Министерство образования и науки Российской Федерации

ФГАОУ ВО «КФУ им. В.И. Вернадского»

физико-технический институт (структурное подразделение)

|  |
| --- |
|  |

Кафедра компьютерной инженерии и моделирования

Пугач Давид Сергеевич

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

Направление подготовки:

09.03.04 "Программная инженерия"

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:**   
**IDE** -интегрированная среда разработки IDE (Integrated Development Envirionment) Visual Studio,   
**CLR** -общеязыковая исполнительная среда CLR (Common Langvidge 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 - библиотека классов,   
**CLS** - общеязыковые спецификации CLS, решение (solution), проект (project),   
**namespace** - пространство имен , способ организации системы типов в единую группу. Концепция пространства имён обеспечивает эффективную организацию и навигацию по этой библиотеке. Вне зависимости от языка программирования доступ к определённым классам обеспечивается за счёт их группировки в рамках общих пространств имён, сборка (assembly,  
**IL -**  MSIL (Microsoft Intermediate Language, IL), управляемый код, двухэтапная компиляция,  
**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.

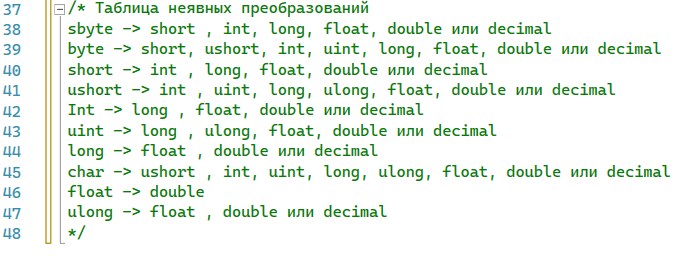
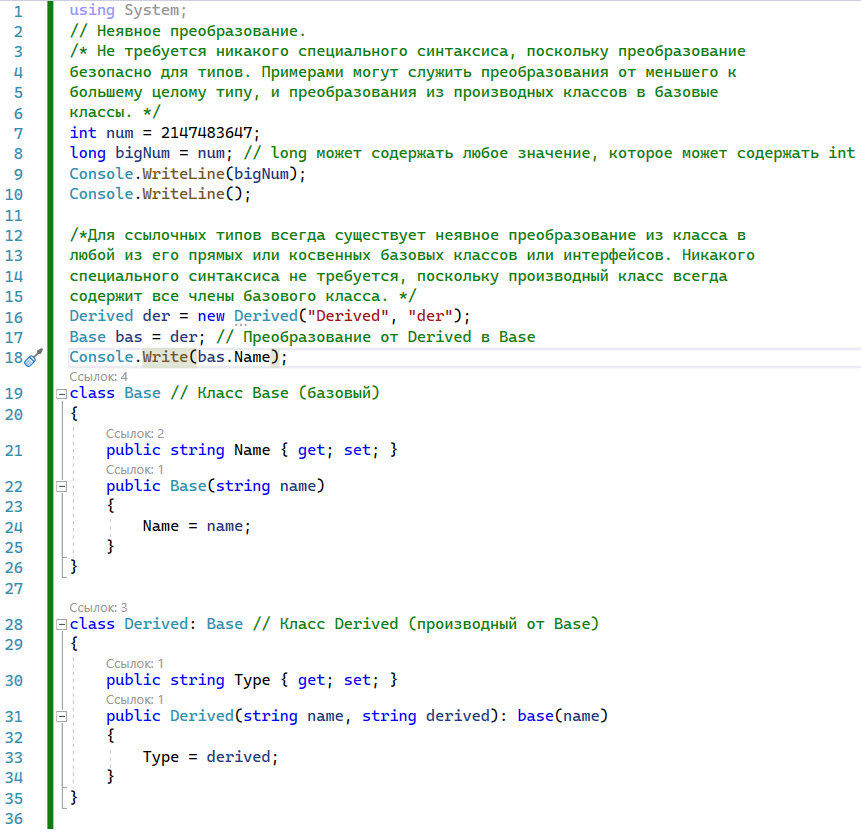
Исключение **InvalidCastException** возникает, когда преобразование экземпляра одного типа в другой тип не поддерживается. Например, при попытке преобразовать Char значение в DateTime значение возникает InvalidCastException исключение.

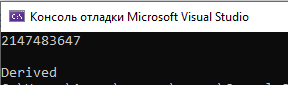
**Перед выполнением лабораторной работы изучена следующая литература:**

1. Презентации лектора курса: «Основы Net Framework» и «Использование конструкций языка C#».
2. Прослушана видеолекция «Основы Net Framework».
3. Прочитаны 1-2 лекции «Основы Net Framework» и «Использование конструкций языка C#».
4. Получены начальные сведения о спецификации языка C#.
5. Изучен раздел C# главы 1-4 сайта Metanit.com.

**Выполнены 4 задания, описанных в методических указания к выполнению лабораторных работ.   
Задание 1: Приведение и преобразование типов**

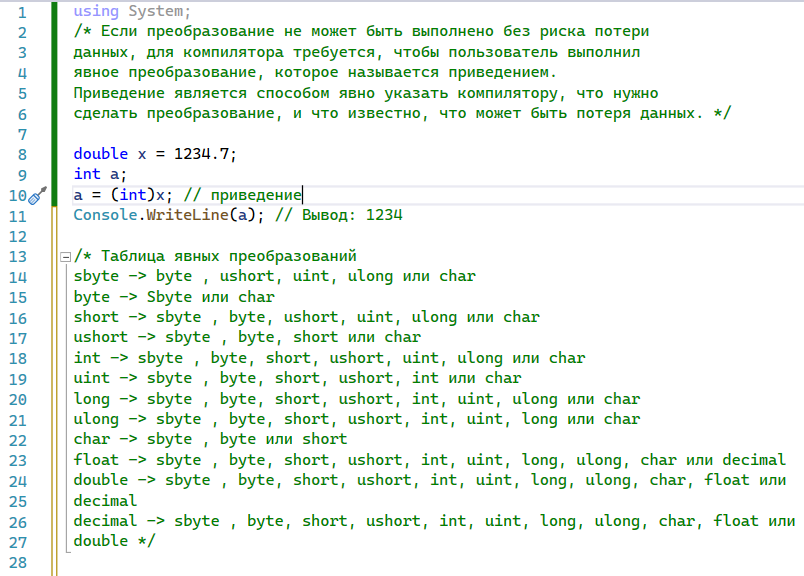
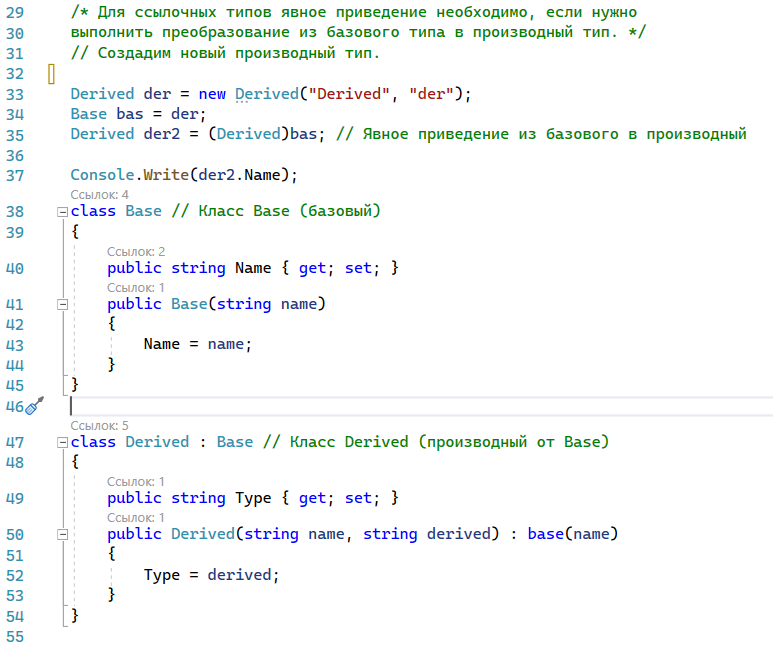
1) Неявное преобразование простых и ссылочных типов, в виде комментариев внести в программу таблицу неявных преобразований;





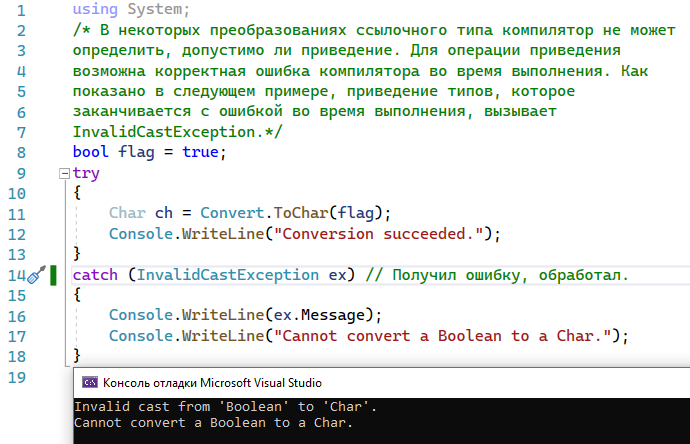
Использовал презентацию «Использование конструкций языка C#» и сайт metanit (Глава 2 -> Преобразования базовых типов данных).

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

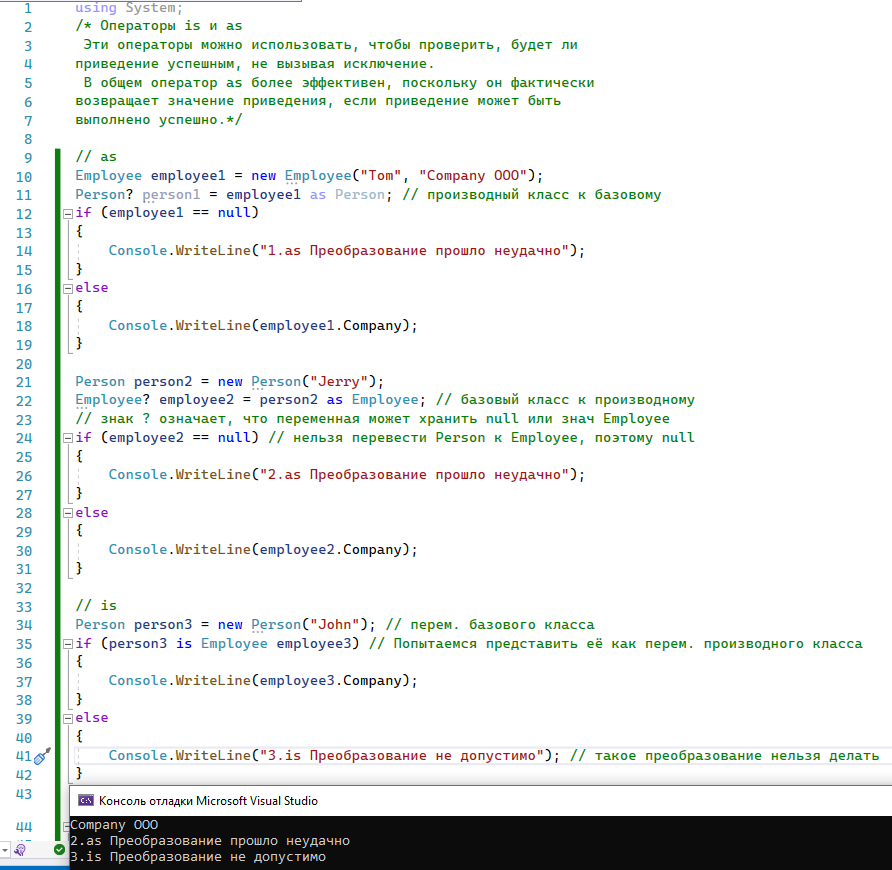
Использовал презентацию «Использование конструкций языка C#» и сайт metanit (Глава 2 -> Преобразования базовых типов данных).

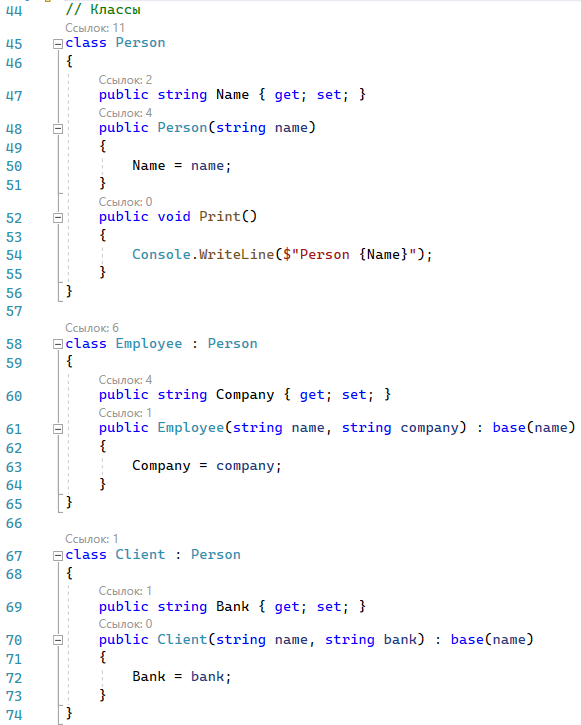
3) Вызвать и обработать исключение преобразования типов;



Использовал презентацию «Использование конструкций языка C#» и сайт Microsoft ( [Learn](https://learn.microsoft.com/ru-ru/?view=net-7.0) / [.NET](https://learn.microsoft.com/ru-ru/dotnet/?view=net-7.0) / [Браузер API .NET](https://learn.microsoft.com/ru-ru/dotnet/api/?view=net-7.0) / [Система](https://learn.microsoft.com/ru-ru/dotnet/api/system?view=net-7.0) / InvalidCastException Класс).

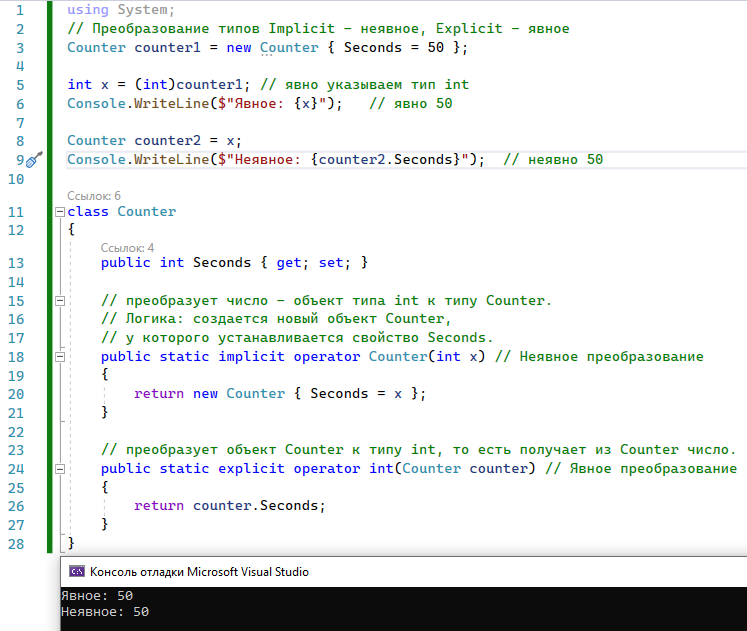
4) Безопасное приведение ссылочных типов с помощью операторов as и is;





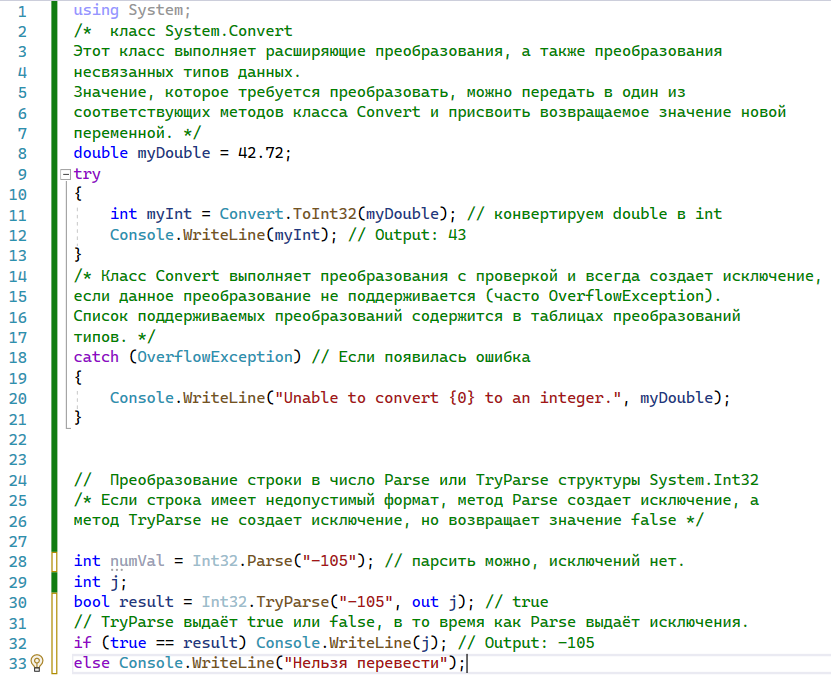
Использовал презентацию «Использование конструкций языка C#» и сайт metanit (Глава 4 -> Преобразования типов).

5) Пользовательское преобразование типов Implicit, Explicit;



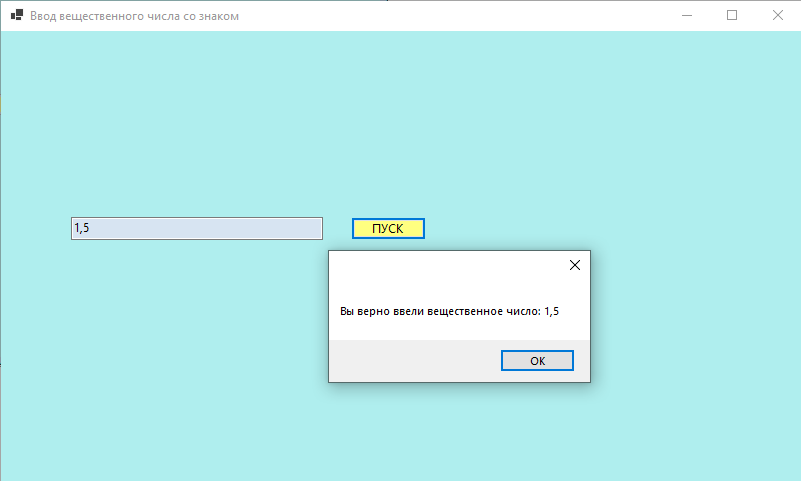
Использовал сайт metanit (Глава 8 -> Перегрузка операций преобразования типов).

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

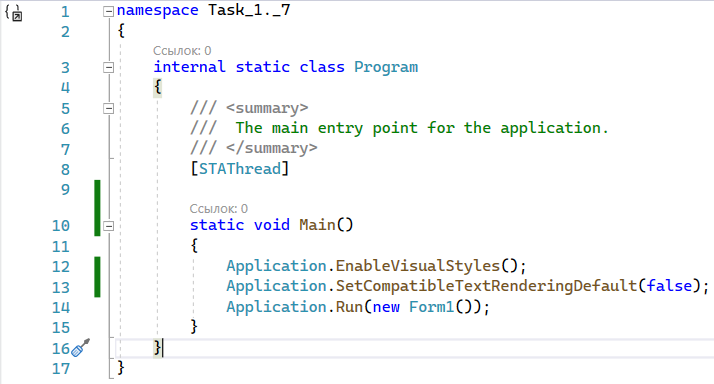


Использовал презентацию «Использование конструкций языка C#».

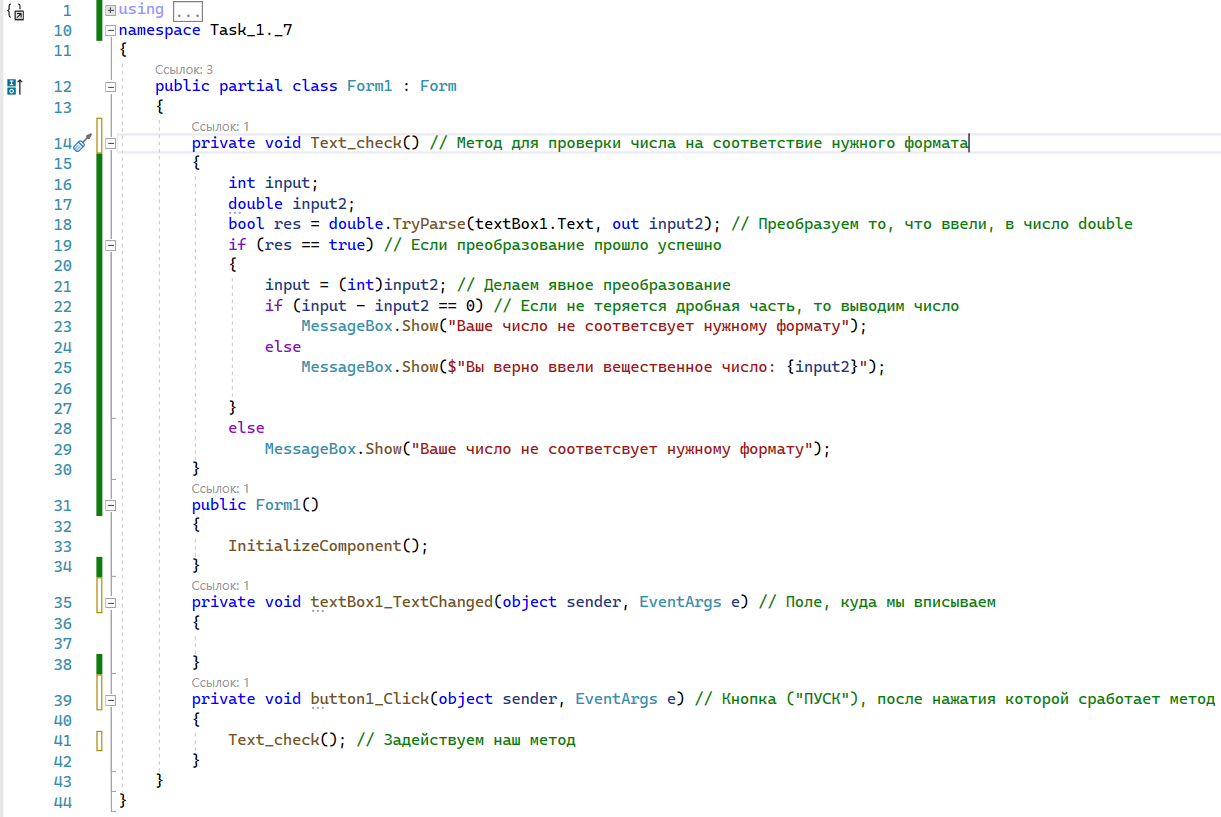
7) Написать программу, позволяющую ввод в текстовое поле TextBox только символов, задающих правильный формат вещественного числа со знаком.



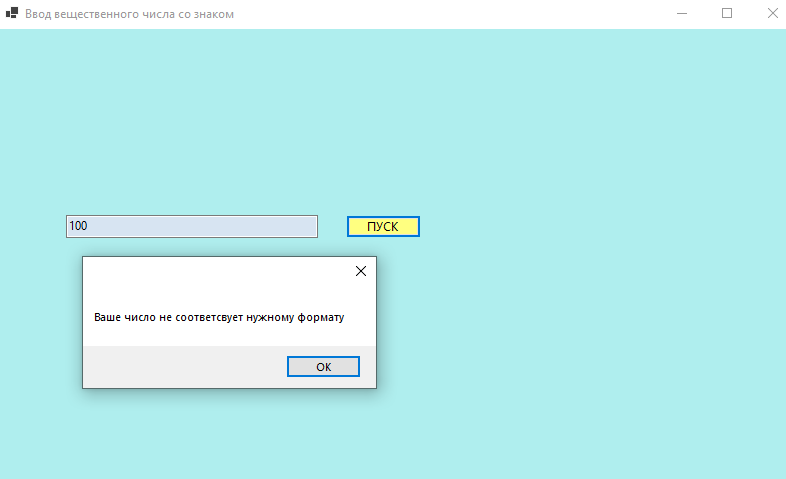
Форма с TextBox и Button с результатом работы программы.



Файл Program.cs



Файл Form1.cs

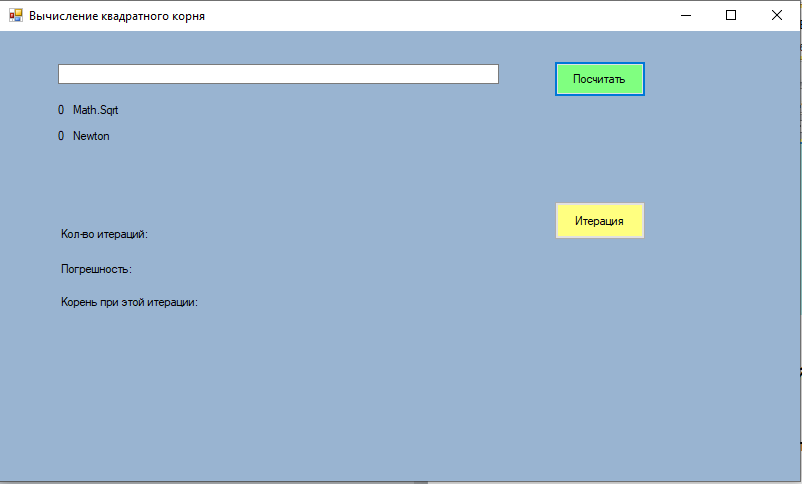


Пример ввода целого числа.

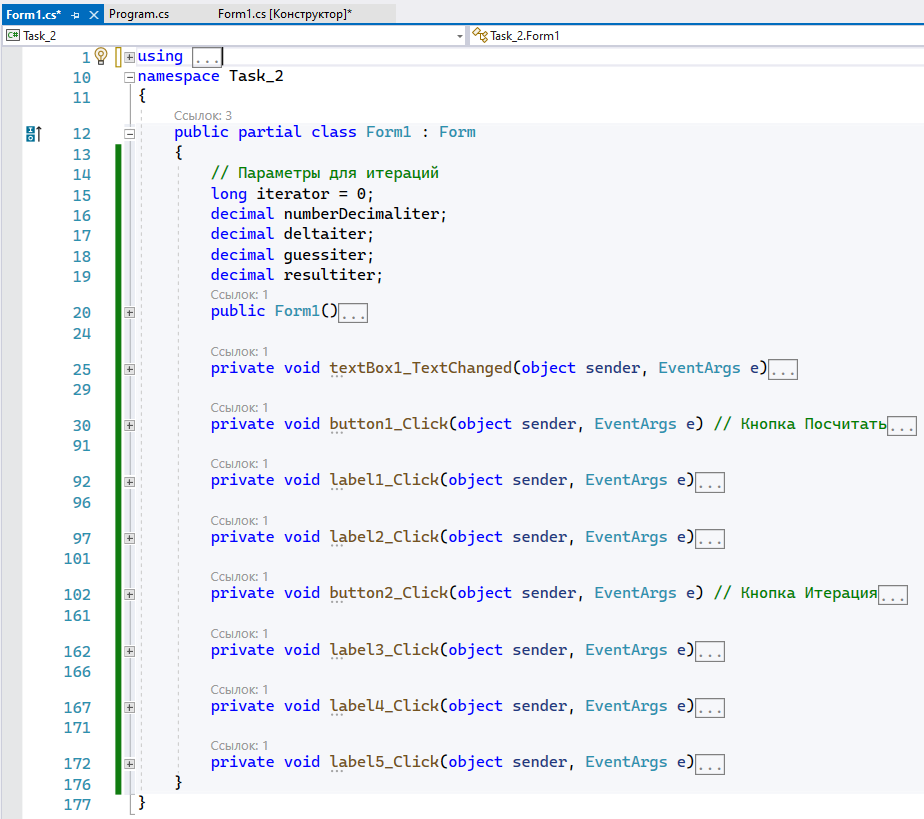


Пример ввода строки.

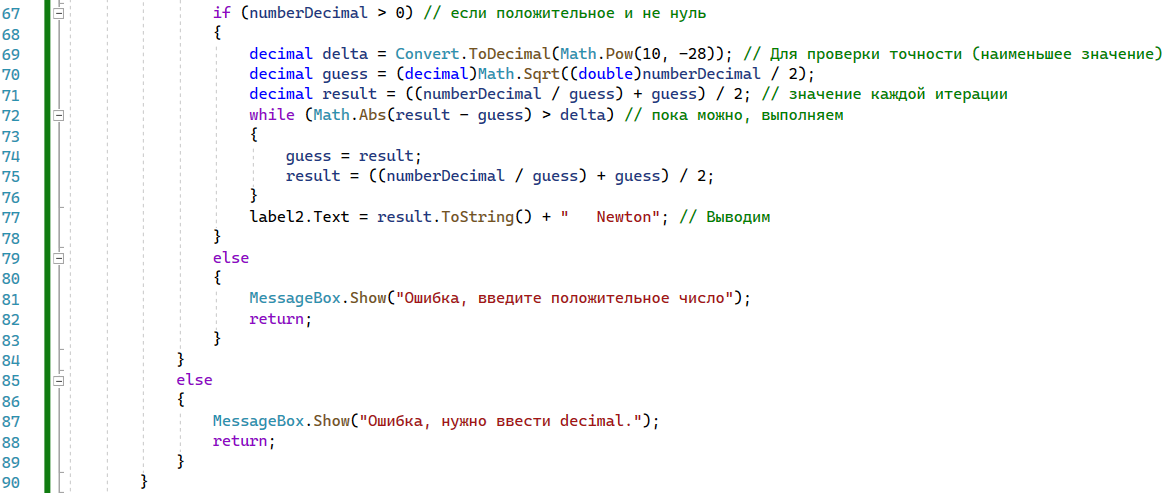
**Задание 2: Вычислить квадратный корень с контролем точности**



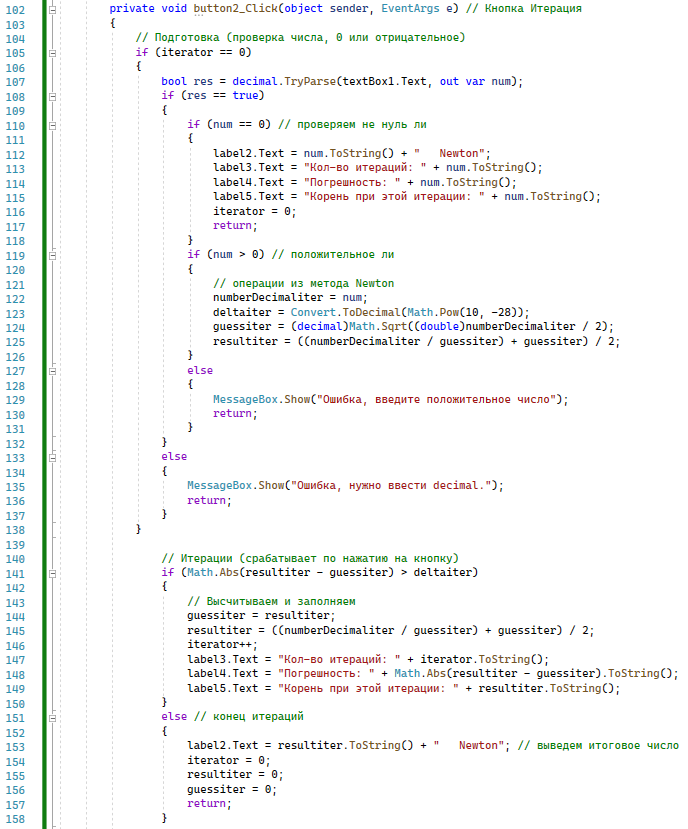
Интерфейс программы



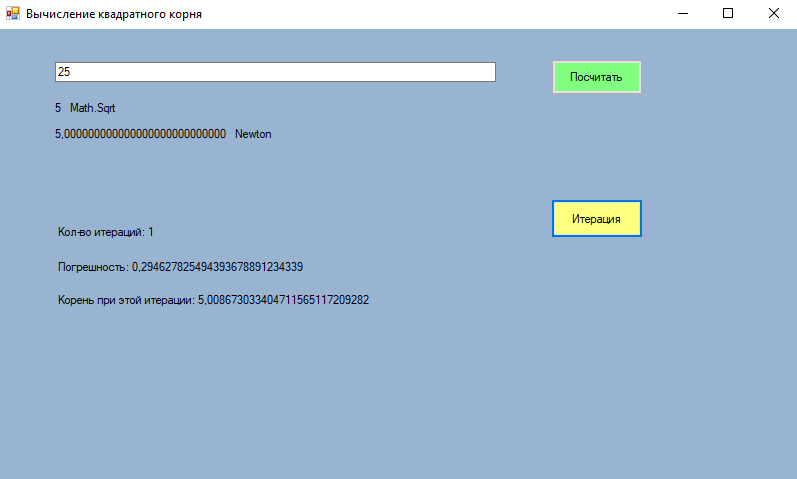
Файл Form1.cs

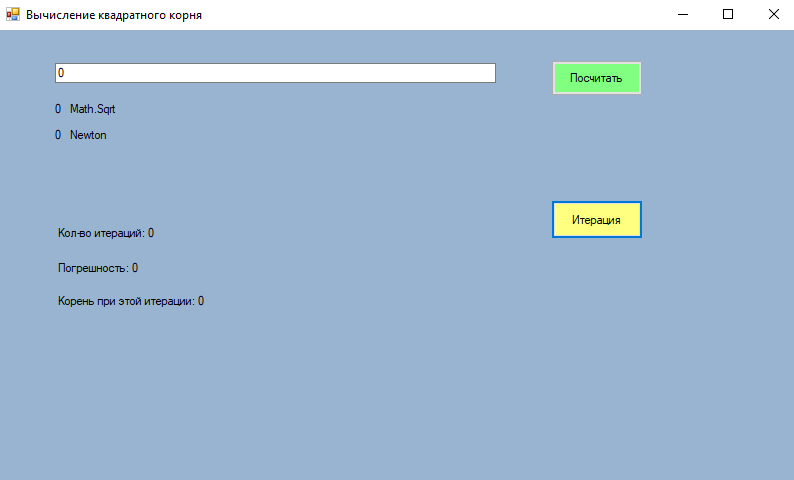
 

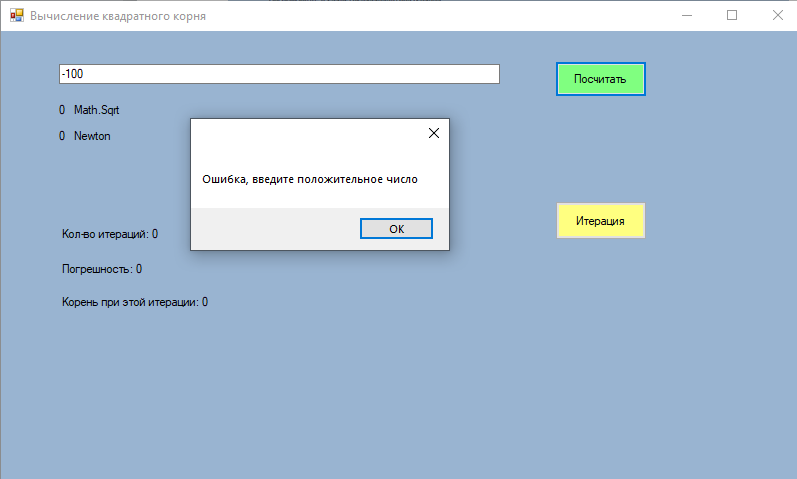
Код, который срабатывает при нажатии кнопки «Посчитать».



Код, который срабатывает при нажатии на кнопку «Итерация».





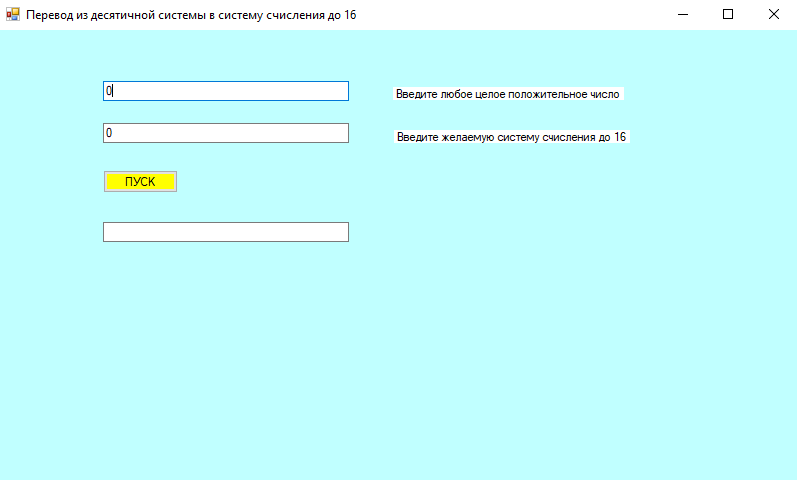


Примеры работы программы.

**Задание 3: Конвертация целочисленных данных в бинарные.**

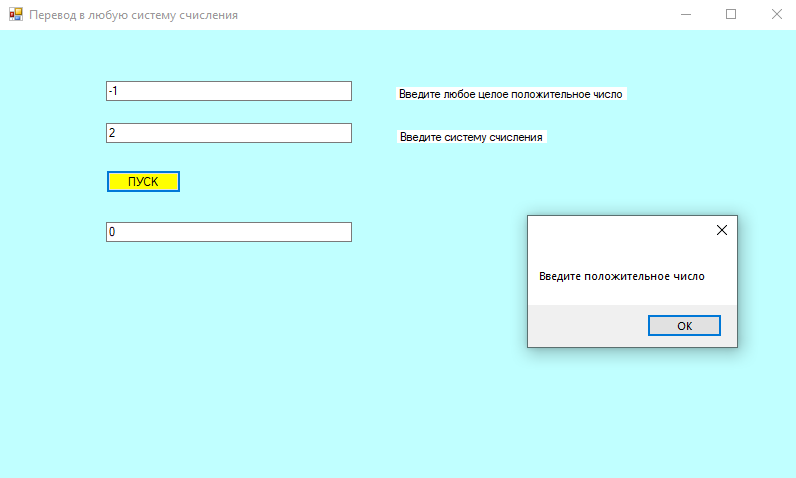
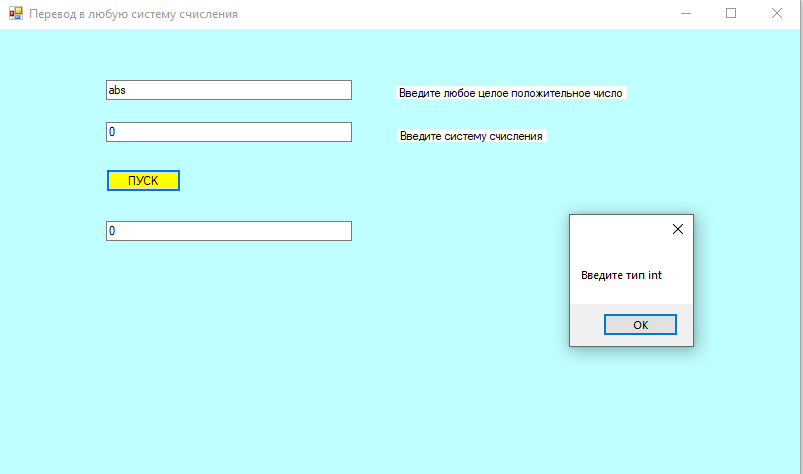
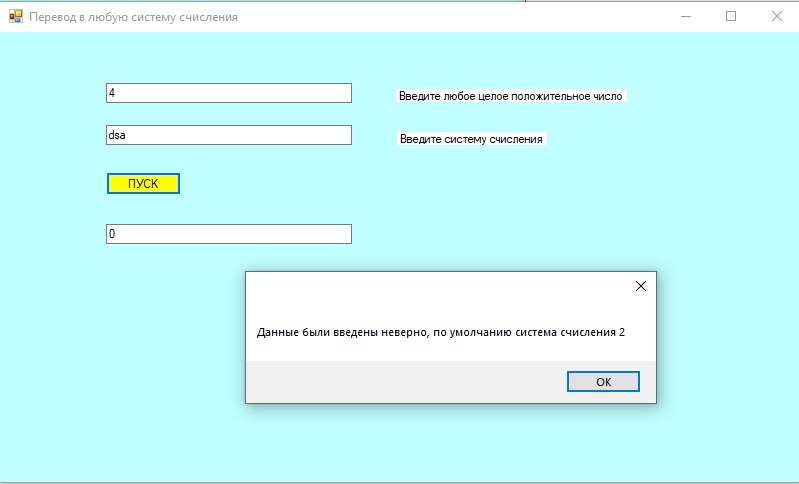
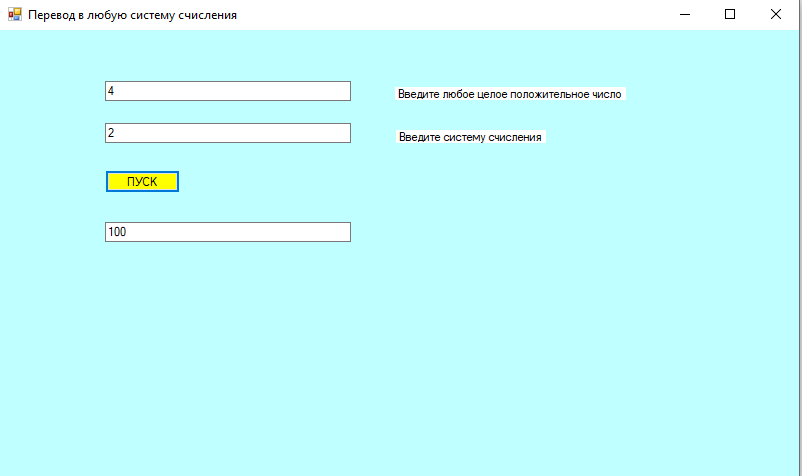
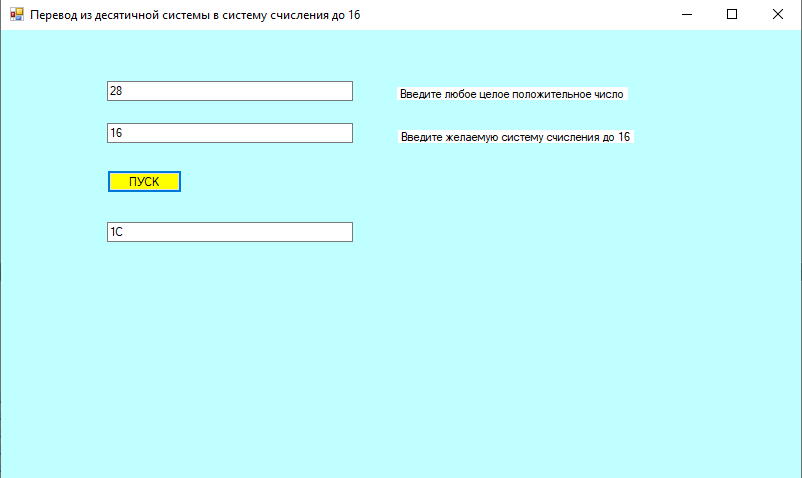
 

Файл Form1.cs

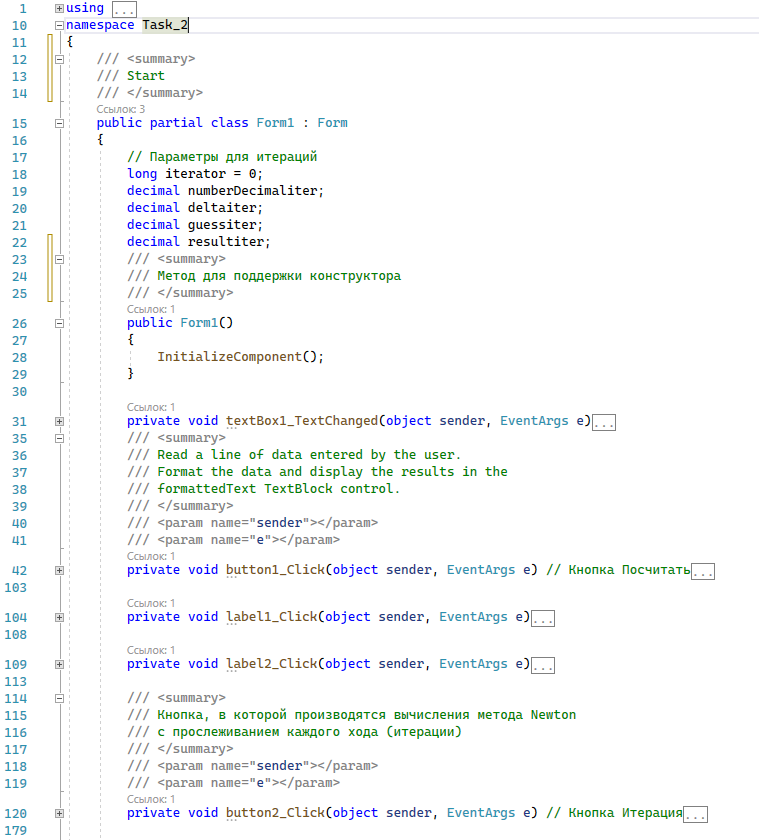


Окно формы.

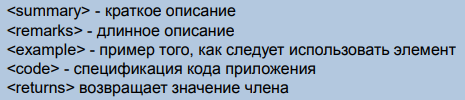
Примеры работы программы:

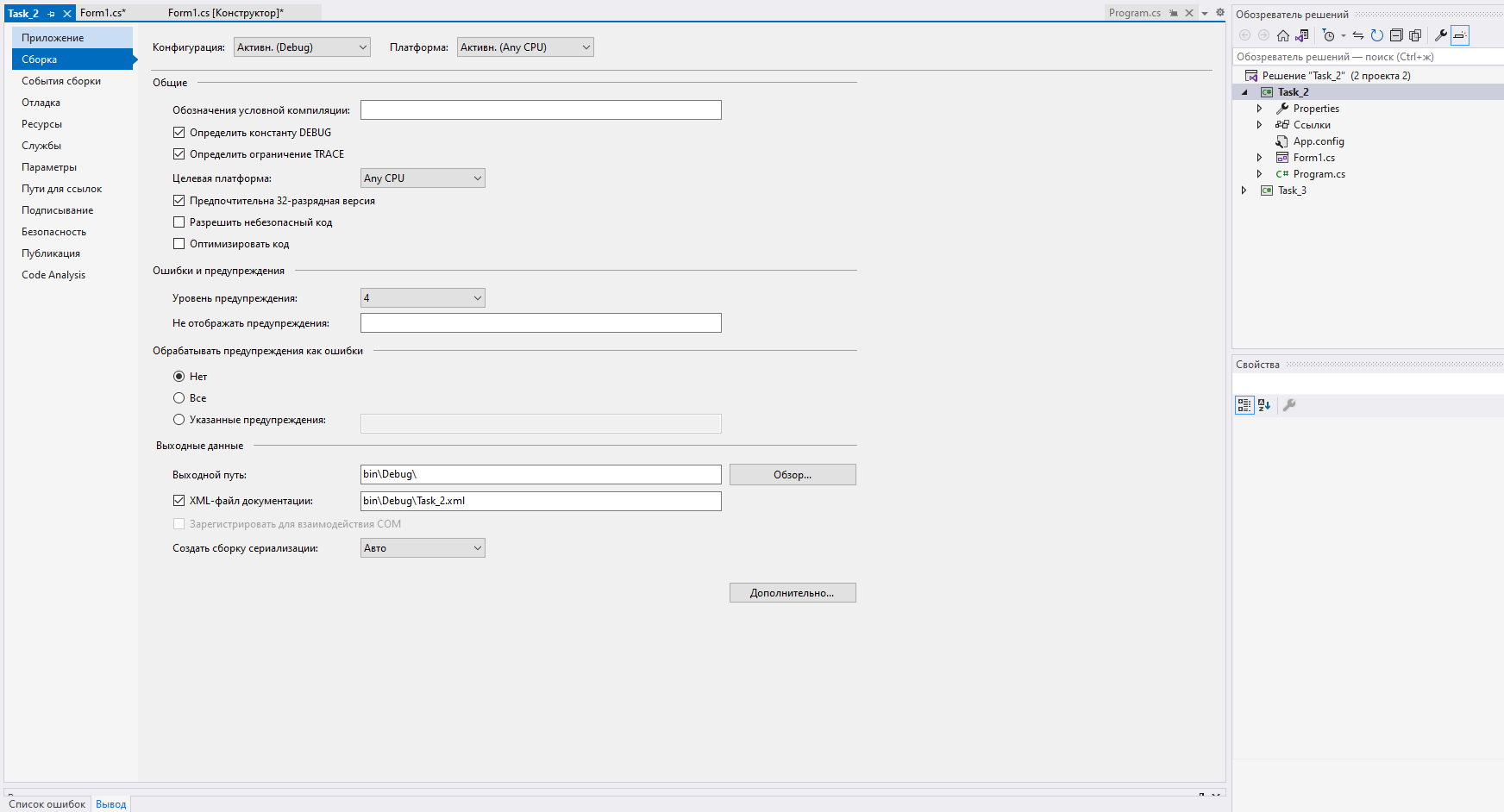
    

**Задание 4: Создание документации для приложения**



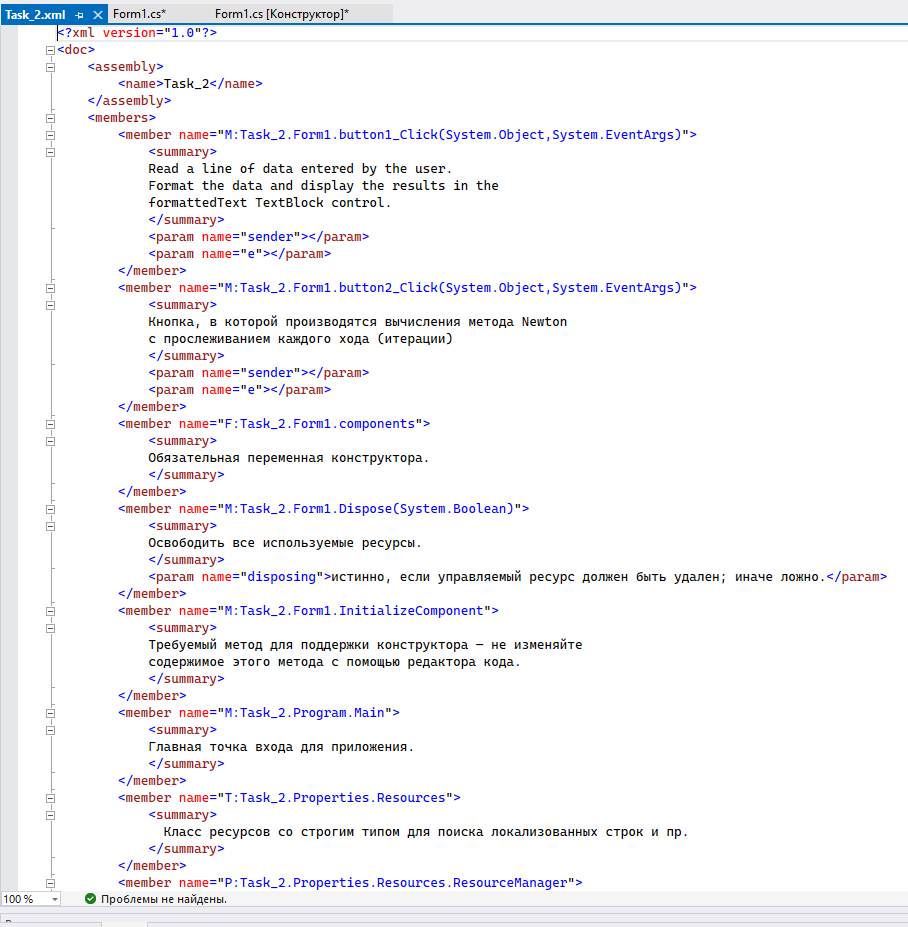
Комментарии XML, теги:





Заходим в свойство проекта и ставим галочку XML-файл документации, после чего нужно «Собрать» проект.

Текст файла документации приведен ниже



XML файл.

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

**Ссылка на репозиторий с программами**

https://github.com/Bananaws/Object-oriented-programming

**Вопросы, заданные преподавателем:**

* 1. Что такое IDE и какие её преимущества?

IDE - интегрированная среда разработки IDE (Integrated Development Envirionment) Visual Studio. Оно повышает производительность разработчиков, объединяя такие возможности, как редактирование, создание, тестирование и упаковка программного обеспечения в простом для использования приложении.

* 1. Основные различия между WPF и Windows Forms.

Одним из самых важных различий между WinForms и WPF является тот факт, что в WinForms интерфейс пользователя это всего лишь графический слой, использующий стандартные элементы управления Windows (например TextBox), а WPF-интерфейс, в свою очередь, построен "с чистого листа", не опираясь в большинстве случаев на стандартные элементы.

* 1. Зачем нужен XML файл документации?

Расширяемый язык разметки (XML) позволяет определять и хранить данные совместно используемым способом. XML поддерживает обмен информацией между компьютерными системами, такими как веб-сайты, базы данных и сторонние приложения. XML используется везде, где требуется выделить логическое содержимое документа для обработки.

* 1. Может ли быть потеряна точность в неявных преобразованиях из int, uint, long, или ulong в float и из long или ulong в double.

Точность, но не величина может быть потеряна в преобразованиях из int, uint, long, или ulong в float и из long или ulong в double.

* 1. Как осуществить неявное преобразование в тип char.

Не поддерживается неявное преобразование в тип char.

* 1. Как осуществить неявные преобразования между типами с плавающей запятой и типом decimal?

Неявные преобразования между типами с плавающей запятой и типом decimal отсутствуют.

* 1. Всегда ли можно явно преобразовать int в sbyte, byte, short, ushort, uint или ulong?

Выражение типа int можно преобразовать в sbyte, byte, short, ushort, uint или ulong при условии, что значение выражения находится в диапазоне типа назначения. Иначе потеряем значение.

* 1. Зачем в работе используется класс StringBuilder?

Он нужен, чтобы сделать изменяемую строку символов.