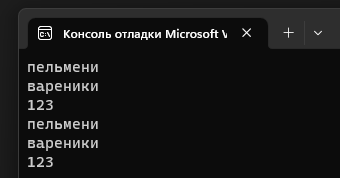


4) Безопасное приведение ссылочных типов с помощью операторов as и is

Скриншот кода приведен ниже

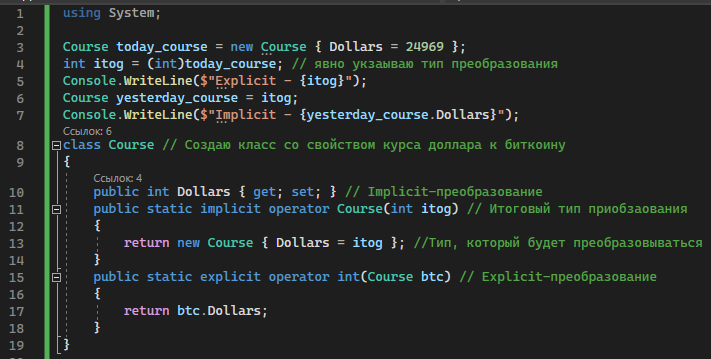


Как мы видим по результату, преобразования прошли успешно:

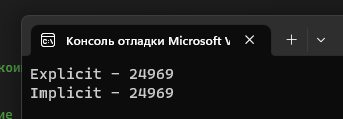


5) Пользовательское преобразование типов Implicit, Explicit

Скриншот кода приведен ниже

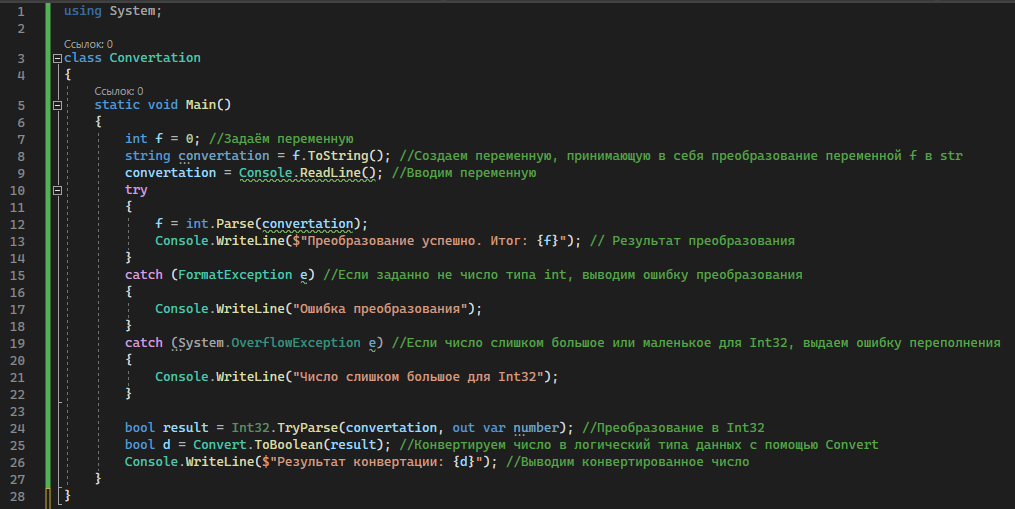


Результат:



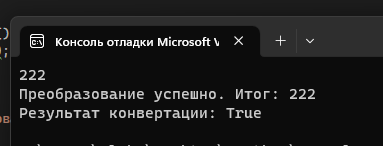
6) Преобразование с помощью класса Convert и преобразование строки в число с помощью методов Parse, TryParse класса System.Int32.

Скриншот кода приведен ниже

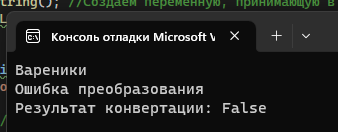


Результаты:

а) Если ввести число, то получим:

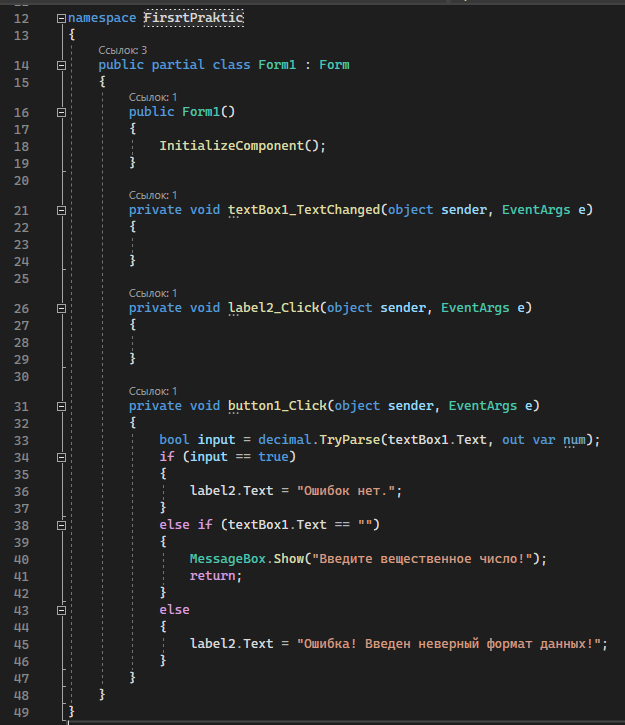


б) Соответственно, если ввести что-то другое, то будет ошибка преобразования:



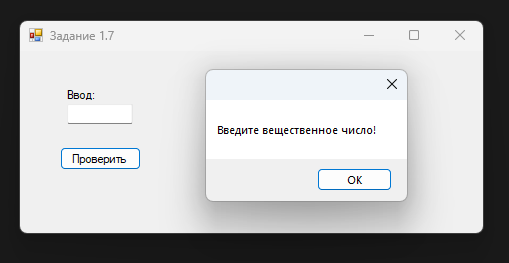
7) Написать программу, позволяющую ввод в текстовое поле TextBox только символов, задающих правильный формат вещественного числа со знаком.

Скриншот кода и интерфейса приведены ниже:

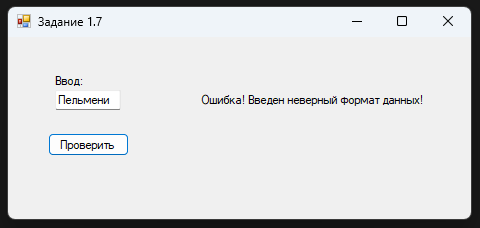


Итог:

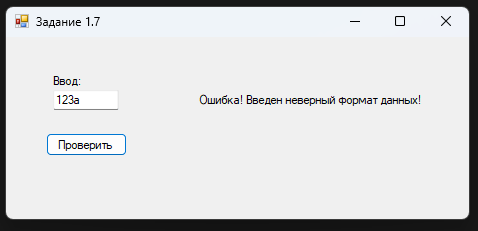
а) Если оставить поле пустым, то программа выдаст ошибку:



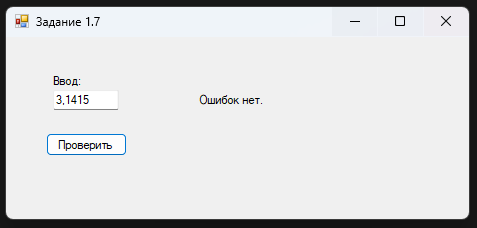
б) Если написать в поле любые символы, кроме чисел и запятой, то программа выведет, что формат введенных данных не верный.



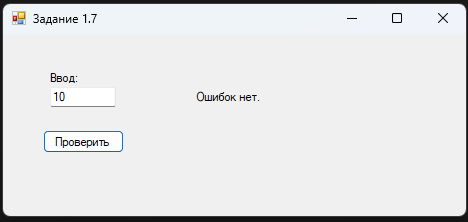
или



в) Если написать только символы, задающих правильный формат вещественного числа со знаком (или без), то программа выведет:



или



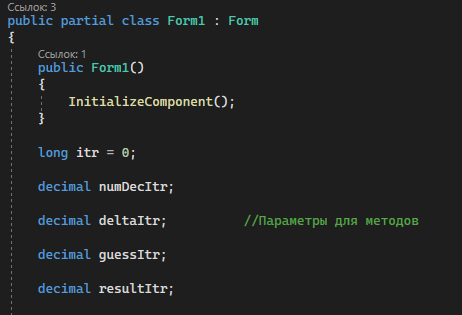
5)

**Задание 2**

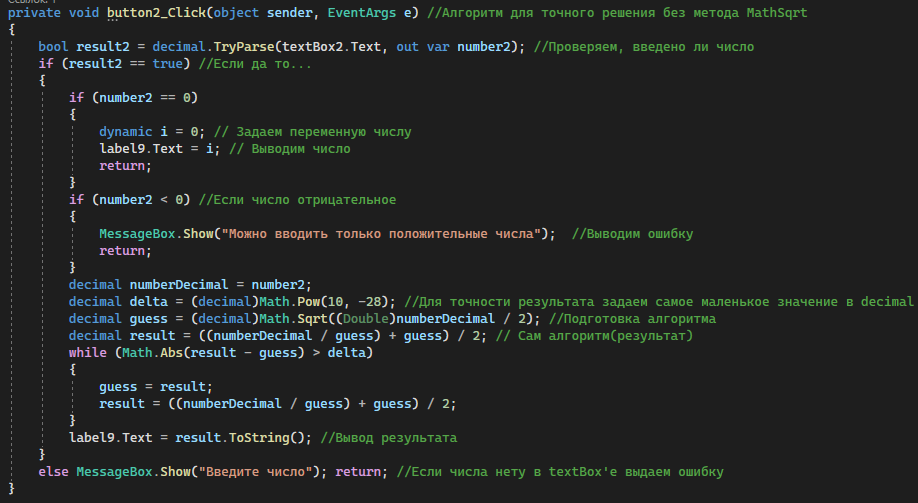
**Вычислить квадратный корень с контролем точности**

Скриншот кода и интерфейса приведены ниже:

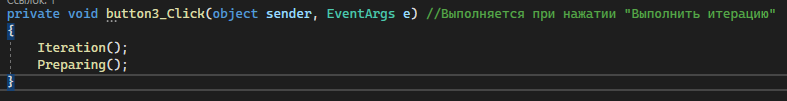
А) Параметры для методов:

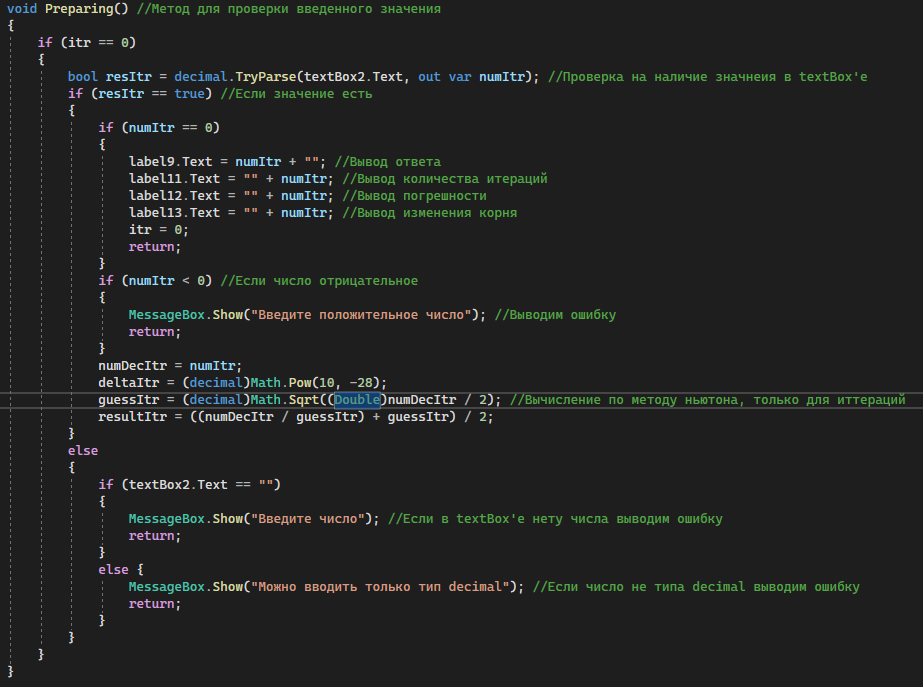


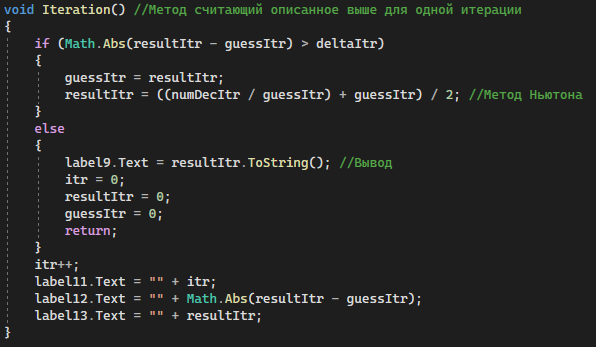
Б) Алгоритм для точного решения без метода MathSqrt()



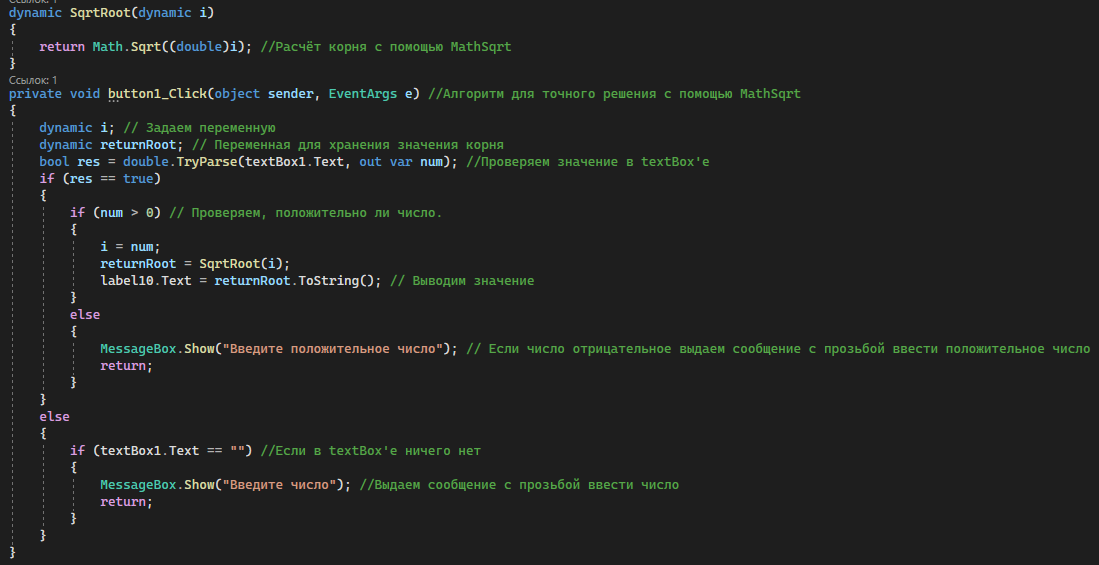
В) Алгоритм выполнения расчёта поитерационно:





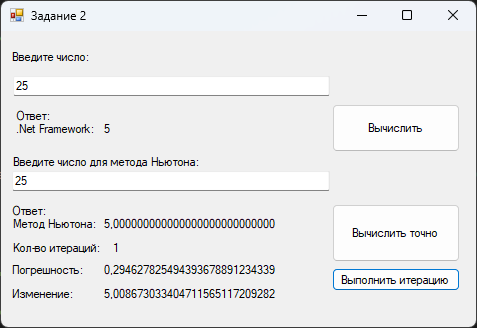


Г) Алгоритм выполнения задачи с помощью функция MathSqrt():

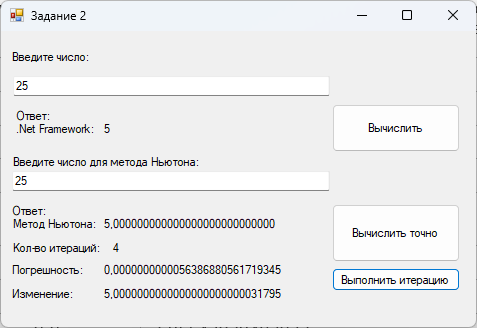


Введем в программу тестовые значения из методических указаний:

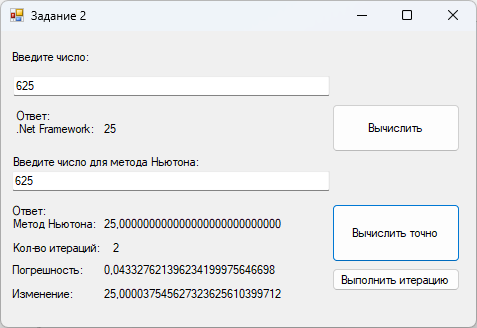
1) Значение 25, одна итерация:



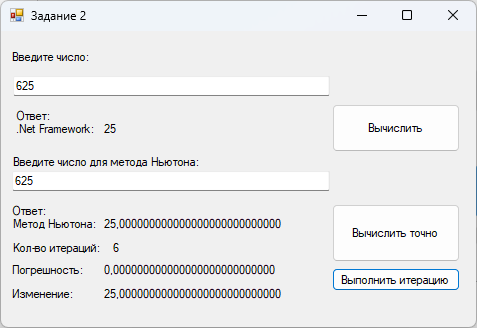
1-а) Значение 25, четыре итерации:



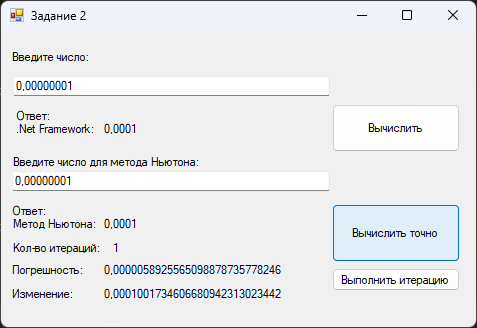
2) Значение 625, две итерации



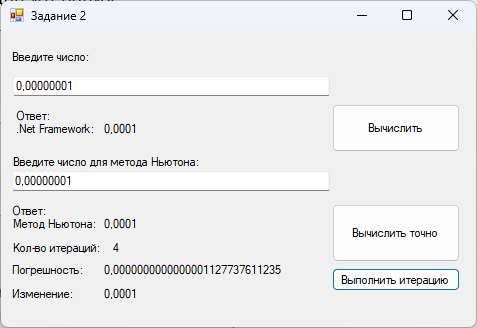
2-а) Значение 625, шесть итераций:



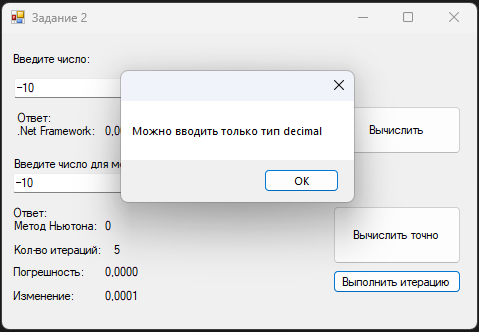
3) Значение 0.00000001, одна итерация:



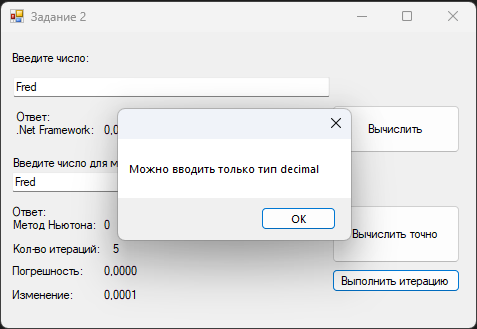
3-а) Значение 0.00000001, четыре итерации:



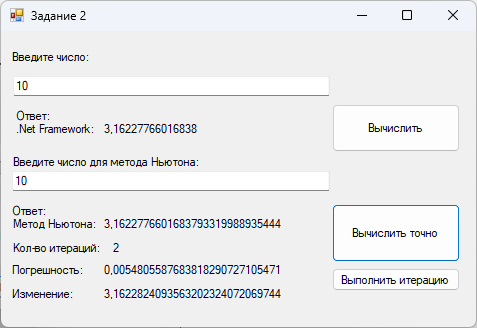
4) Значение -10. Примечание: при вводе любого значения, кроме положительного числа, выведется эта ошибка:



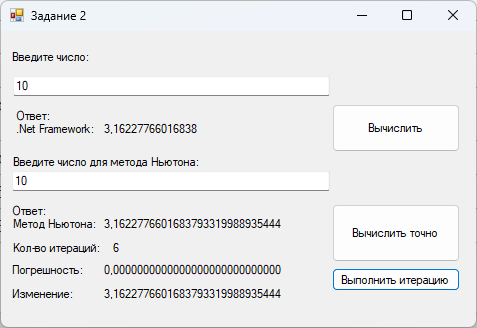
5) Значение Fred:



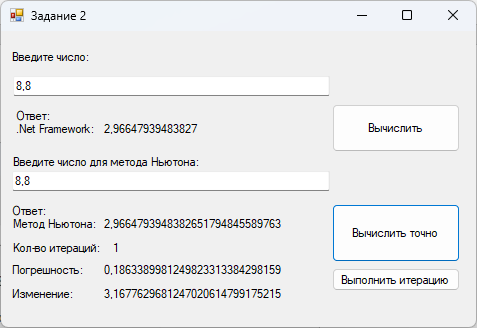
6) Значение 10, две итерации:



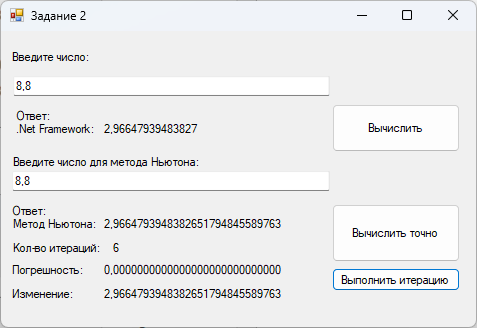
6-а) Значение 10, шесть итераций:



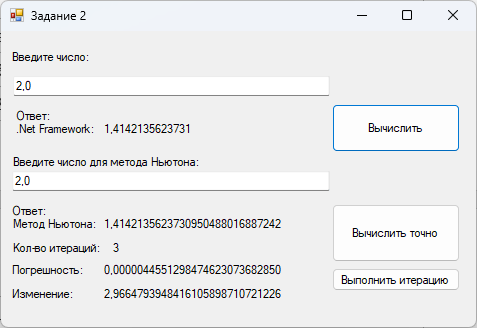
7) Значение 8.8, одна итерация:



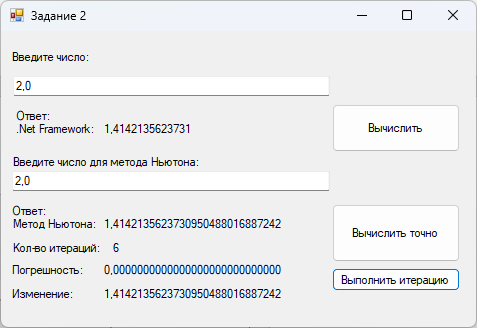
7-а) Значение 8.8, шесть итераций:



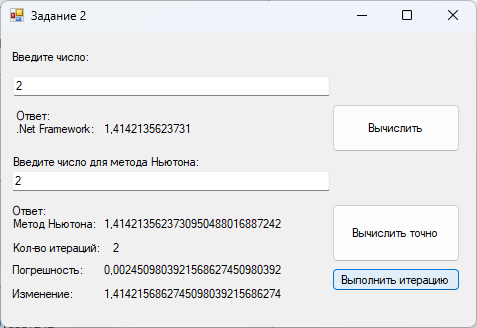
8) Значение 2.0, три итерации:



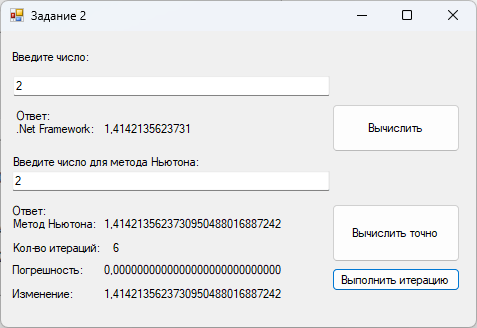
8-а) Значение 2.0, шесть итераций;



9) Значение 2, две итерации:



9-а) Значение 2, шесть итераций:

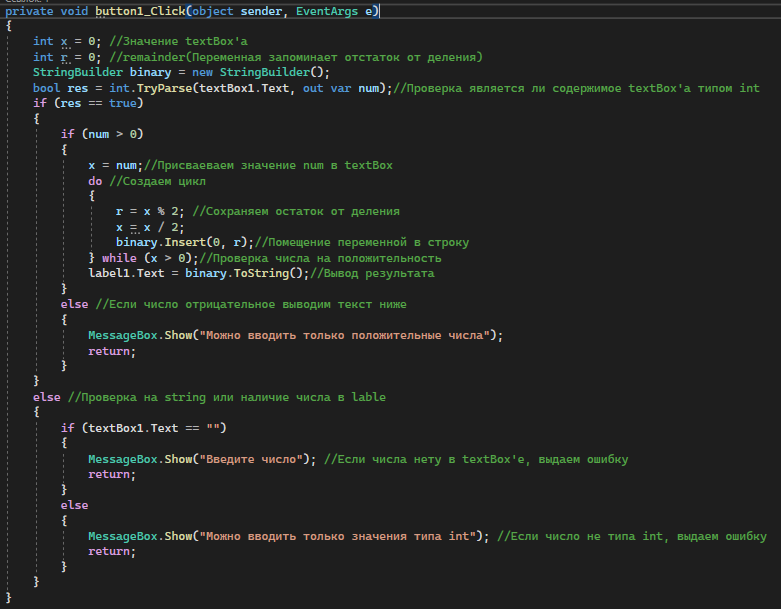


Примечание: максимальное количество итераций с любым числом будет шесть.

**Задание 3**

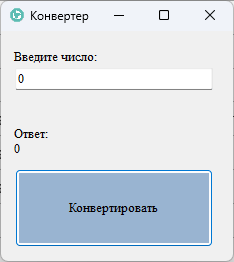
Конвертация целочисленных данных в бинарные.

Скриншот кода и интерфейса приведены ниже:

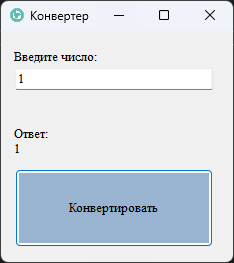


Введем в программу тестовые значения из методических указаний:

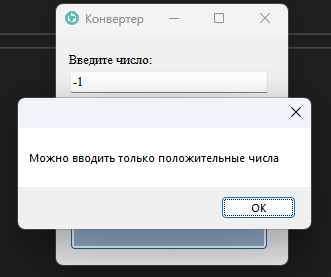
А) Значение 0:



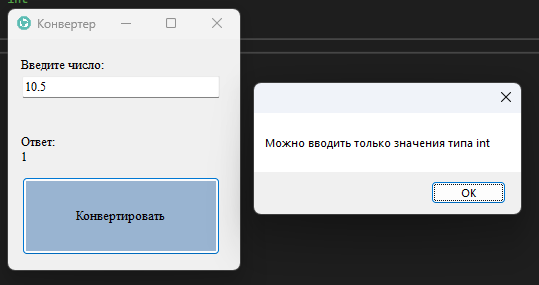
Б) Значение 1:



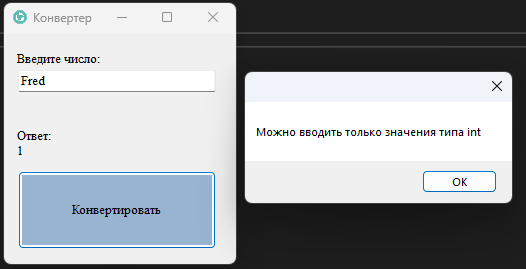
В) Значение -1:



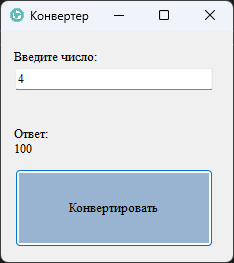
Г) Значение 10.5:



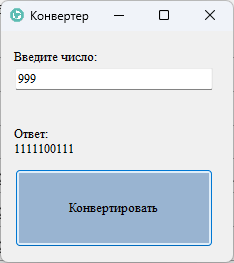
Д) Значение Fred:



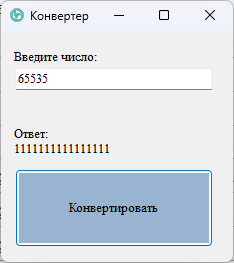
Е) Значение 4:



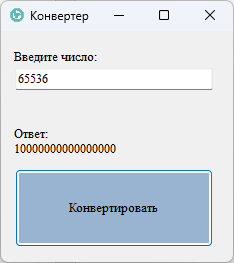
Ж) Значение 999:



З) Значение 65535:



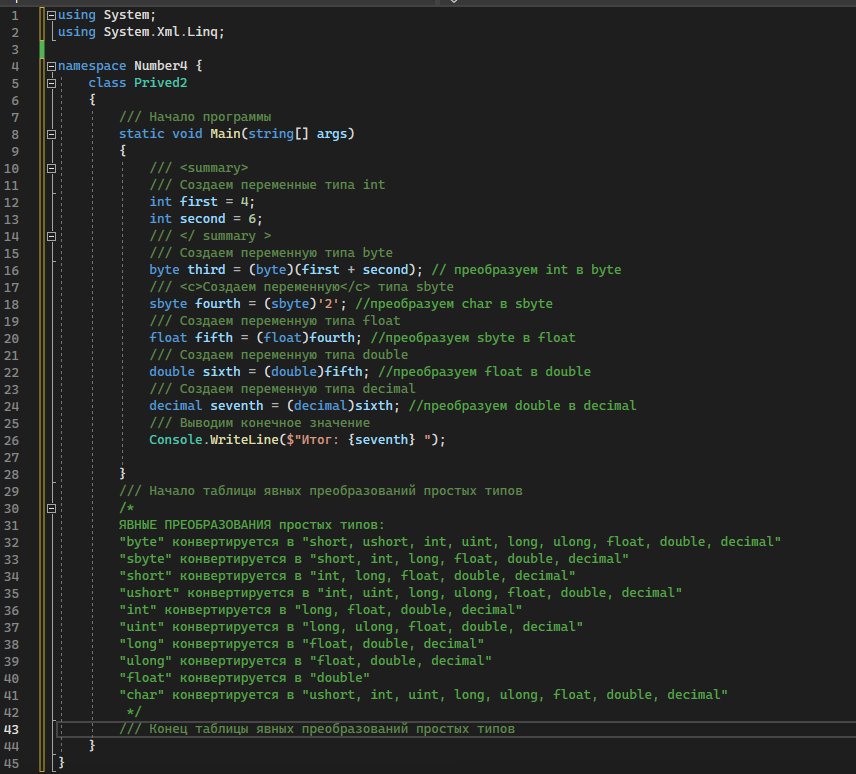
И) Значение 65536:



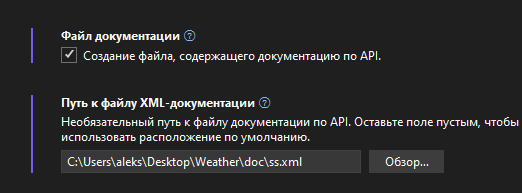
**Задание 4**

**Создание документации для приложения**

А) Добавить к коду XML-комментариев:



Б) Настроить генерацию XML файла и указать путь:



В) Получить XML файл (только нужно открыть не через Блокнот и настроить кодировку):

