# Практическое занятие № 14-15. Перегрузка операторов и методов. Использование делегатов

**1 Цель занятия**

Получить практические навыки создания перегруженных методов и использования делегатов

**2 Перечень оборудования и программного обеспечения**

Персональный компьютер

Microsoft Office (Word)

Microsoft Visual Studio

**3 Краткие теоретические сведения**

**3.1 Перегрузка операторов и методов**

***Полиморфизм*** — обеспечивает возможность задания различных реализаций некоторого единого по названию метода для классов различных уровней иерархий.

В C# допускается совместное использование одного и того же имени двумя или более методами одного и того же класса, при условии, что их параметры объявляются по-разному.

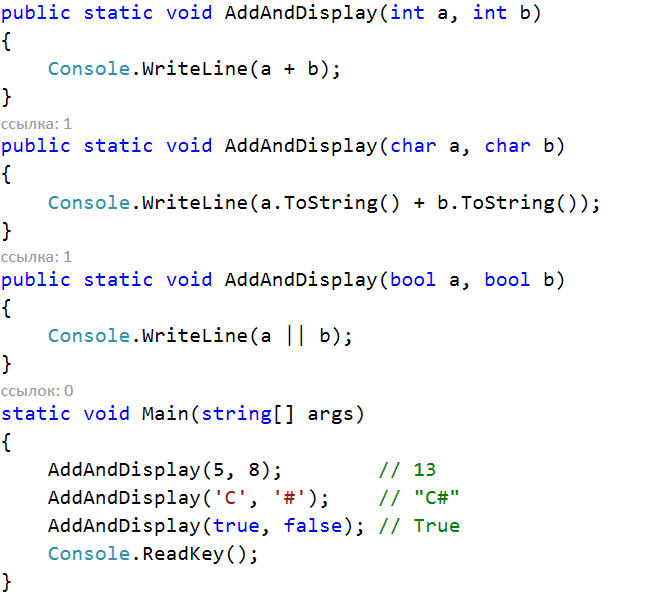
Существование в классе методов с одним и тем же именем называется ***перегрузкой***, а сами одноименные методы называются ***перегруженными***. Перегрузка методов полезна, когда требуется решать подобные задачи с разным набором аргументов. Перегрузка методов относится к одному из способов реализации полиморфизма в C#.

Перегрузка характерна и для знаков операций. В зависимости от типов аргументов, один и тот же знак может выполнять фактически разные операции. Классическим примером является знак операции сложения «+», который играет роль операции сложения не только для арифметических данных разных типов, но и выполняет конкатенацию строк.

Перегрузка требует уточнения семантики вызова метода. Когда встречается вызов неперегруженного метода, то имя метода в вызове однозначно определяет, тело какого метода должно выполняться в точке вызова. Когда же метод перегружен, то знания имени недостаточно - оно не уникально. Уникальной характеристикой перегруженных методов является их ***сигнатура***. Перегруженные методы, имея одинаковое имя, должны отличаться либо числом аргументов, либо их типами, либо ключевыми словами (отметим: с точки зрения сигнатуры, ключевые слова ref и out не отличаются). Уникальность сигнатуры позволяет вызвать требуемый перегруженный метод.

Перегрузка широко используется при построении библиотеки .NET FCL (стандартной библиотеки классов платформы .NET Framework). Например, класс Convert имеет 16 методов преобразований типов данных с разными именами, зависящими от целевого типа преобразования. Каждый из этих 16 методов перегружен, и в свою очередь, имеет 16 реализаций в зависимости от типа источника.

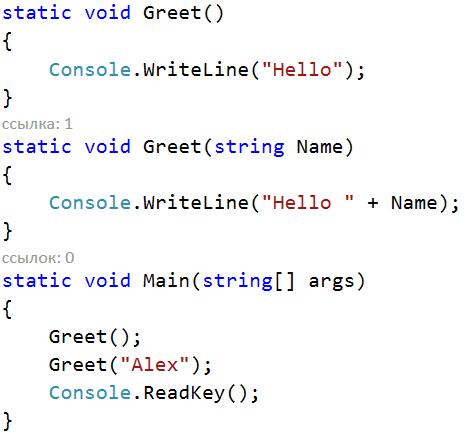
**Пример 1.** Метод AddAndDisplay(), прибавляющий аргументы друг к другу и выводящий результат:



При вызове метода, компилятор сам решает, какой вариант метода AddAndDisplay() необходимо вызвать.

Совершенно недостаточно, чтобы методы отличались только типами возвращаемых значений. Они могут также отличаться числом своих параметров. Когда вызывается перегружаемый метод, то выполняется тот его вариант, параметры которого соответствуют (по типу и числу) передаваемым аргументам.

**Пример 2:**



**3.2 Использование делегатов**

***Делегат*** представляет собой объект, который может ссылаться на метод. Следовательно, когда создается делегат, то в итоге получается объект, содержащий ссылку на метод. Иными словами, делегат позволяет вызывать метод (или список методов), на который он ссылается.

Объект делегата поддерживает три важных фрагмента информации:

* адрес метода, на котором он вызывается;
* аргументы (если есть) этого метода;
* возвращаемое значение (если есть) этого метода.

Как только делегат создан и снабжен необходимой информацией, он может динамически вызывать методы, на которые указывает, во время выполнения. Каждый делегат в .NET Framework автоматически снабжается способностью вызывать свои методы синхронно или асинхронно. Этот факт значительно упрощает задачи программирования, поскольку позволяет вызывать метод во вторичном потоке выполнения.

Тип делегата в C# объявляется с помощью ключевого слова delegate:

delegate <возвращаемый\_тип> <имя>([<параметры>]);

где <возвращаемый\_тип> обозначает тип значения, возвращаемого методами, которые будут вызываться делегатом;

<имя> — конкретное имя делегата;

<параметры> — параметры методов, вызываемых делегатом.

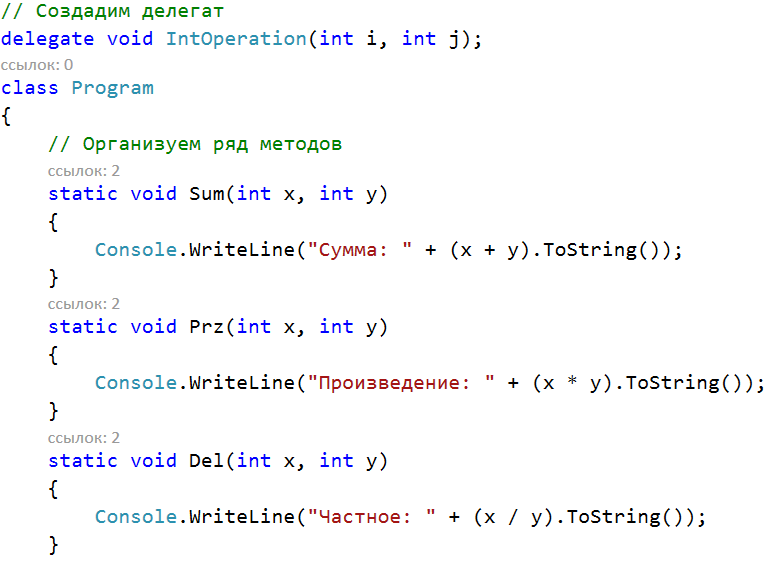
Как только будет создан экземпляр делегата, он может вызывать и ссылаться на методы, возвращаемый тип и параметры которых соответствуют указанным в объявлении делегата.

Делегат может служить для вызова любого метода с соответствующей сигнатурой и возвращаемым типом. Более того, вызываемый метод может быть методом экземпляра, связанным с отдельным объектом, или же статическим методом, связанным с конкретным классом. Значение имеет лишь одно: возвращаемый тип и сигнатура метода должны быть согласованы с теми, которые указаны в объявлении делегата.

Для делегатов определены свойства и методы, поддерживающие их использование.

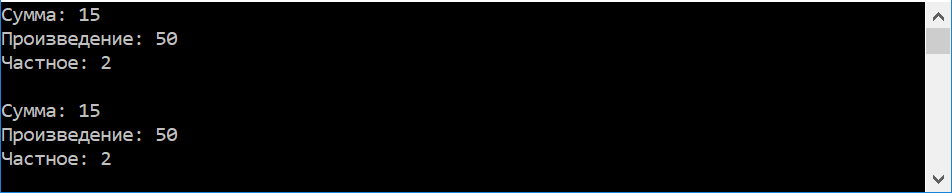
| **Методы и свойства делегатов** | **Назначение** |
| --- | --- |
| Method | Возвращает представляемый делегатом метод (объект типа System.Reflection.MethodInfo) |
| Target | Возвращает экземпляр класса, метод которого вызывает текущий делегат. Если возвращенное Target значение равно null, значит, подлежащий вызову метод является статическим |
| GetInvocationList() | Возвращает массив методов делегата, доступных для вызова |
| Combine() | Добавляет методы в список, поддерживаемый делегатом. В C# этот метод также вызывается за счет использования перегруженной операции += |
| Remove() RemoveAll() | Удаляют метод (или все методы) из списка вызовов делегата. В C# метод Remove() может быть вызван неявно, посредством перегруженной операции -= |

**Пример 3:**







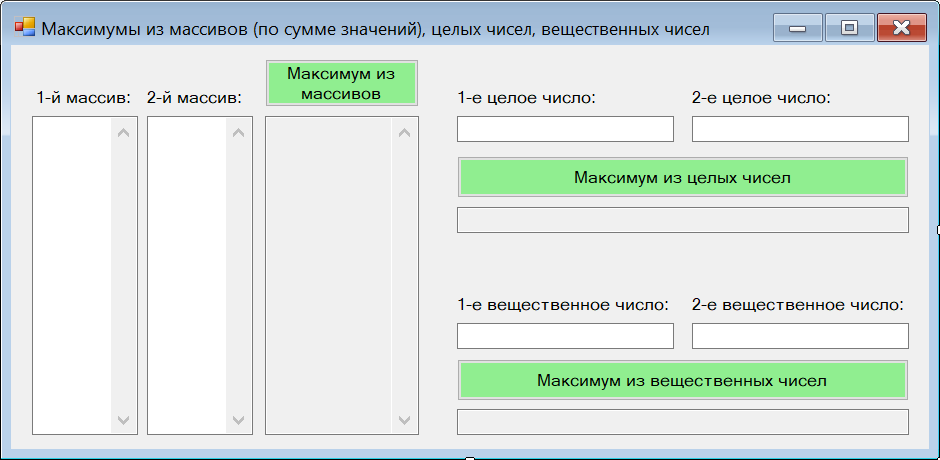


Главный вывод из данного примера заключается в следующем: вызов метода разрешается во время выполнения, а не в процессе компиляции.

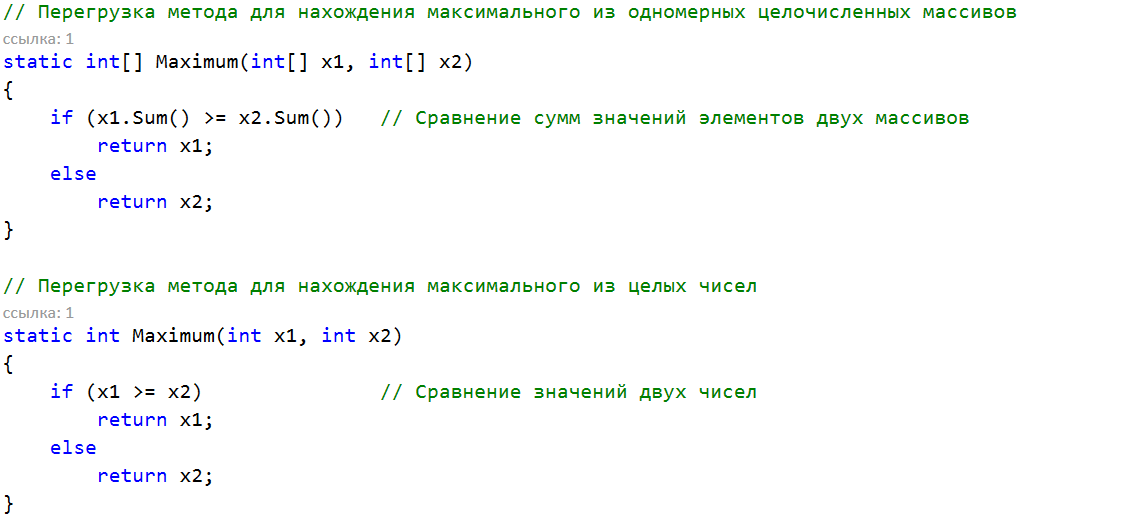
**Задание 1.** Найти максимальный объект в каждой паре заданных объектов: а) среди двух одномерных целочисленных массивов (выбрать максимальный по сумме значений элементов); б) среди двух целых чисел; в) среди двух вещественных чисел. Для этого создать в классе перегруженные методы, различающиеся типом возвращаемых значений и типами аргументов. Разработать приложение, осуществляющее обращение к созданным методам.

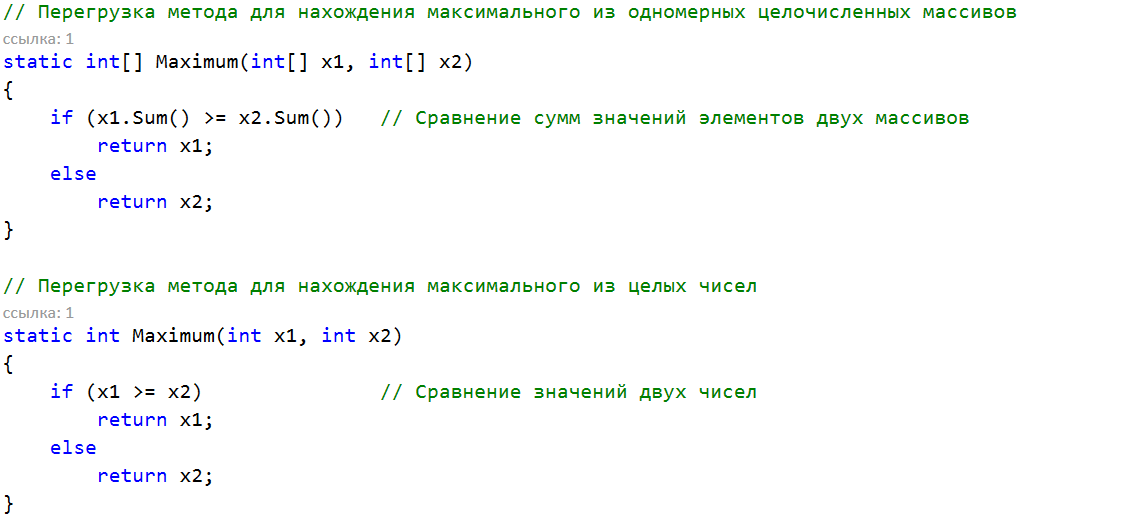
**Решение.**

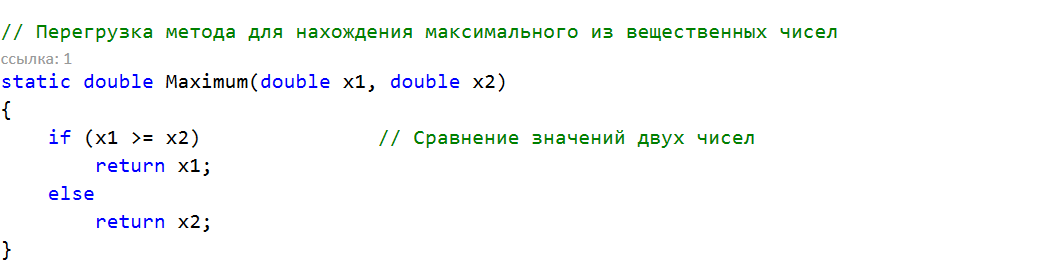
а) Интерфейс приложения Windows Forms:



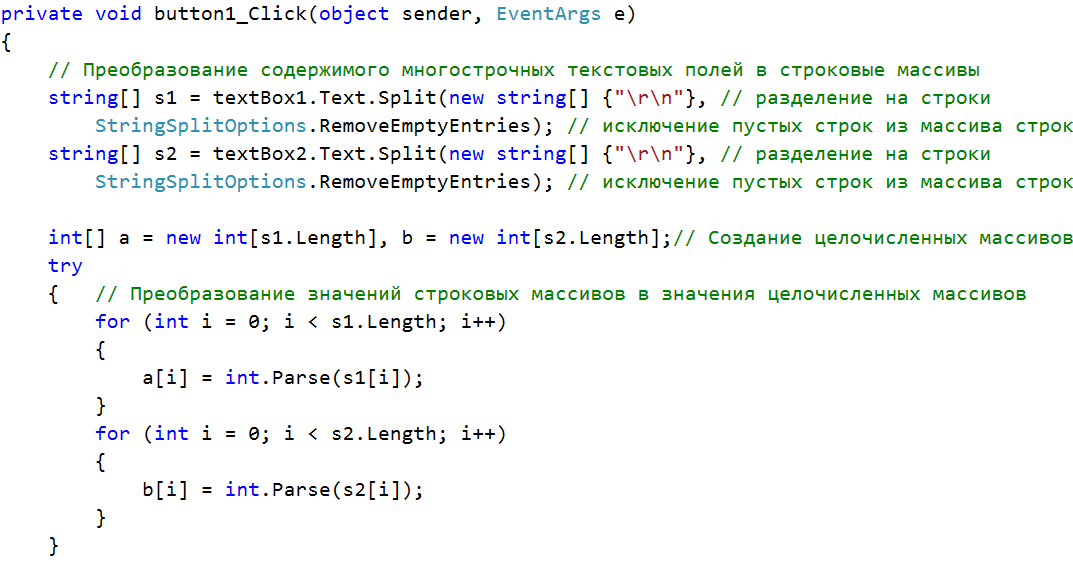
Коды перегрузок метода Maximum(x1, x2) для нахождения максимального объекта в каждой паре заданных объектов:

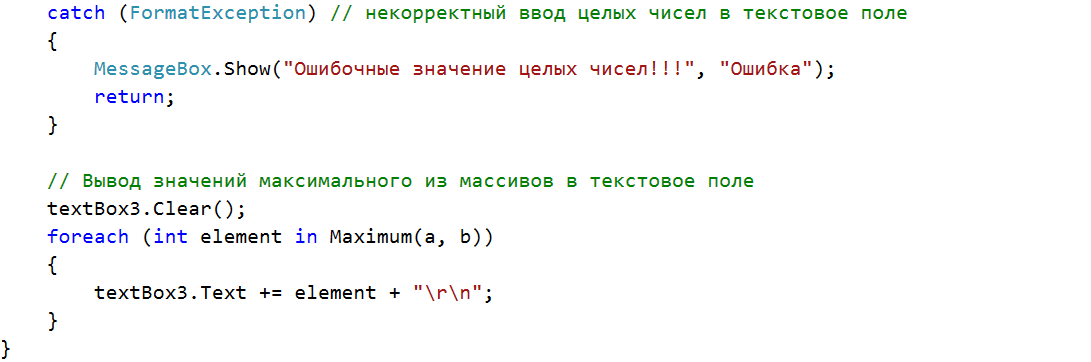




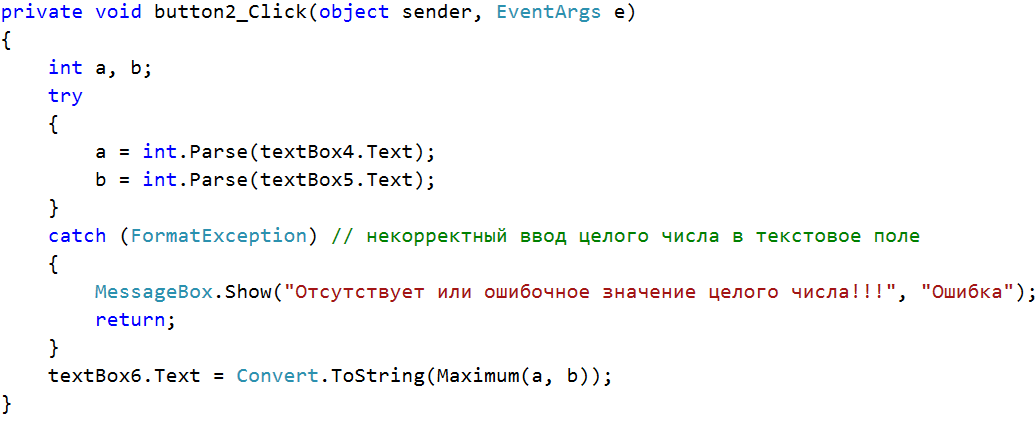


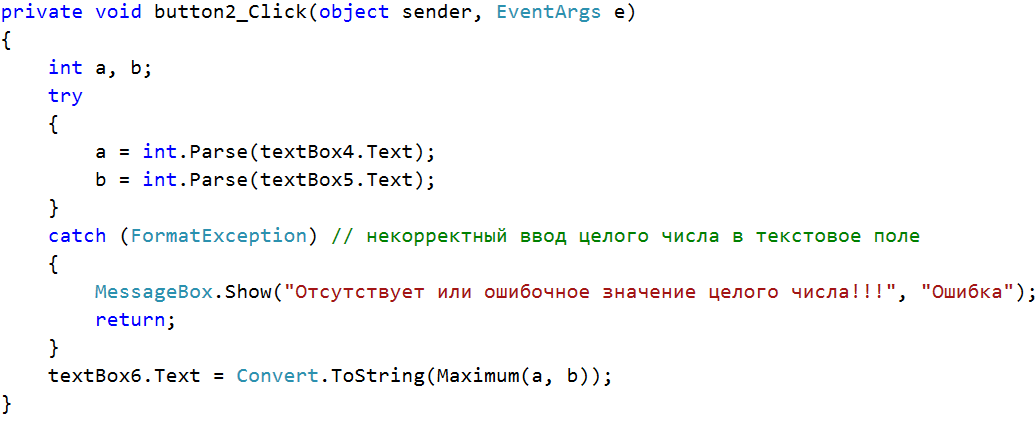
Код, выполняемый при щелчке на кнопку «Максимум из массивов»:



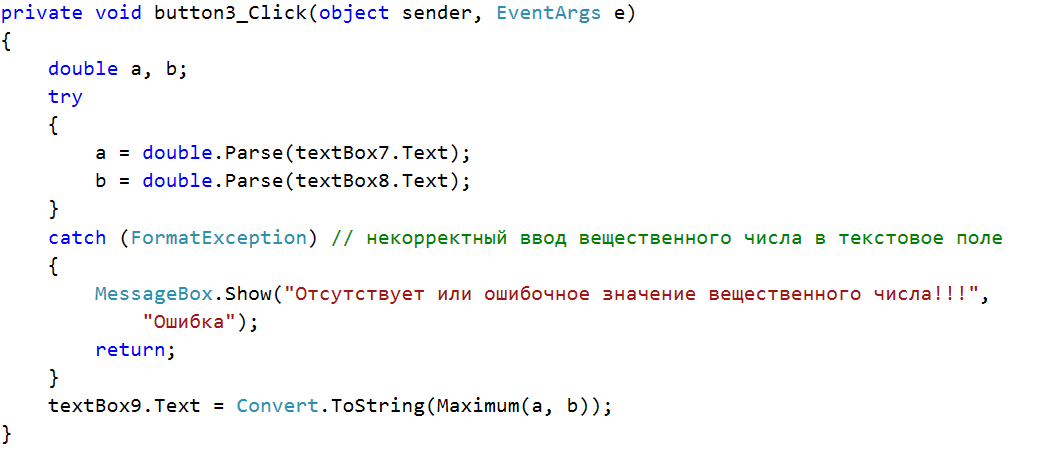


Код, выполняемый при щелчке на кнопку «Максимум из целых чисел»:

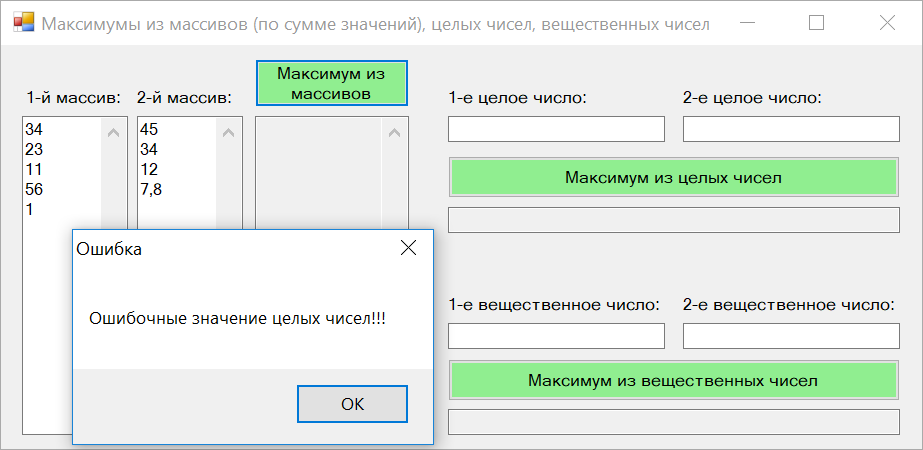


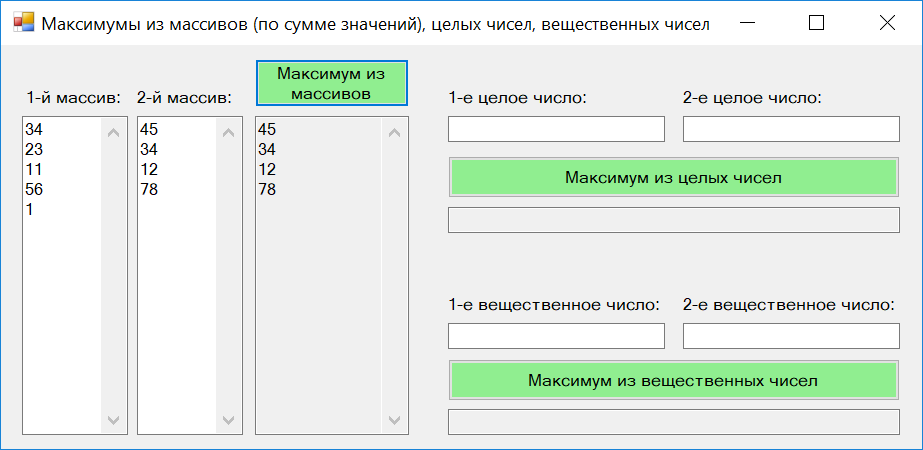


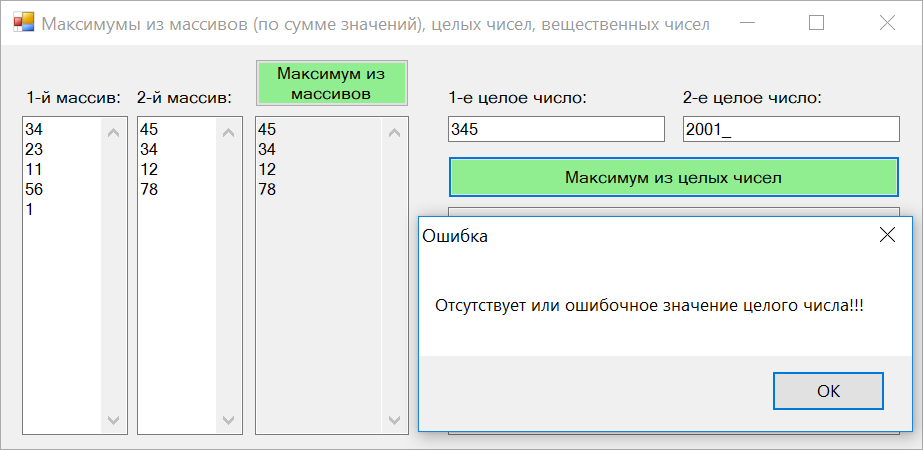
Код, выполняемый при щелчке на кнопку «Максимум из вещественных чисел»:

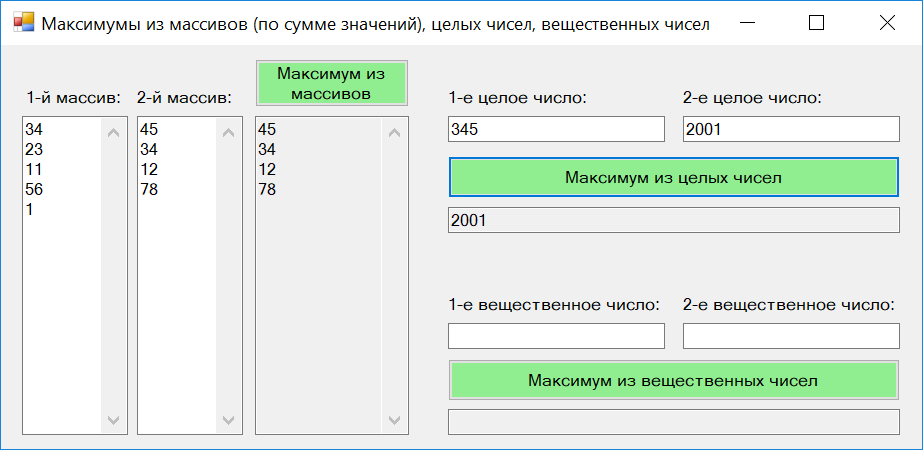


Скрины отлаженной программы:

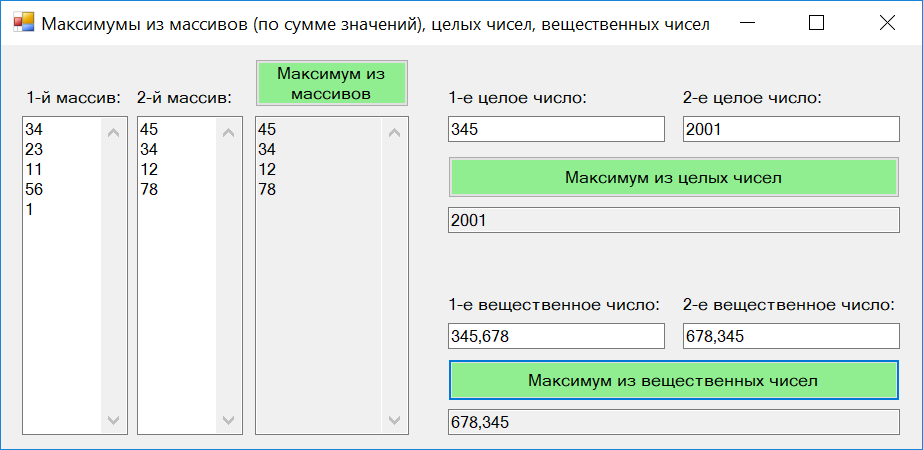




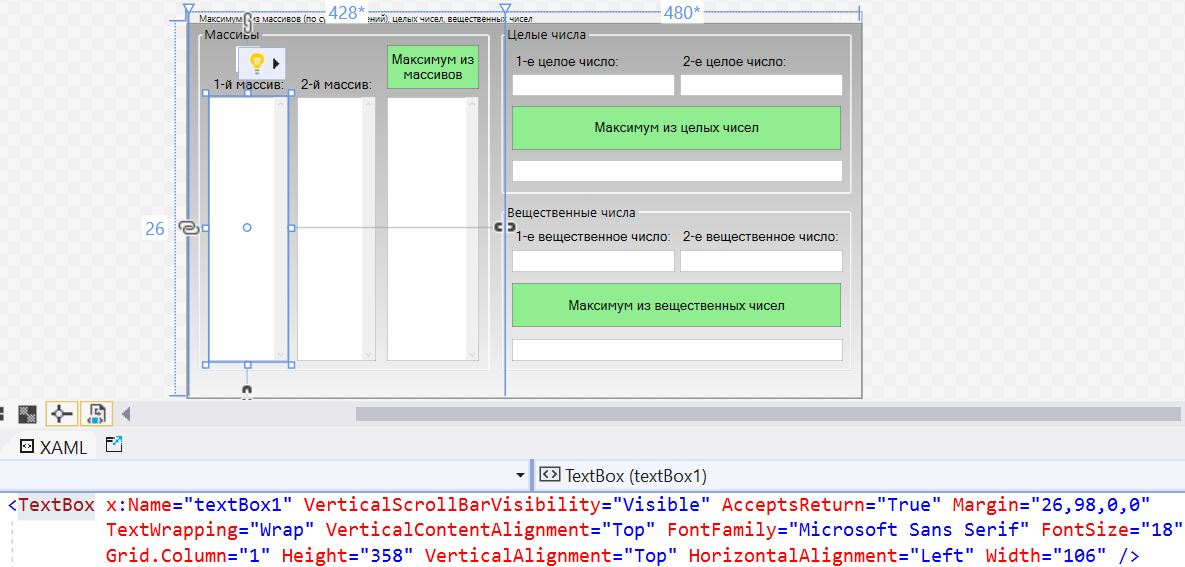




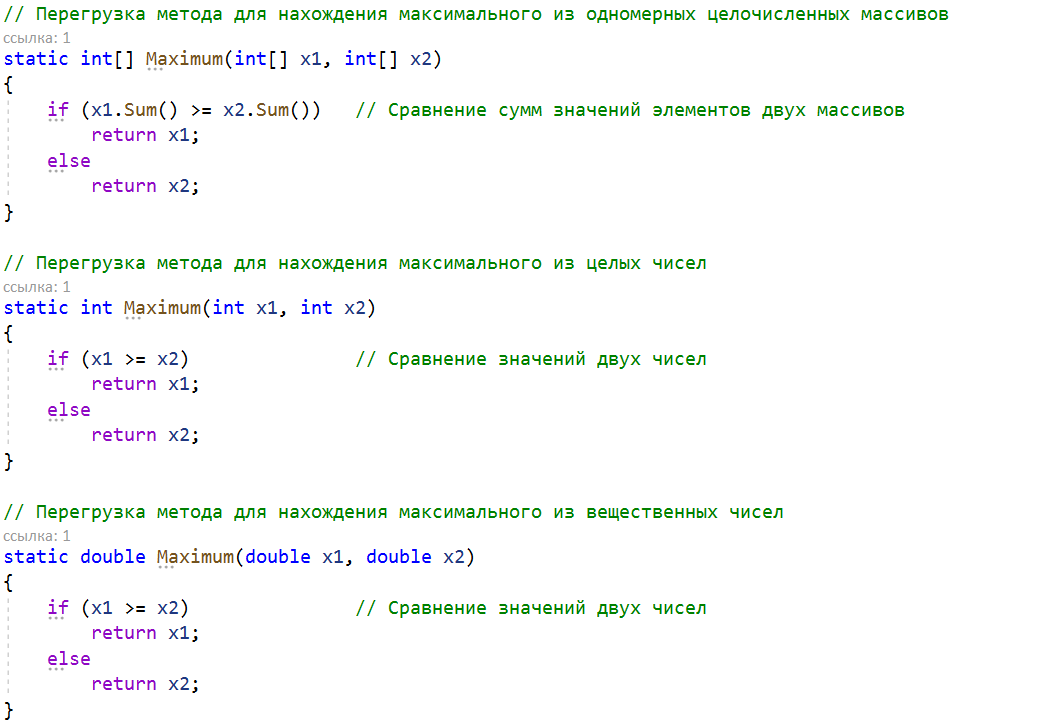




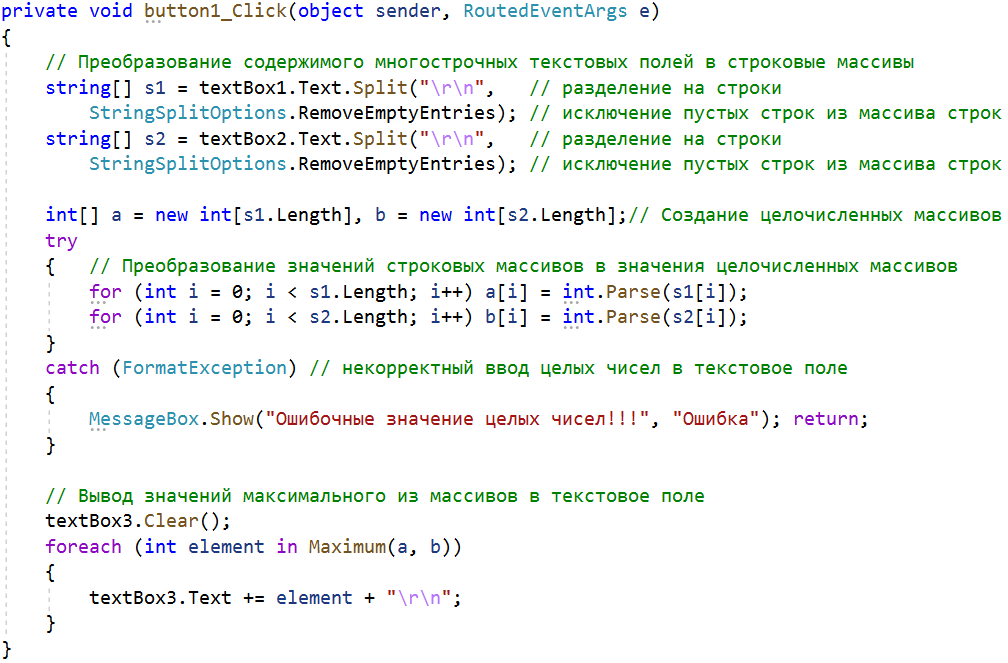
б) Интерфейс приложения WPF:

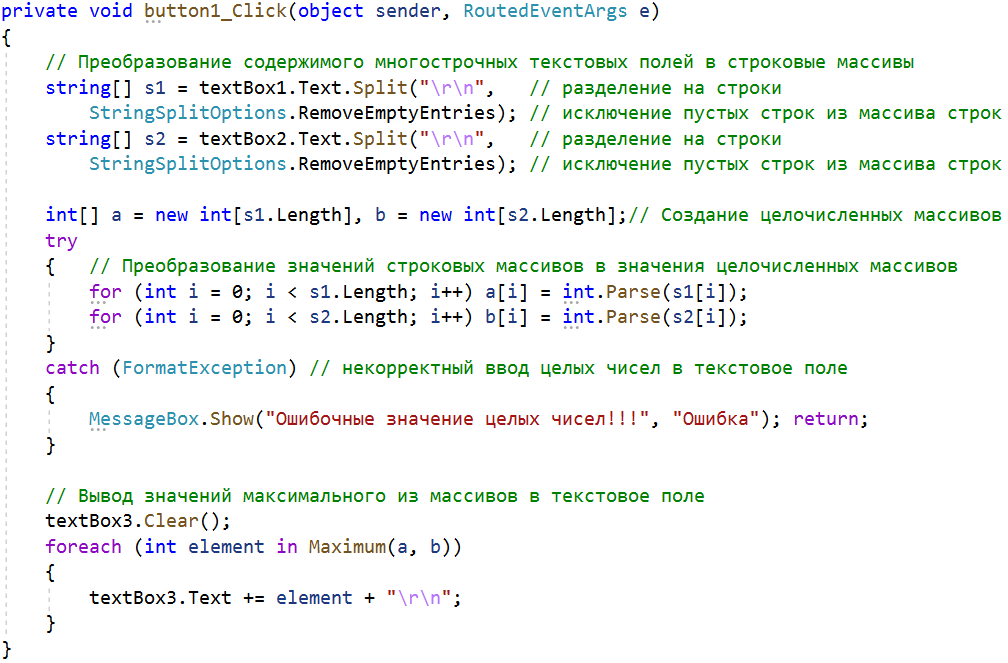


Коды перегрузок метода Maximum(x1, x2) для нахождения максимального объекта в каждой паре заданных объектов:

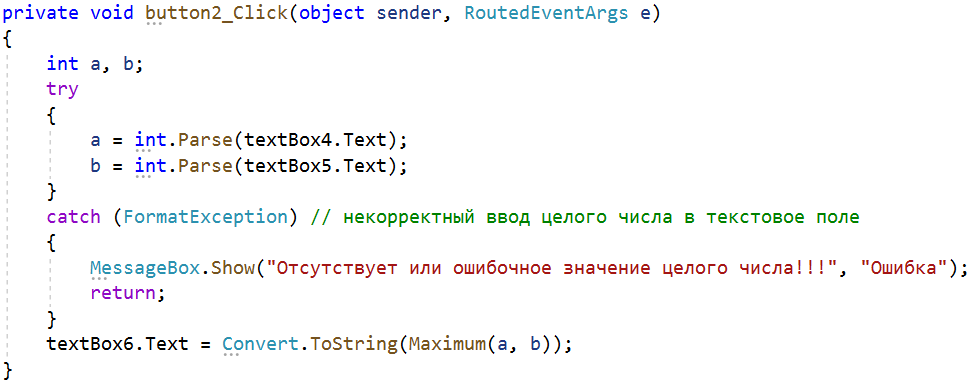


Код, выполняемый при щелчке на кнопку «Максимум из массивов»:

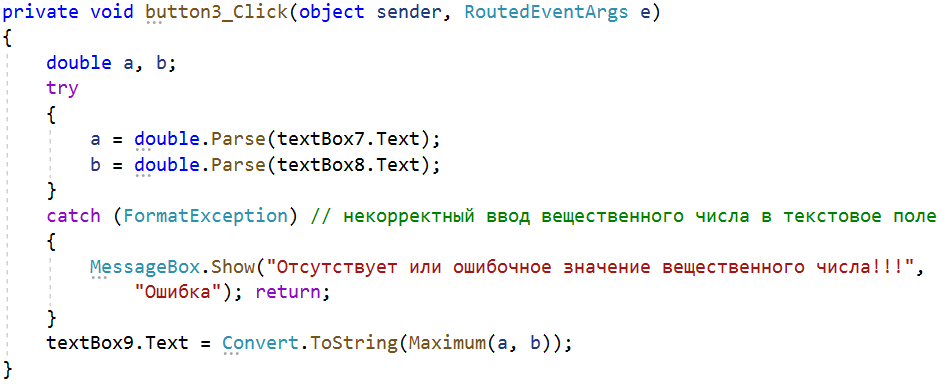




Код, выполняемый при щелчке на кнопку «Максимум из целых чисел»:

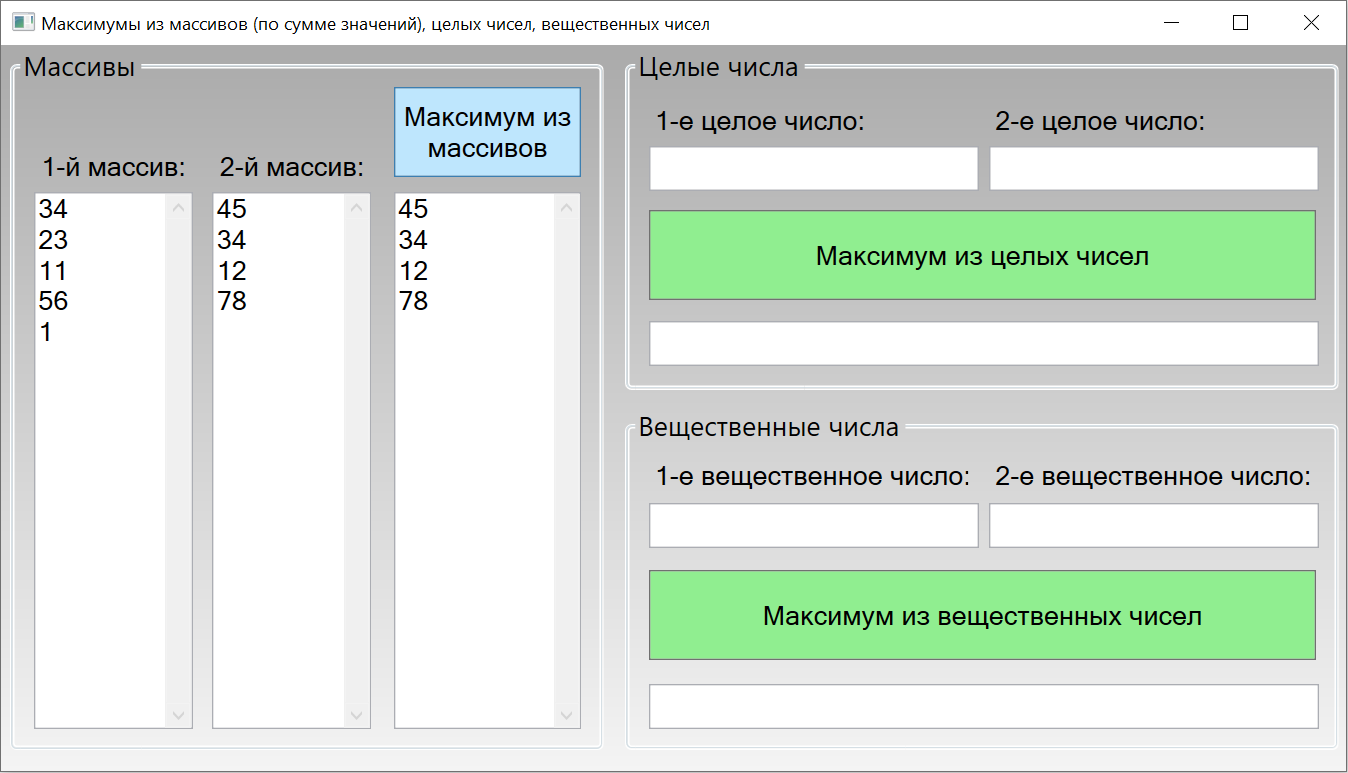


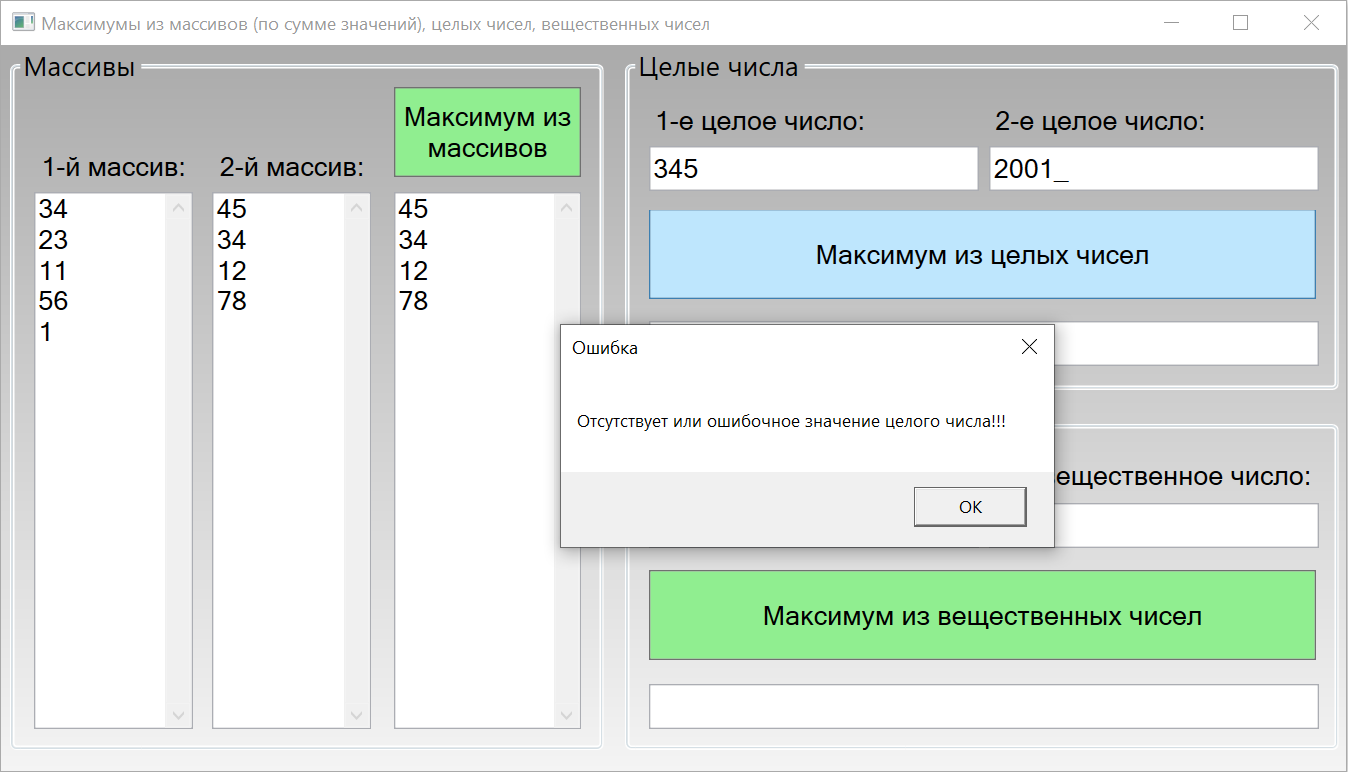
Код, выполняемый при щелчке на кнопку «Максимум из вещественных чисел»:

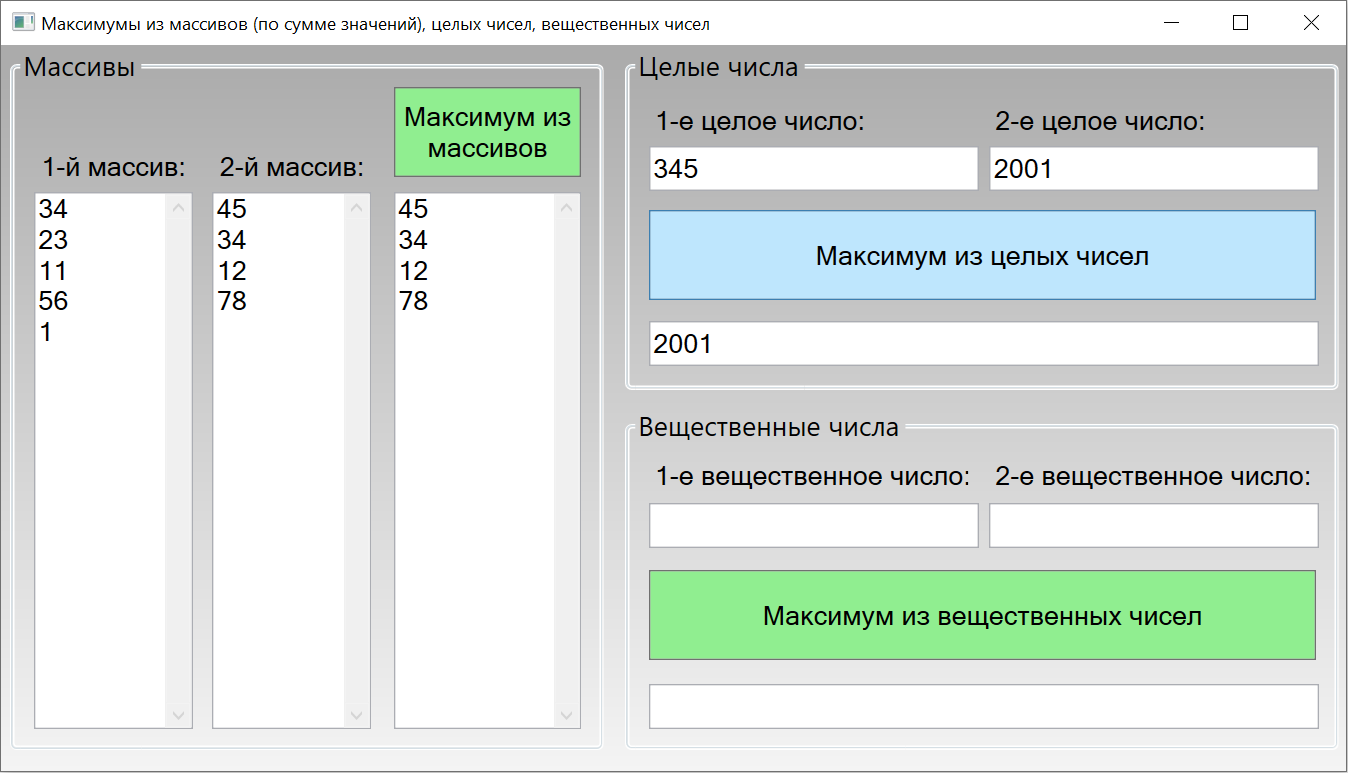


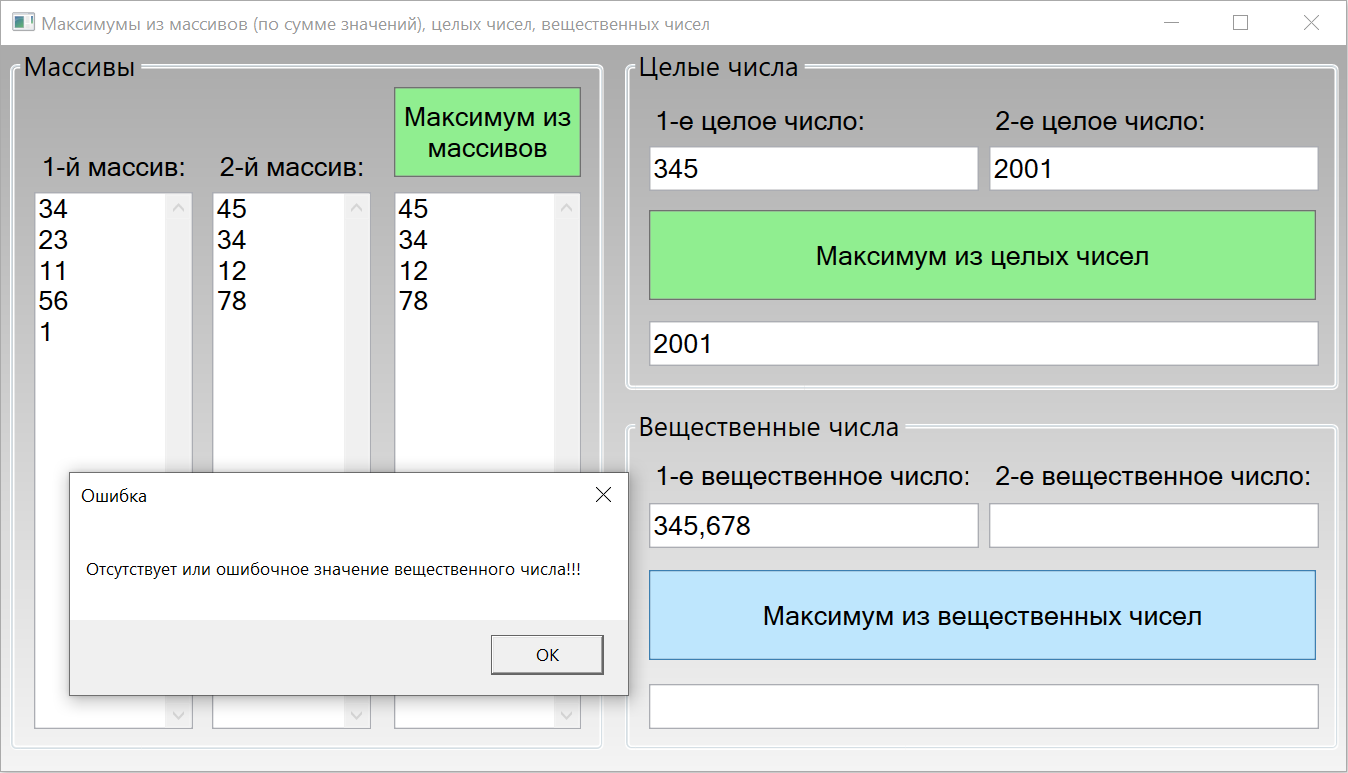
Скрины отлаженной программы:

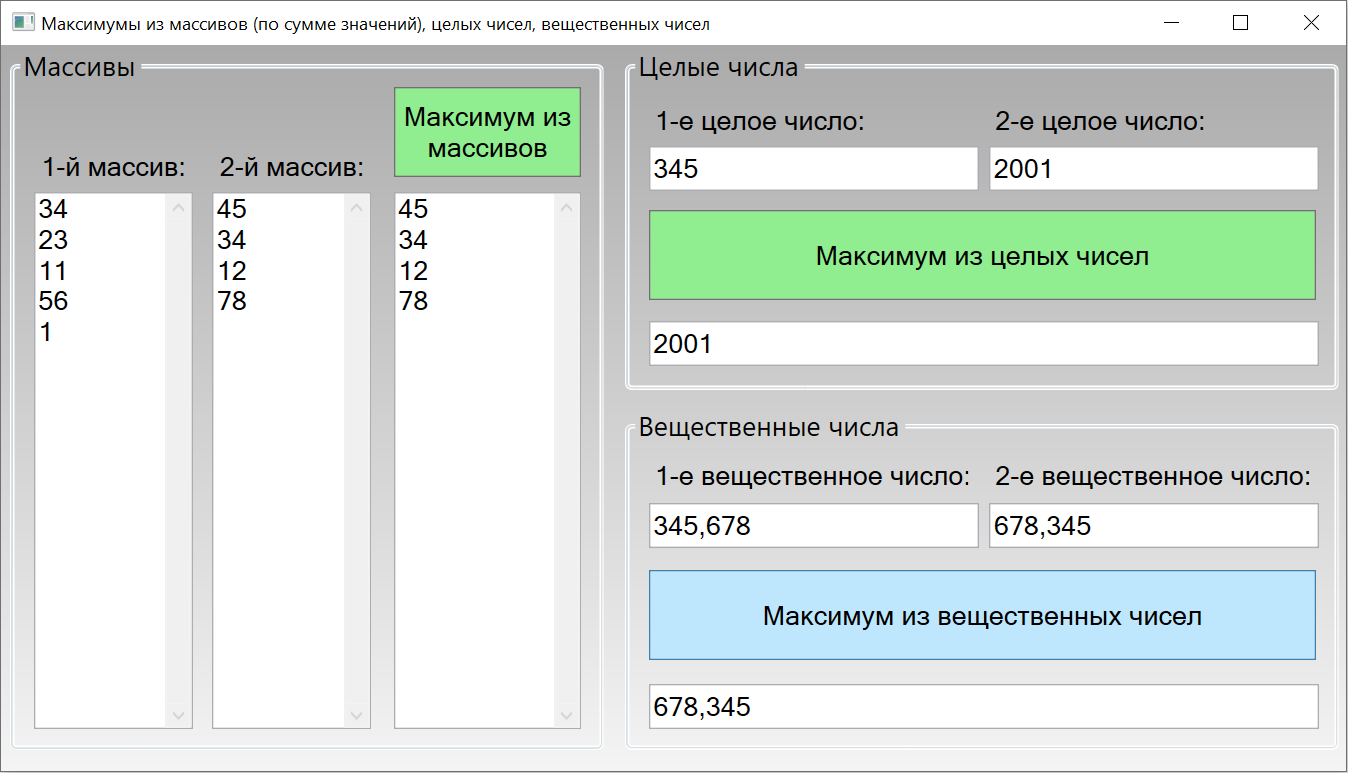




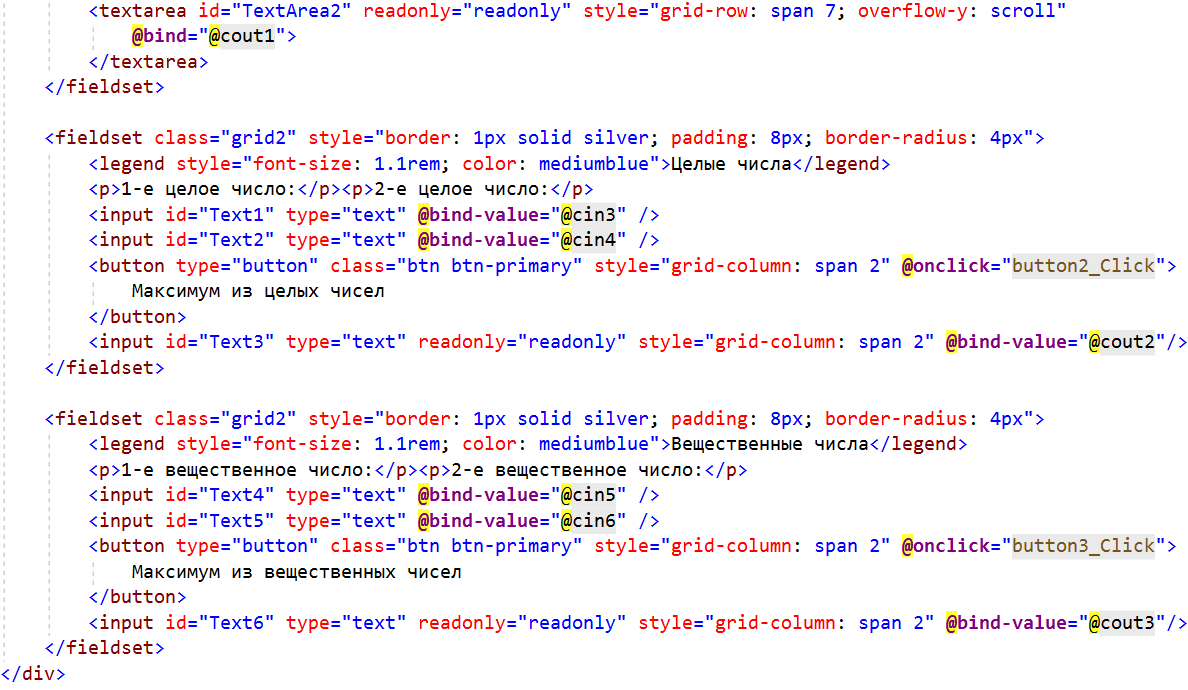
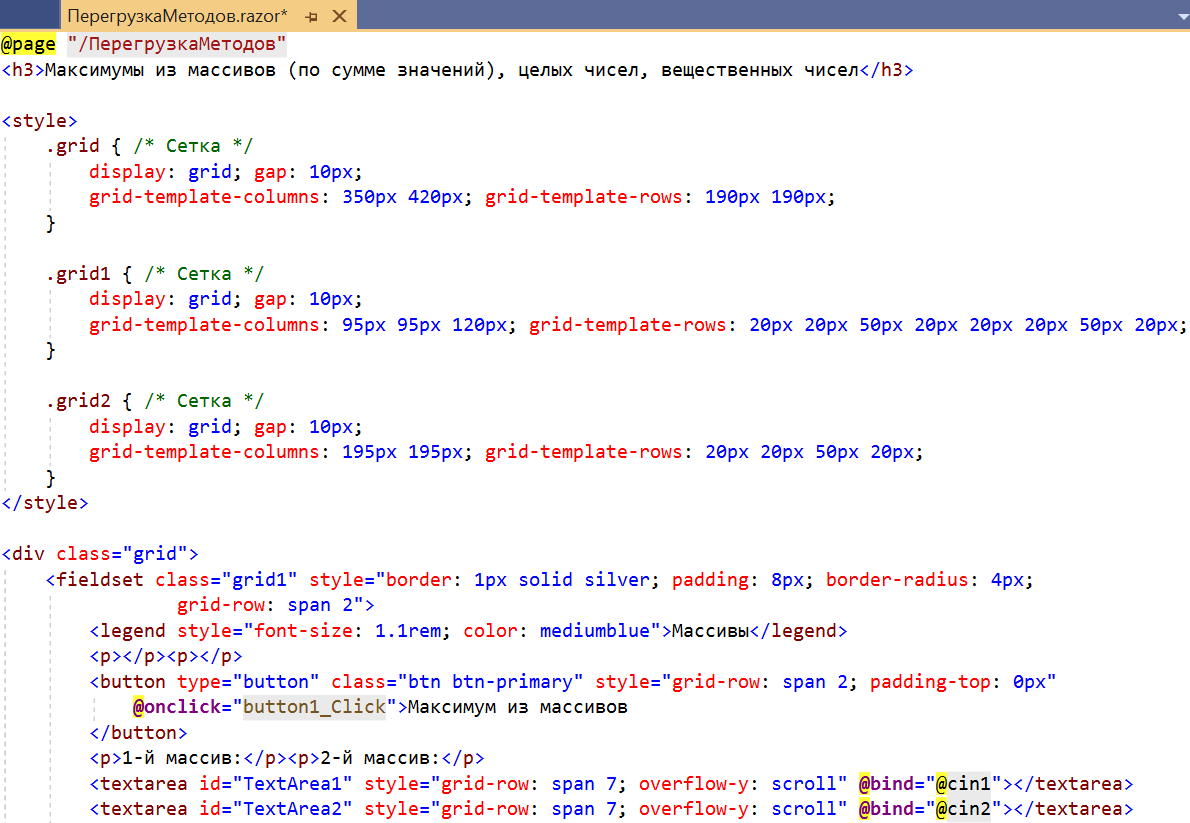


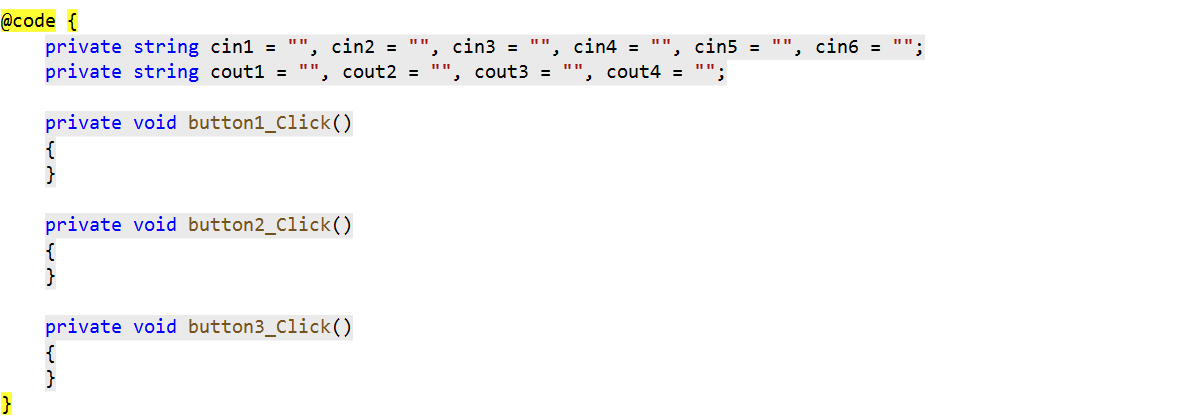


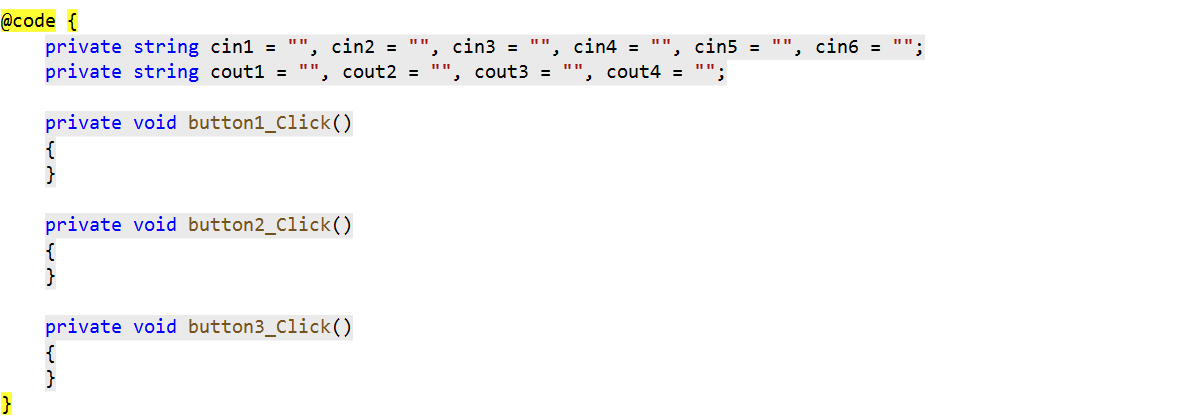


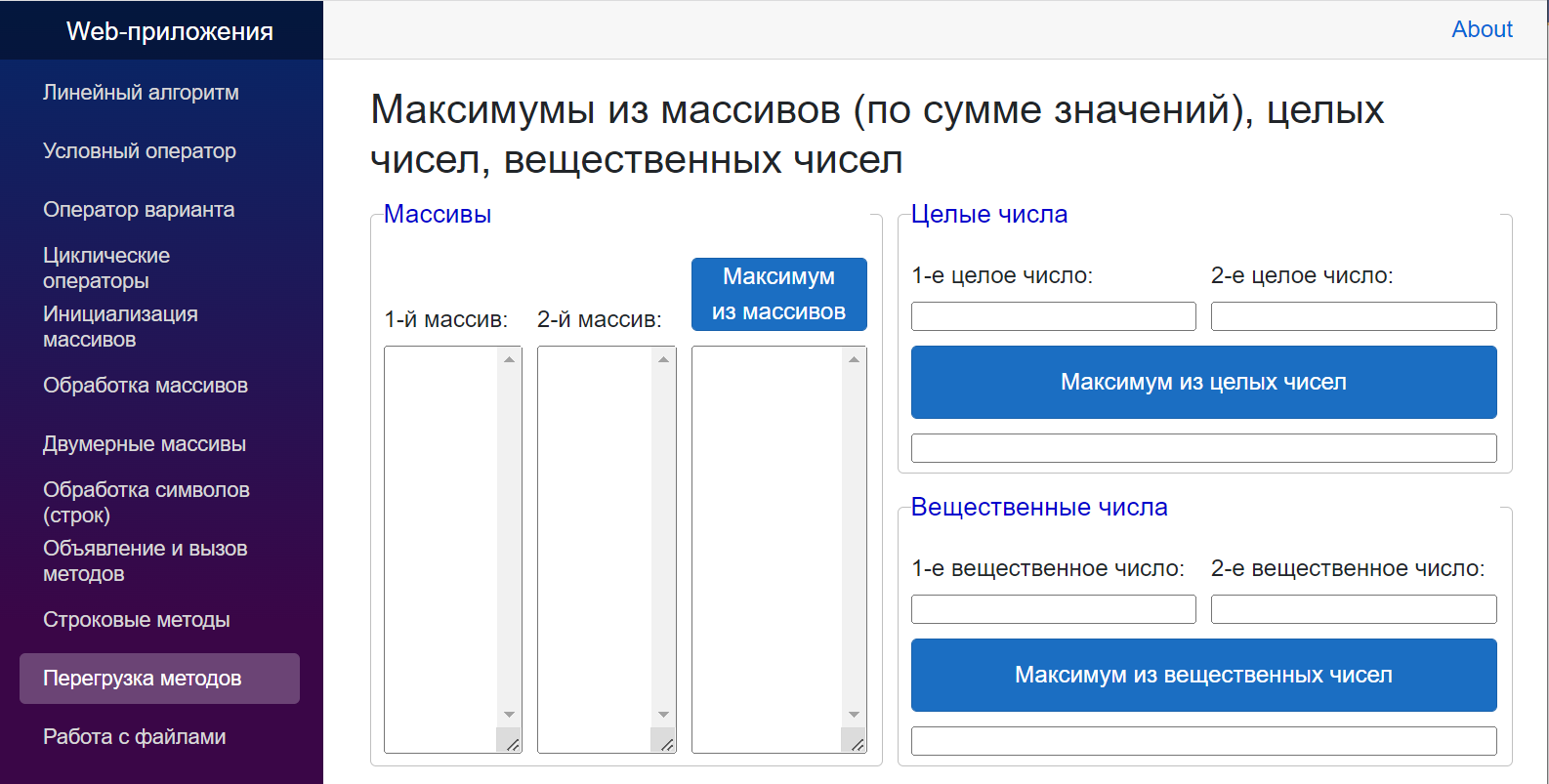


в) Интерфейс приложения WebAssembly Blazor:



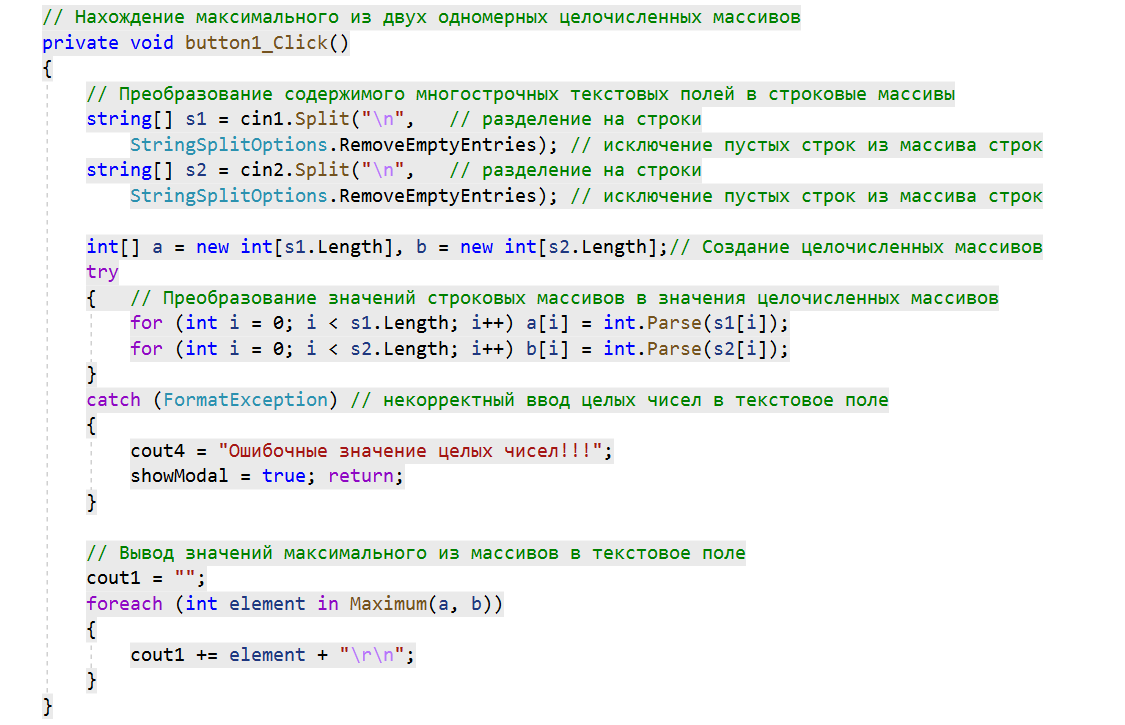


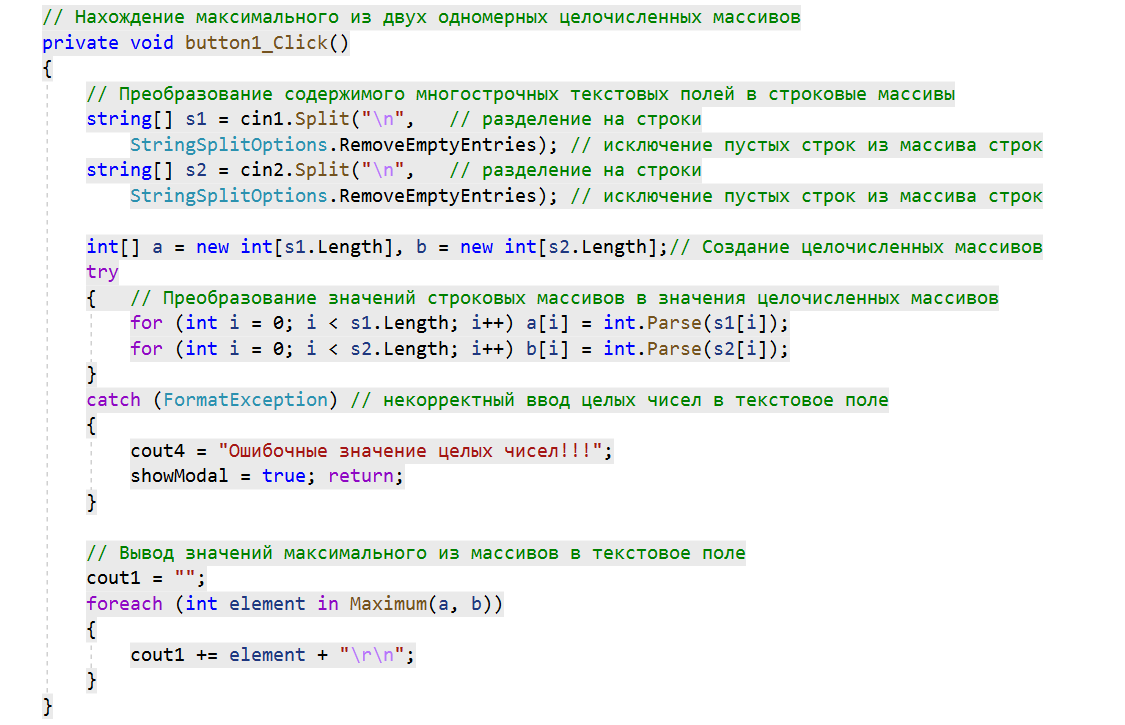


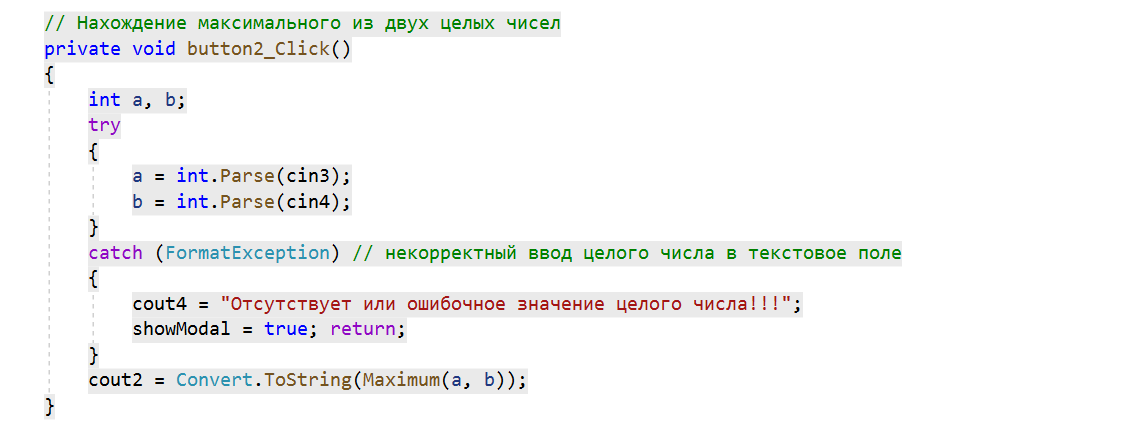


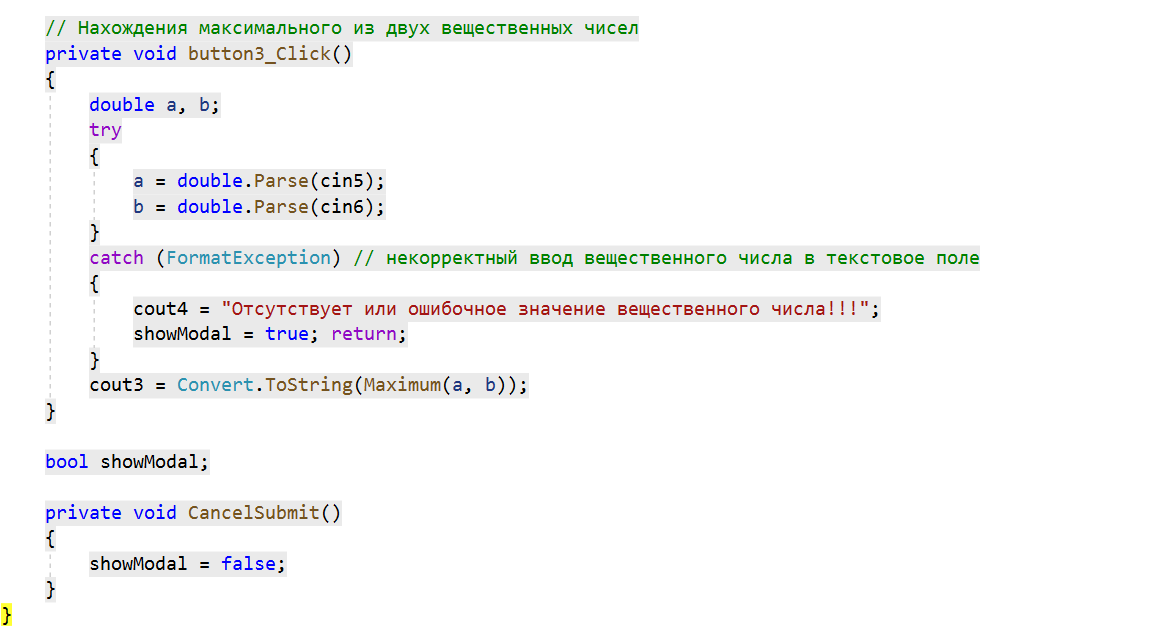
Коды приложения:

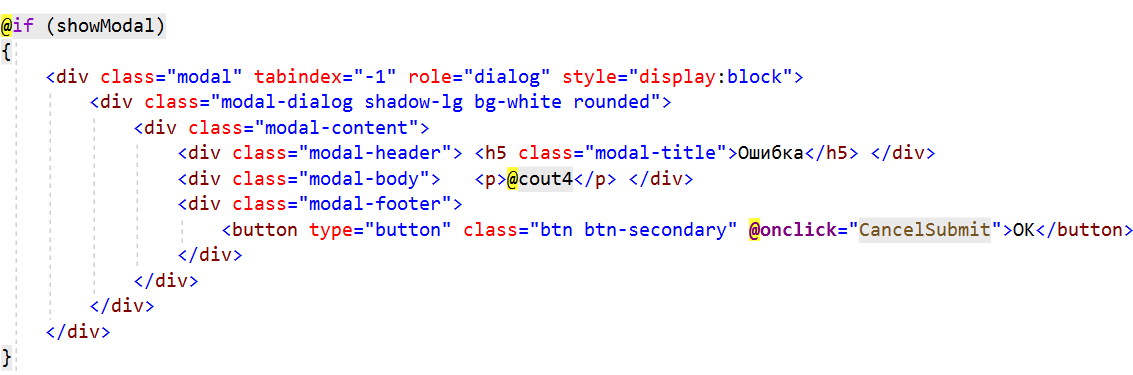


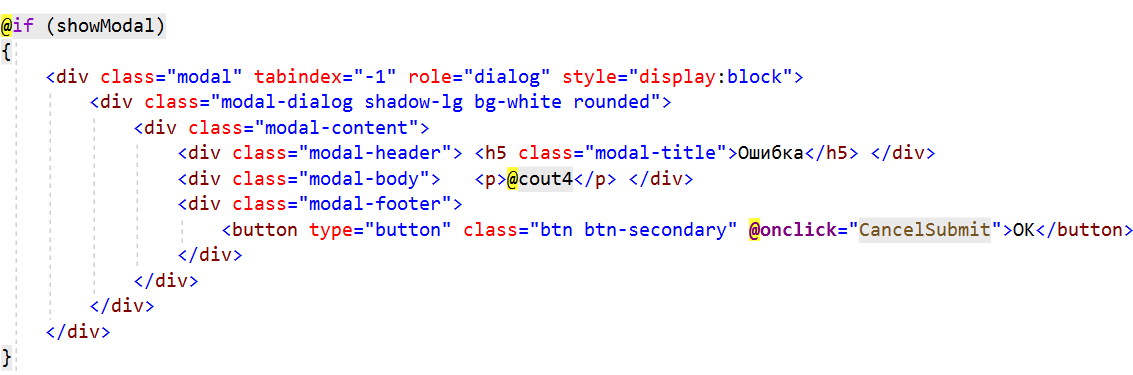




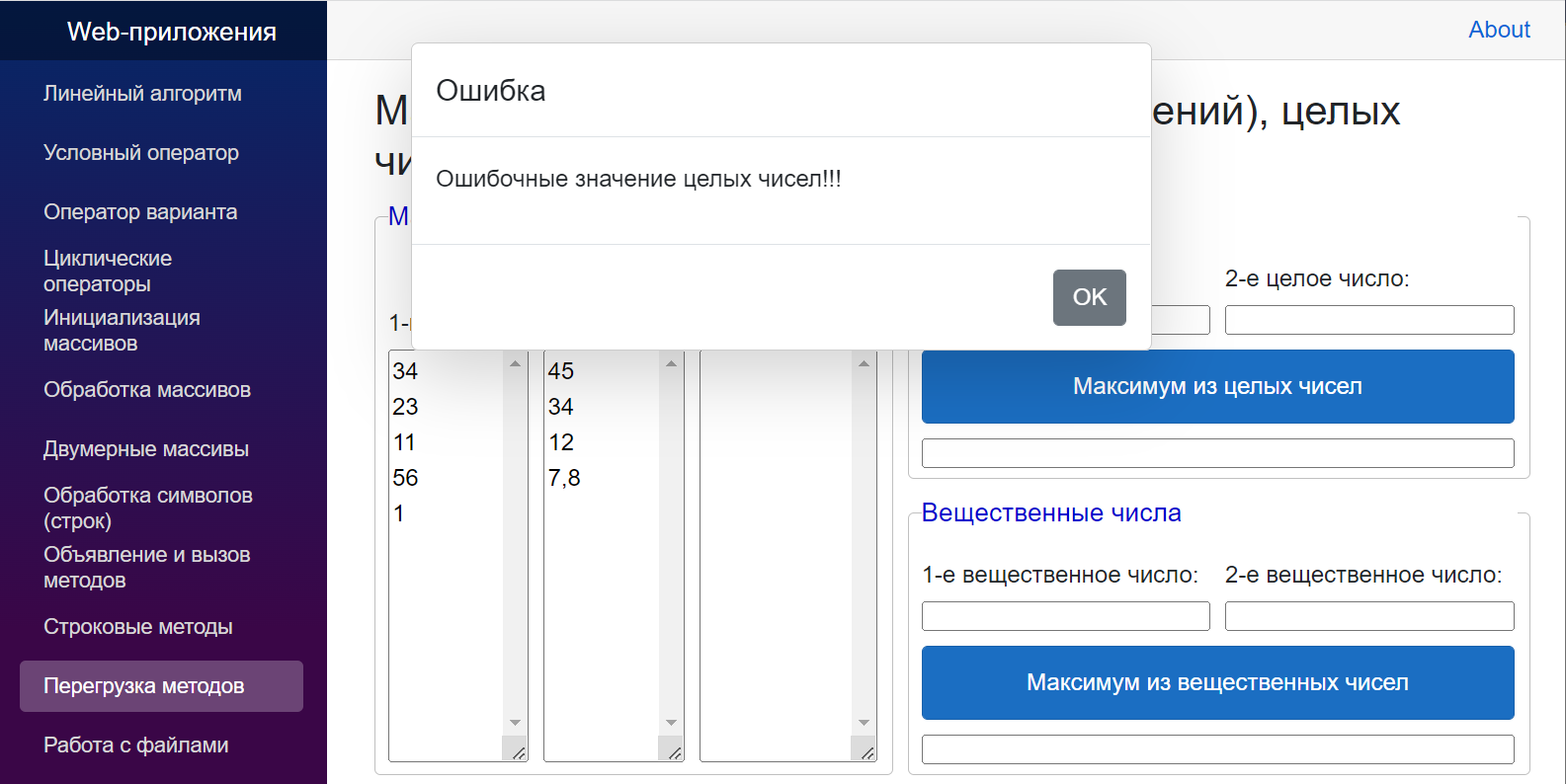


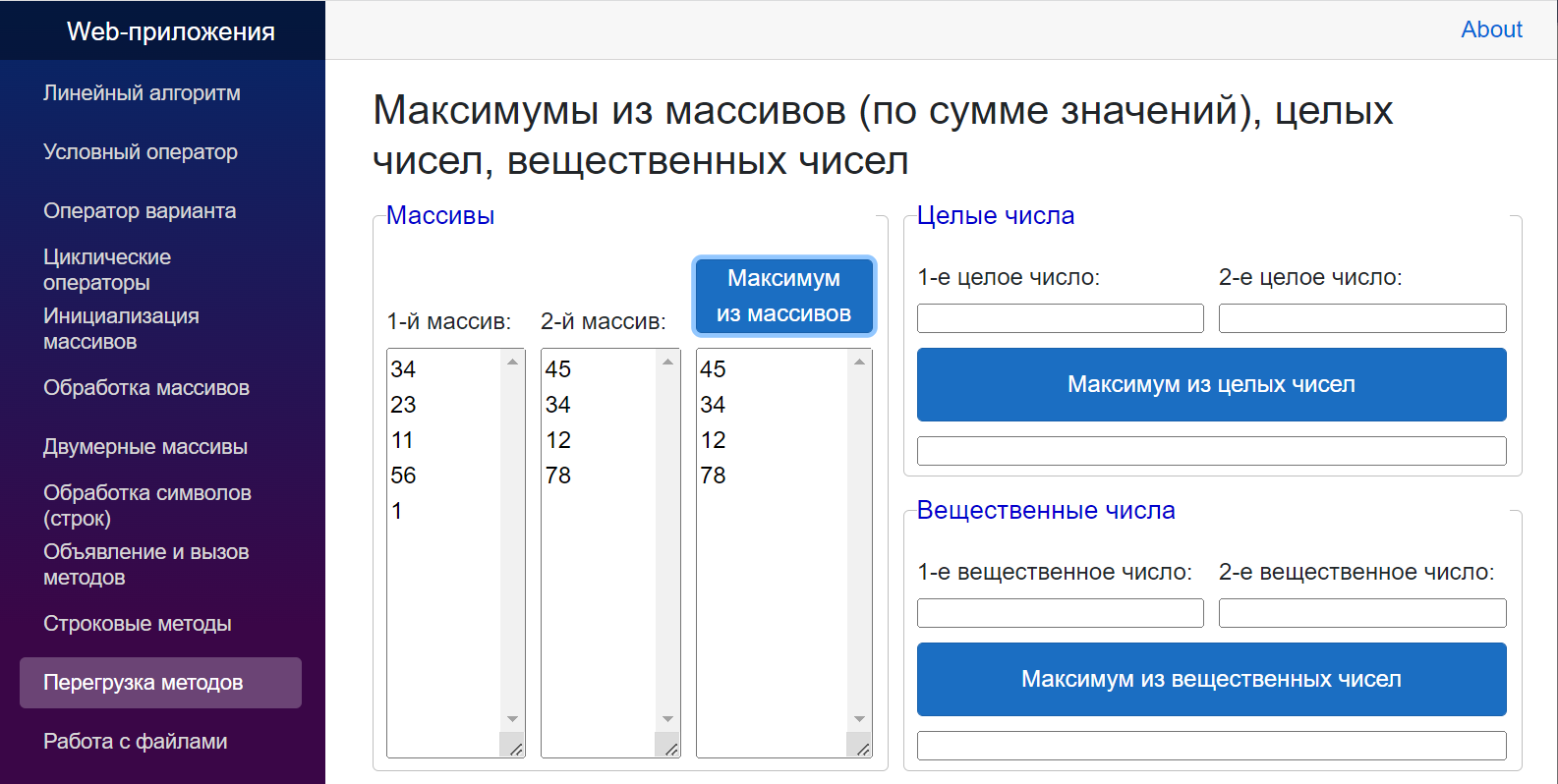


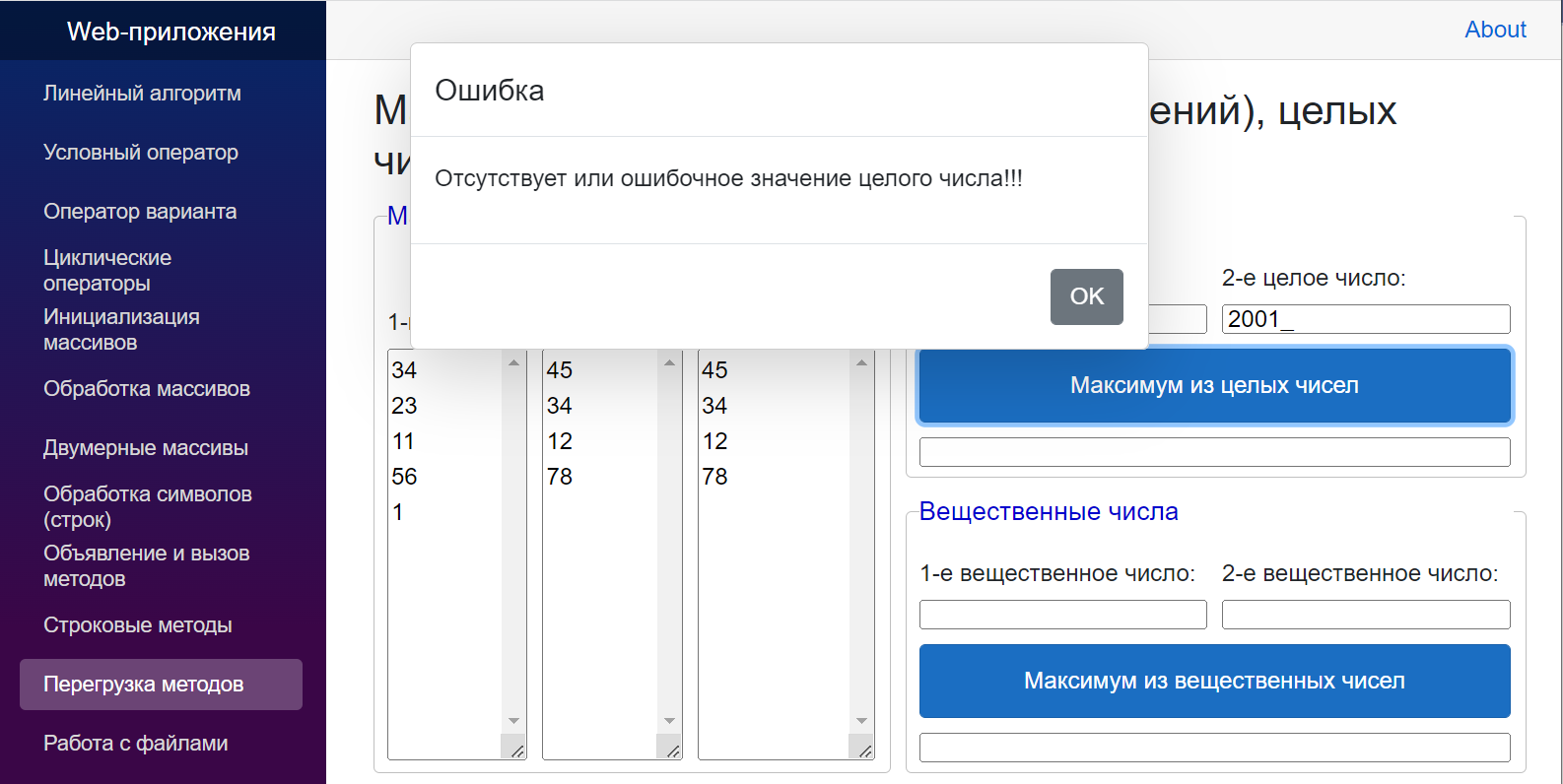


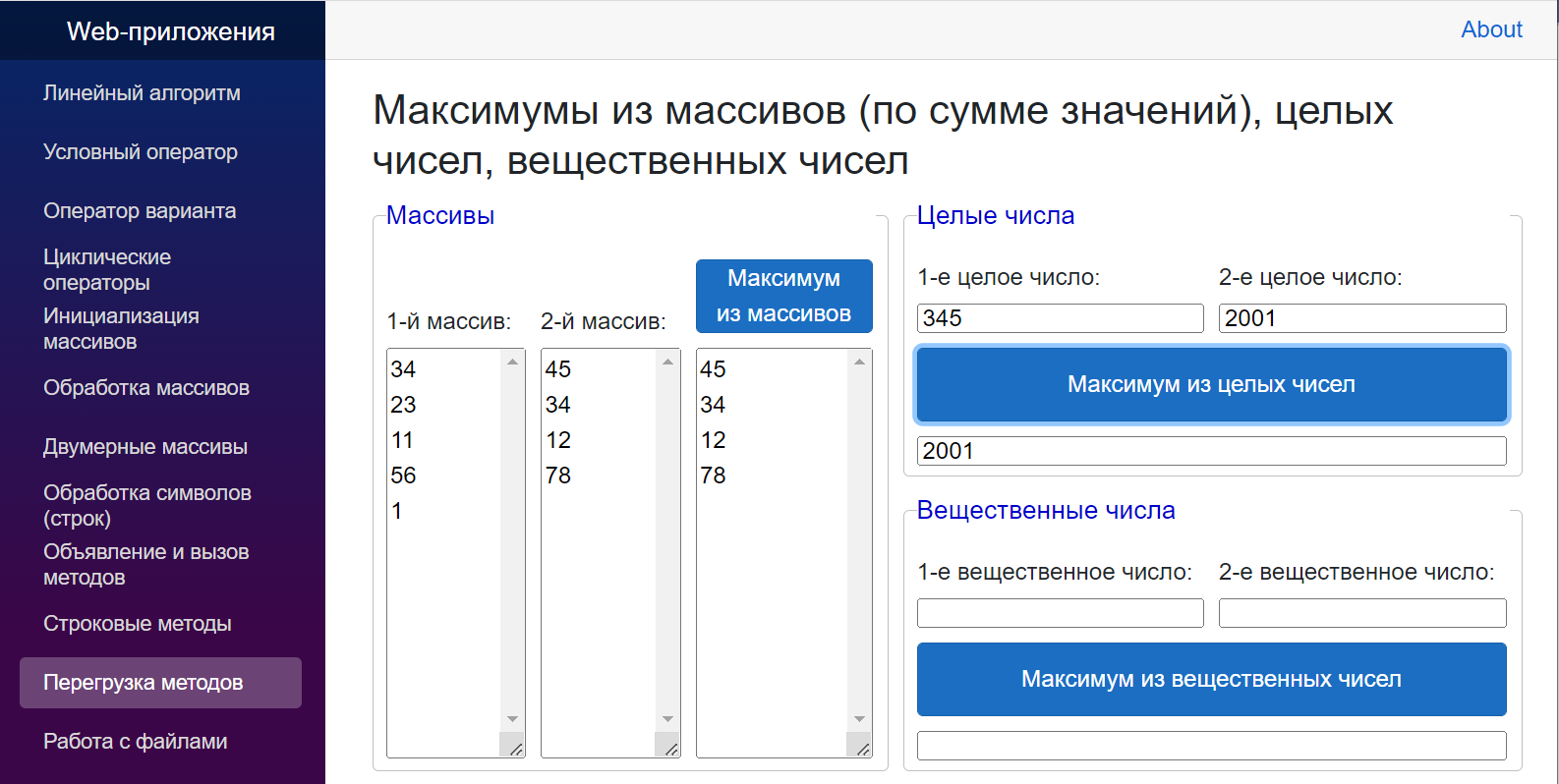


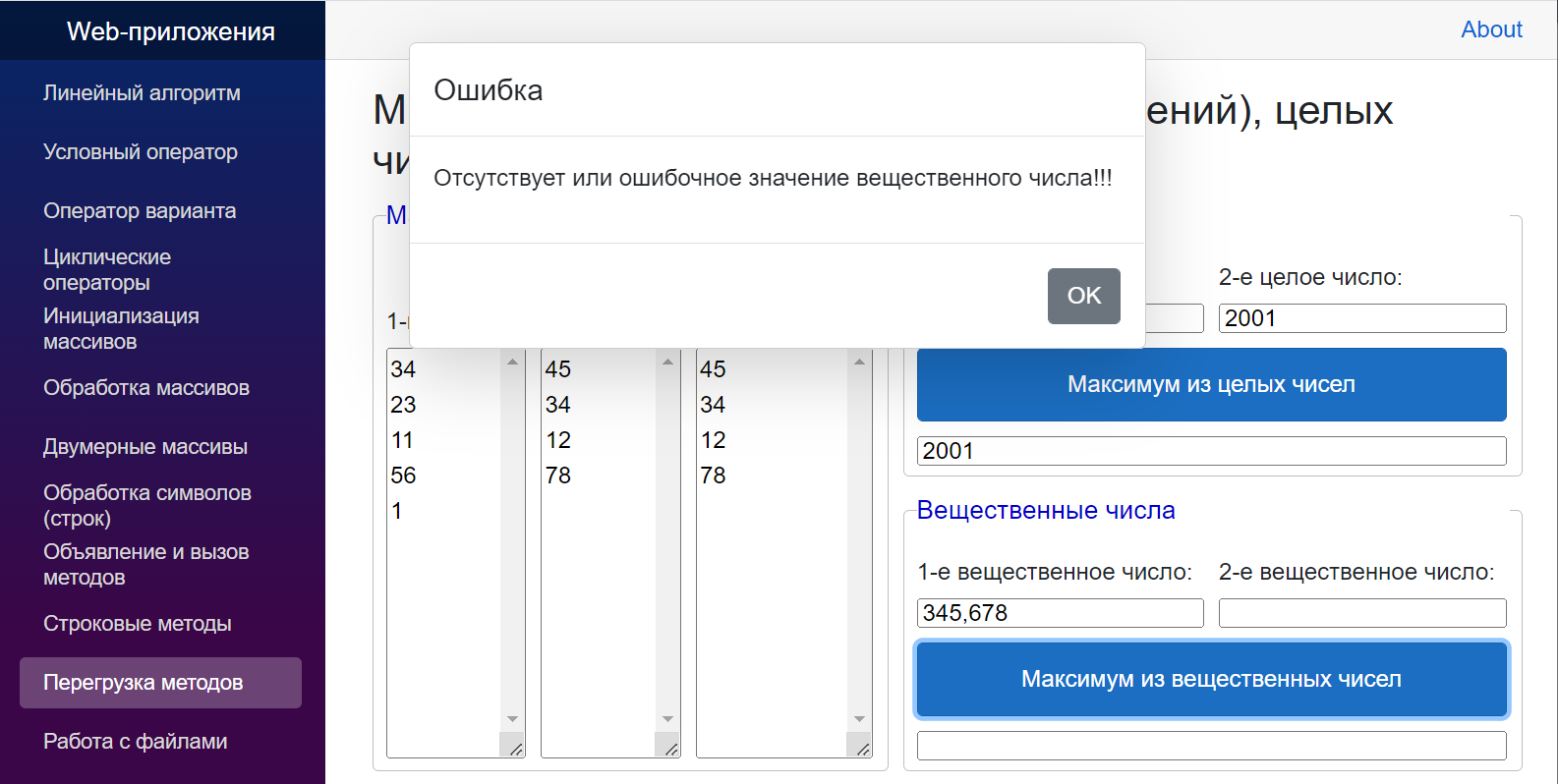
Скрины отлаженной программы:

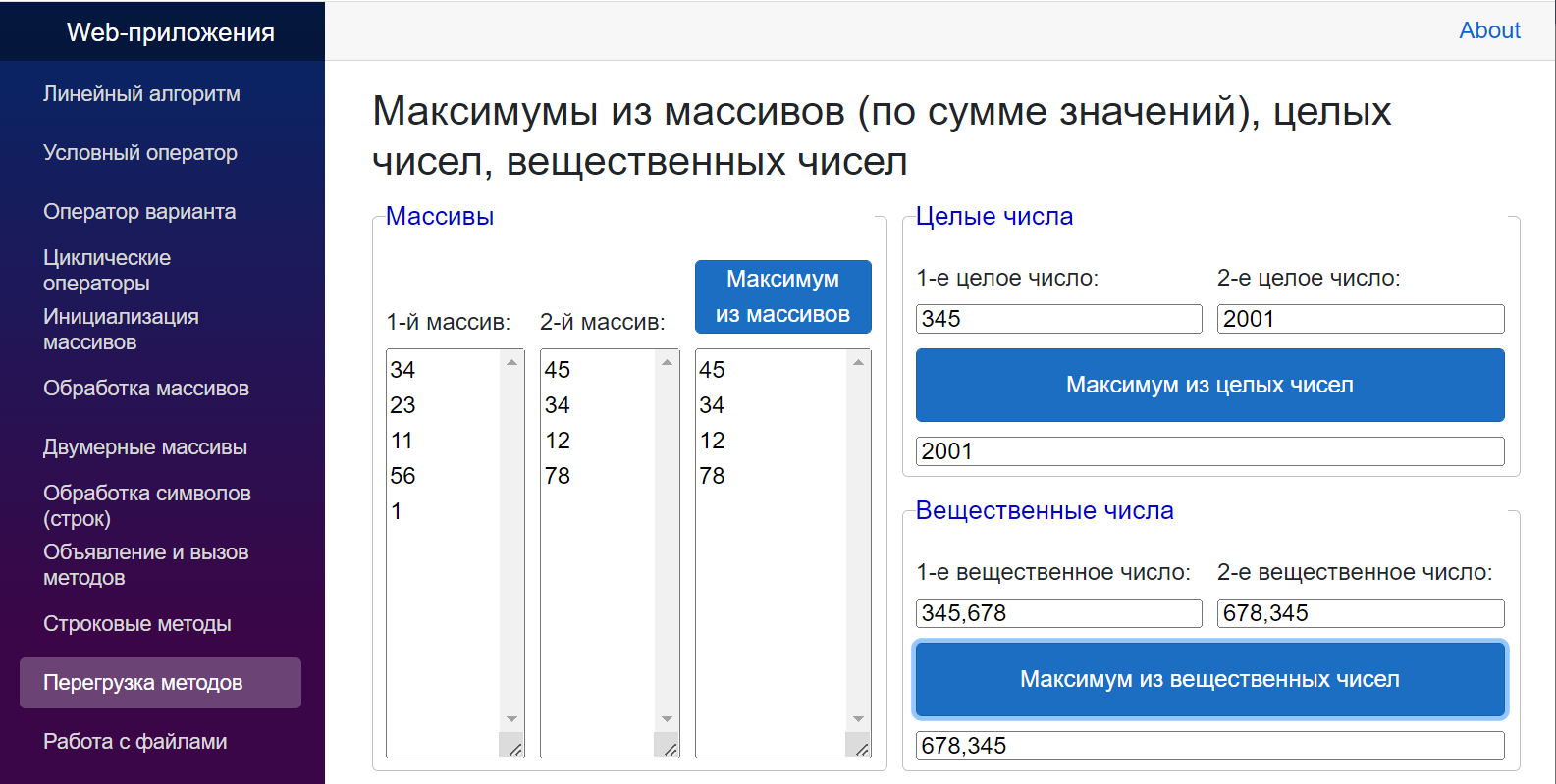








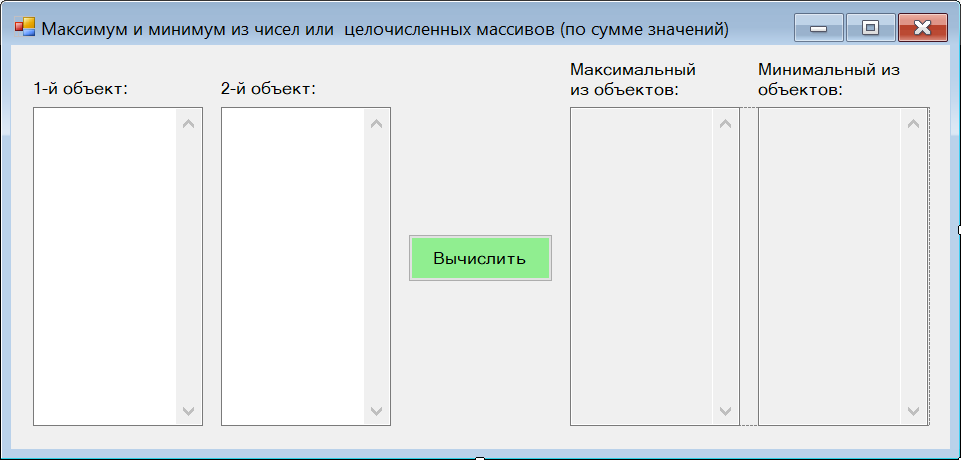




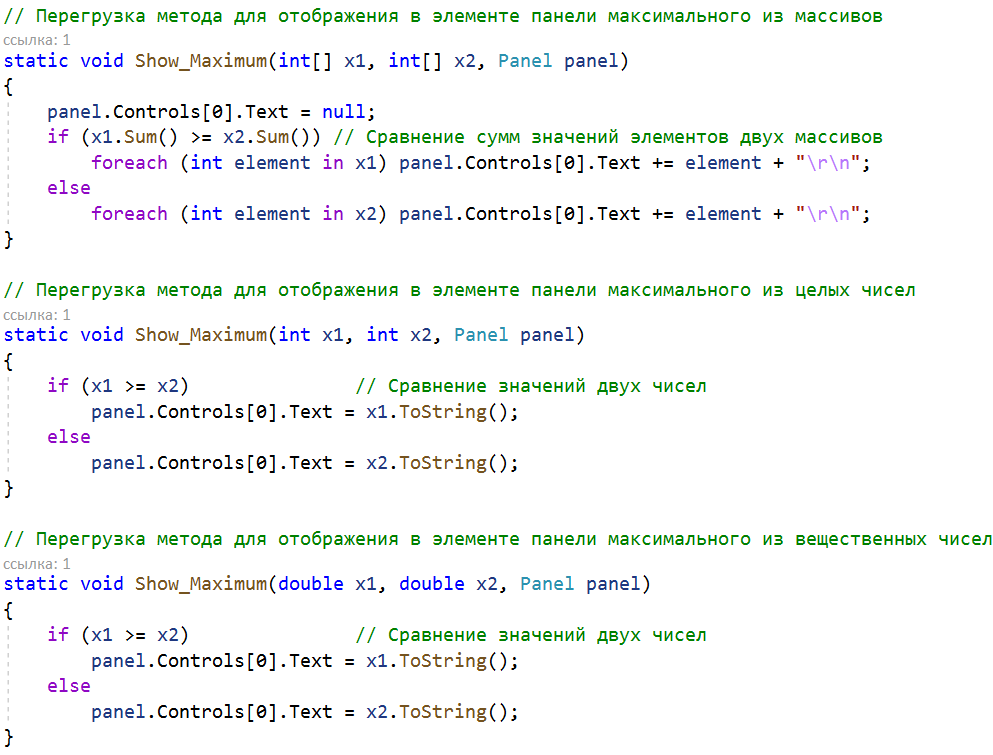
**Задание 2.** Отобразить (вывести) максимальный и минимальный объект в каждой паре заданных объектов: а) среди двух одномерных целочисленных массивов (выбрать максимальный и минимальный по сумме значений элементов); б) среди двух целых чисел; в) среди двух вещественных чисел. Для этого создать в классе перегруженные методы типа void, различающиеся типами аргументов. Разработать приложение, использующее делегаты для вызова списков разработанных методов.

**Решение.**

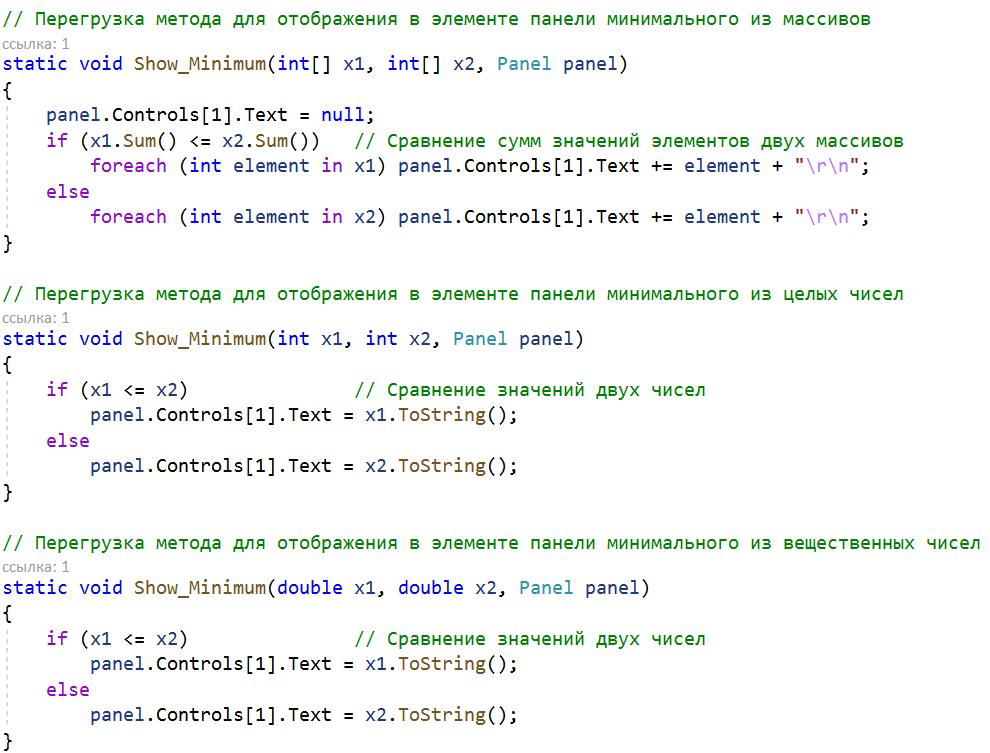
а) Интерфейс приложения Windows Forms:



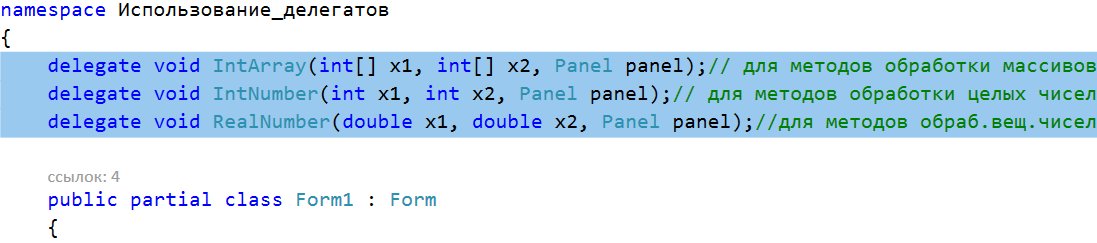
Коды перегрузок метода Show\_Maximum(x1, x2, Panel panel) для отображения в элементе, содержащемся в панели (текстовом поле), максимального объекта в каждой паре заданных объектов:



Коды перегрузок метода Show\_Minimum(x1, x2, Panel panel) для отображения в элементе, содержащемся в панели (текстовом поле), минимального объекта в каждой паре заданных объектов:

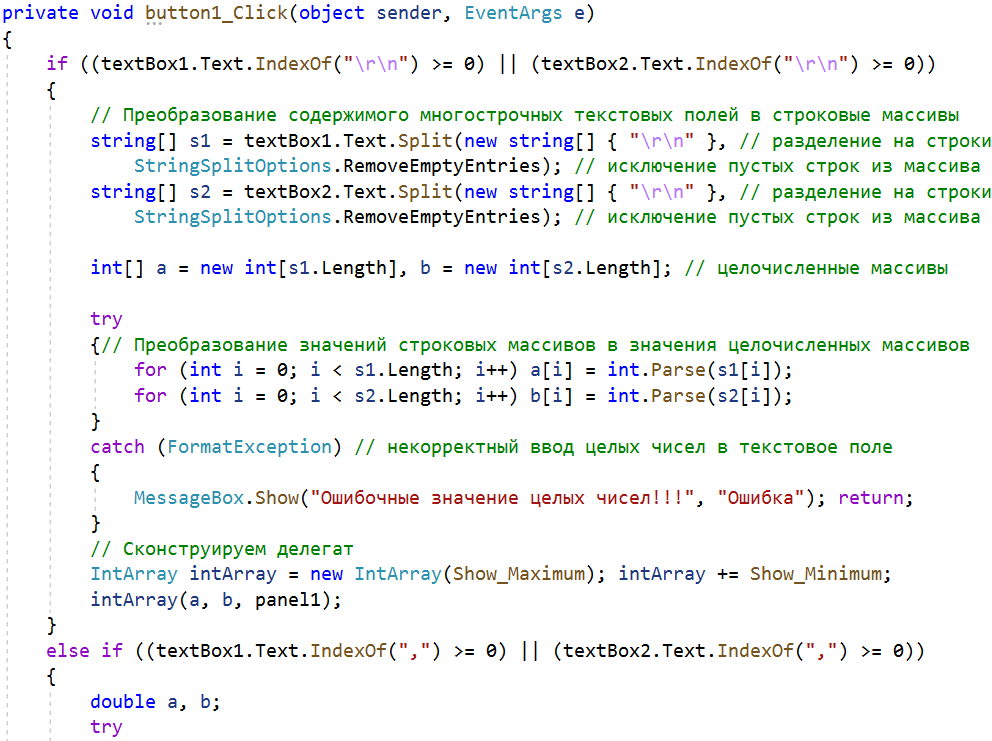


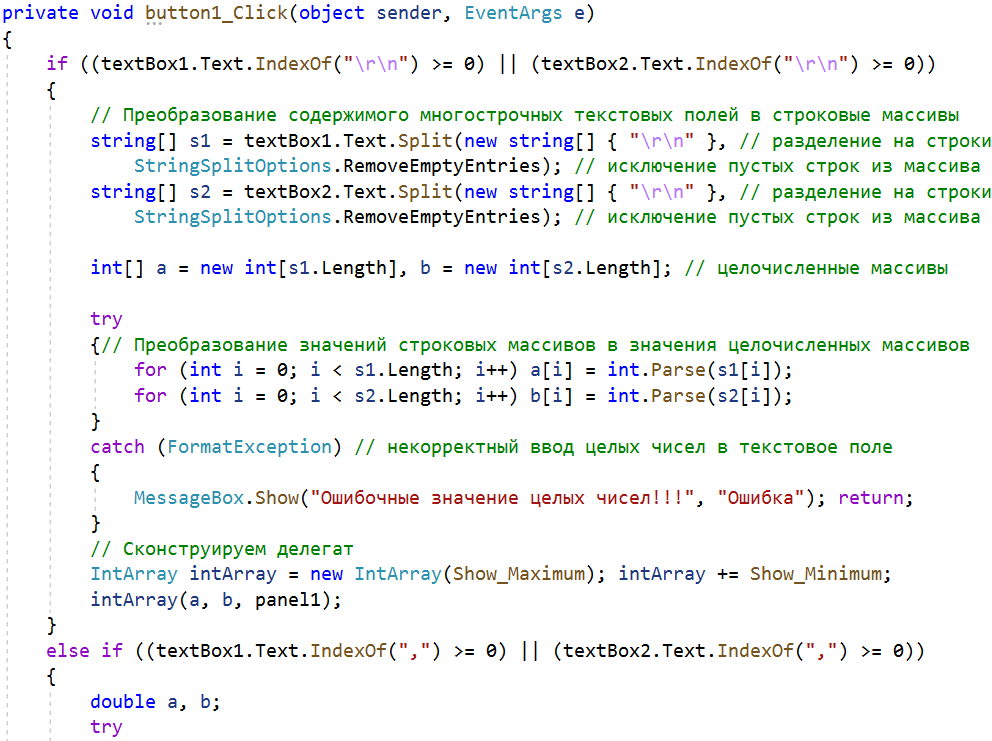
Объявление делегатов для создания и вызова списков разработанных методов при щелчке на кнопку «Вычислить»:

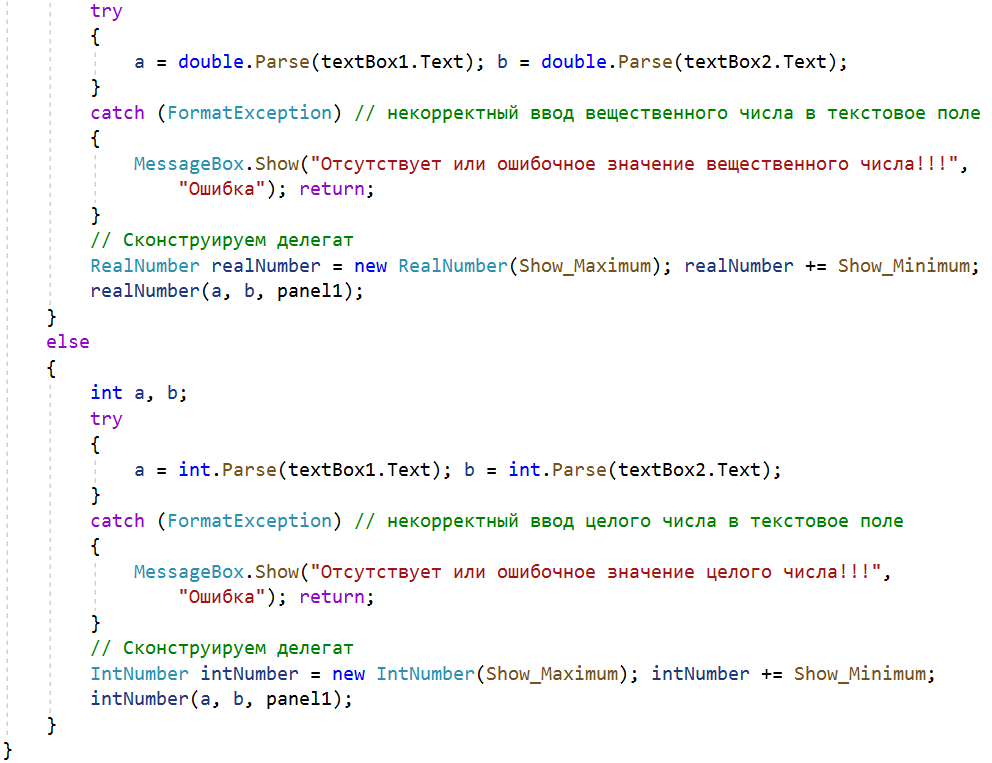


…

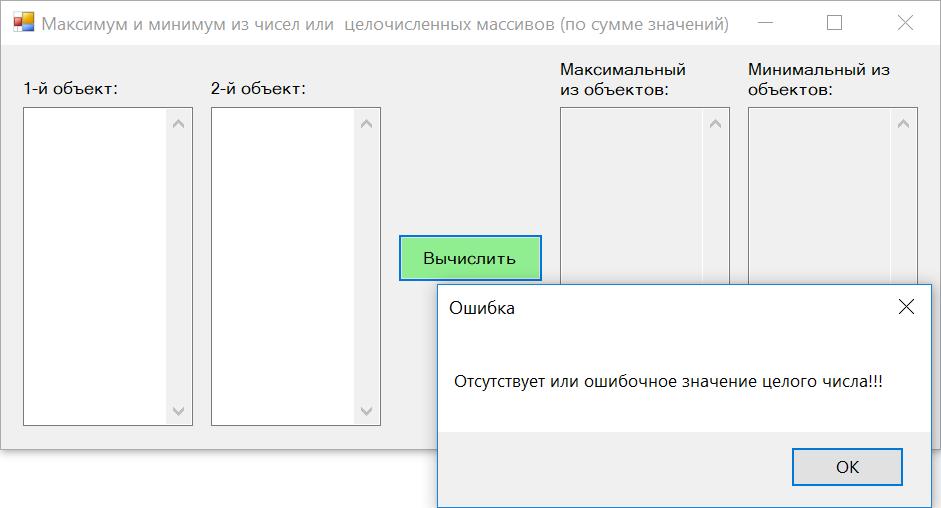
Код, выполняемый при щелчке на кнопку «Вычислить»:

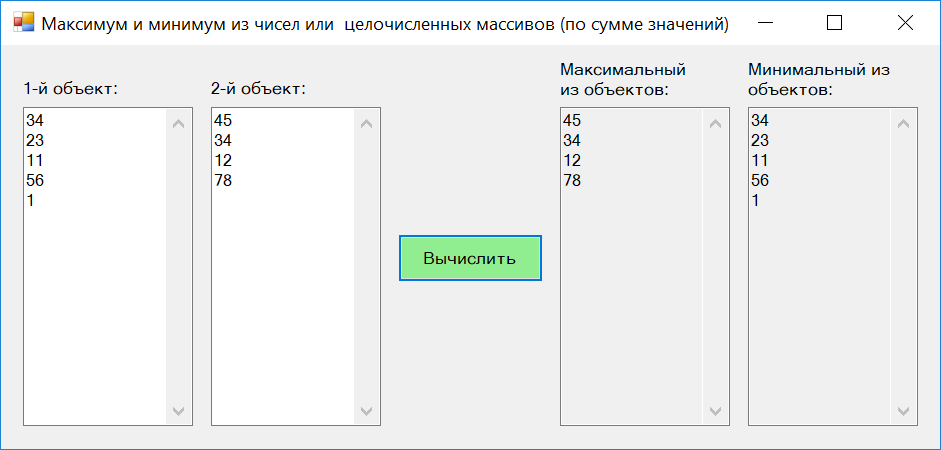


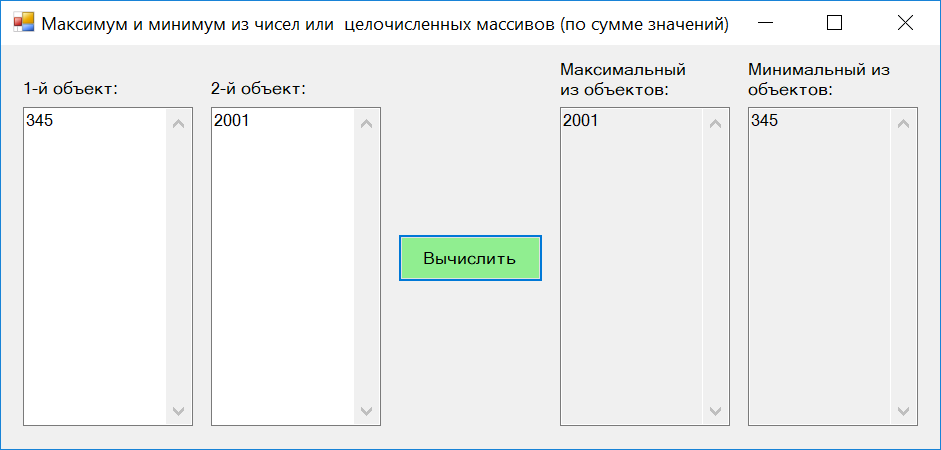


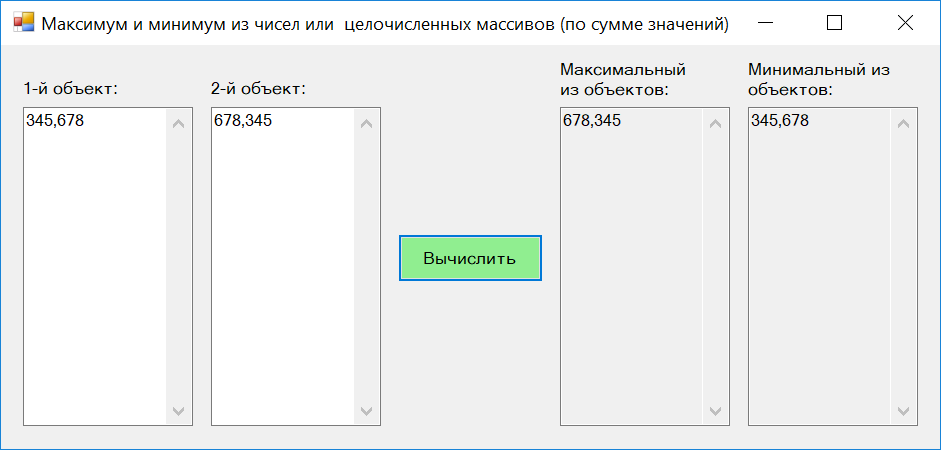


Скрины отлаженной программы:

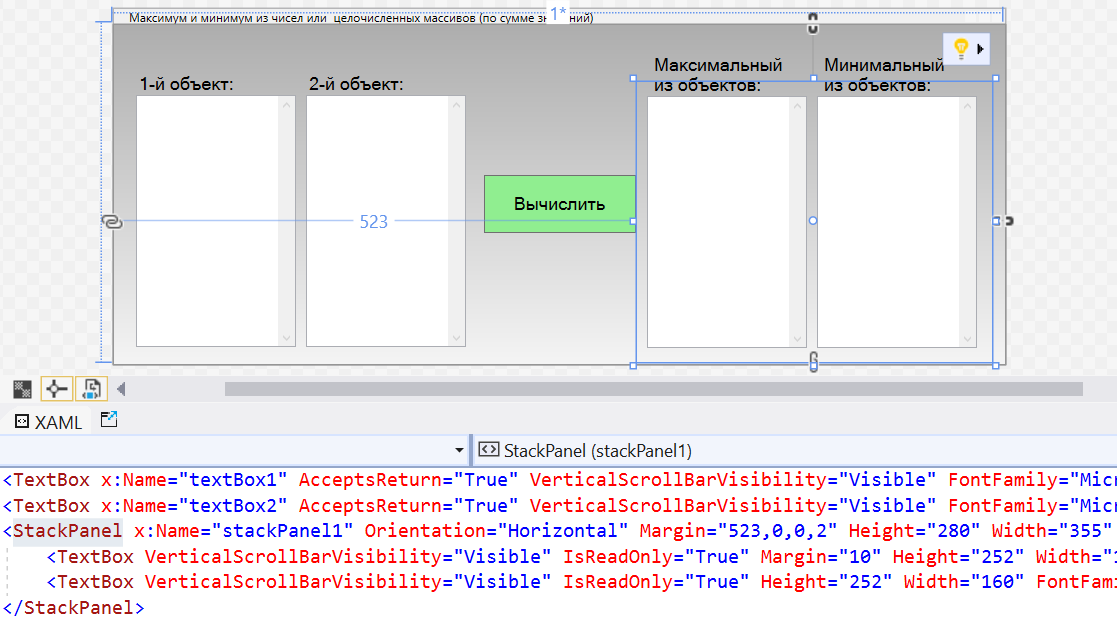




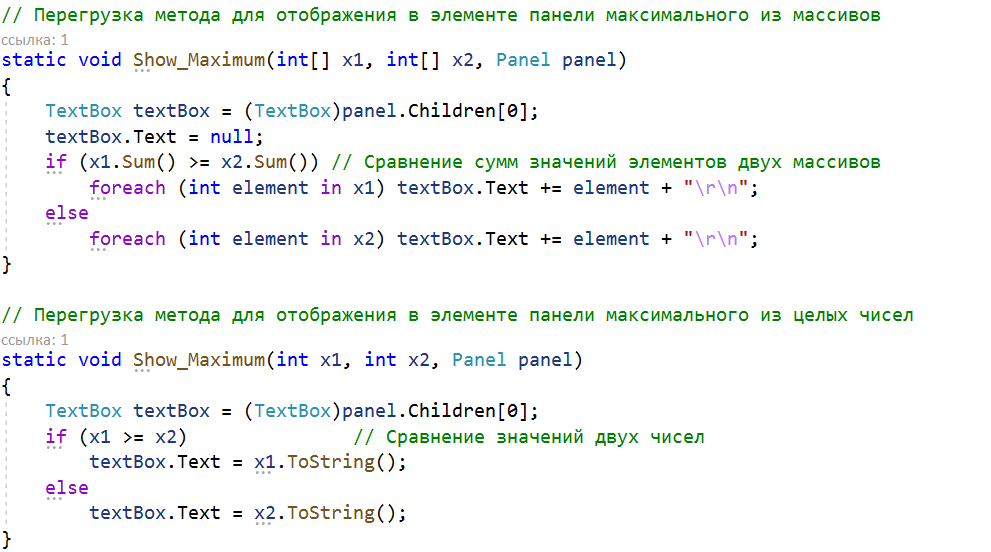


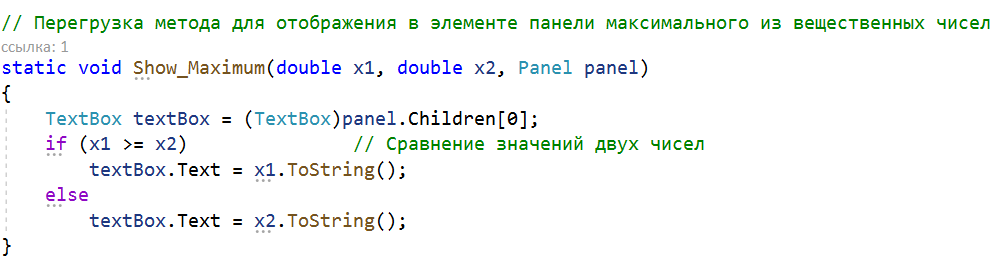


б) Интерфейс приложения WPF:

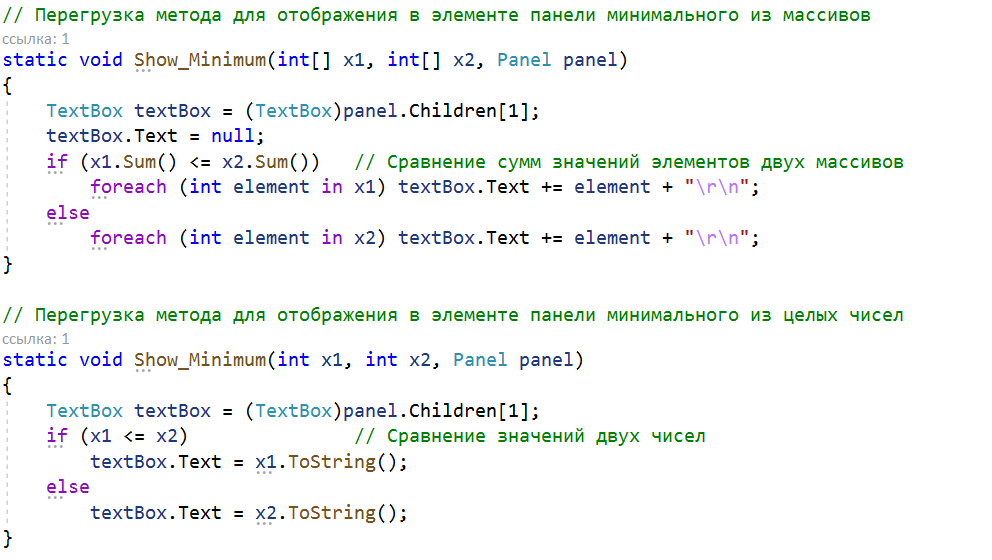


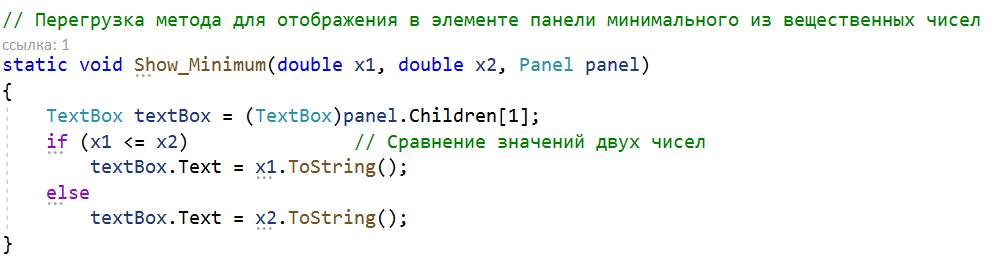
Коды перегрузок метода Show\_Maximum(x1, x2, Panel panel) для отображения в элементе, содержащемся в панели (текстовом поле), максимального объекта в каждой паре заданных объектов:



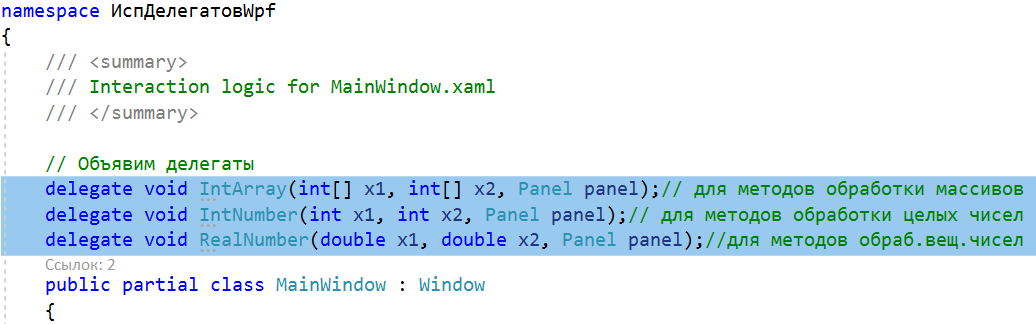


Коды перегрузок метода Show\_Minimum(x1, x2, Panel panel) для отображения в элементе, содержащемся в панели (текстовом поле), минимального объекта в каждой паре заданных объектов:



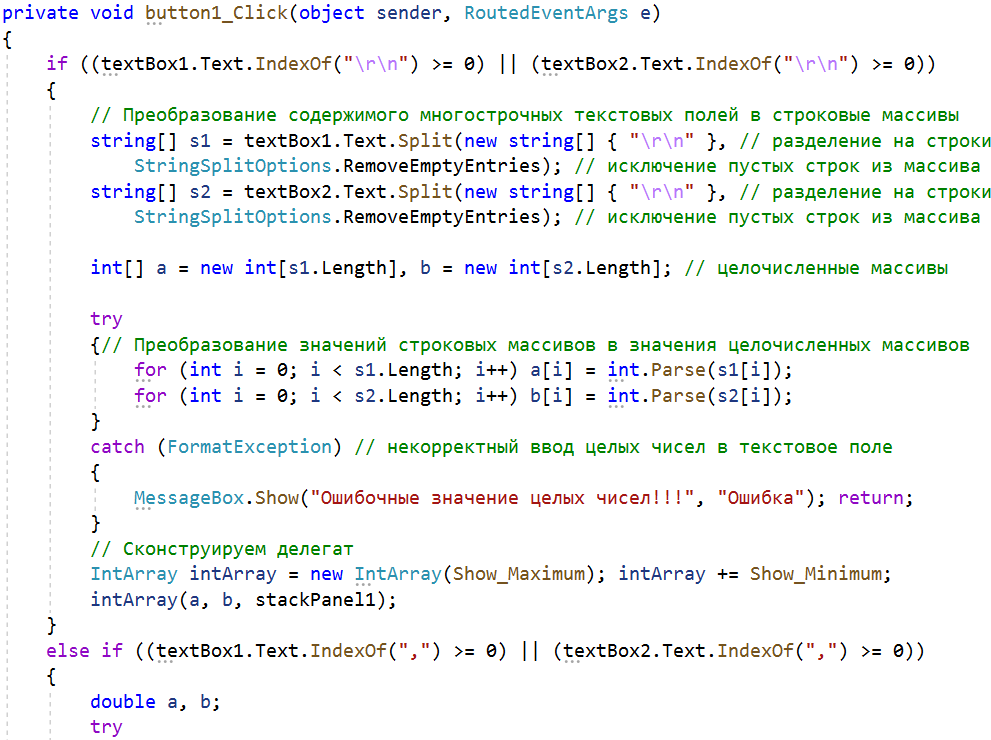


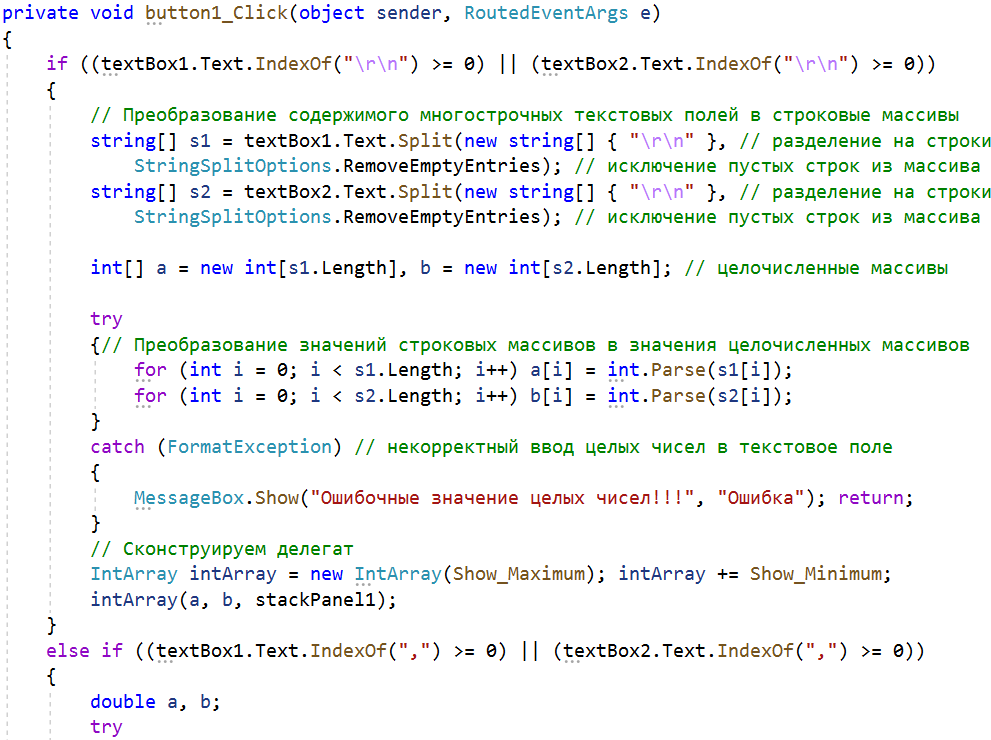
Объявление делегатов для создания и вызова списков разработанных методов при щелчке на кнопку «Вычислить»:

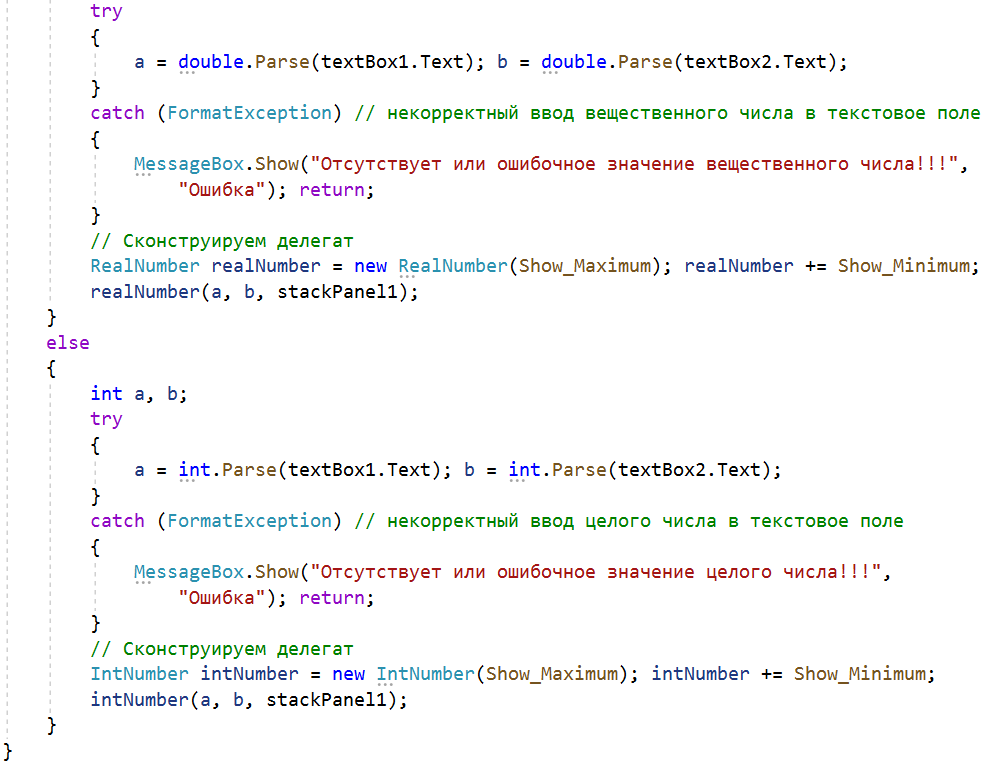


…

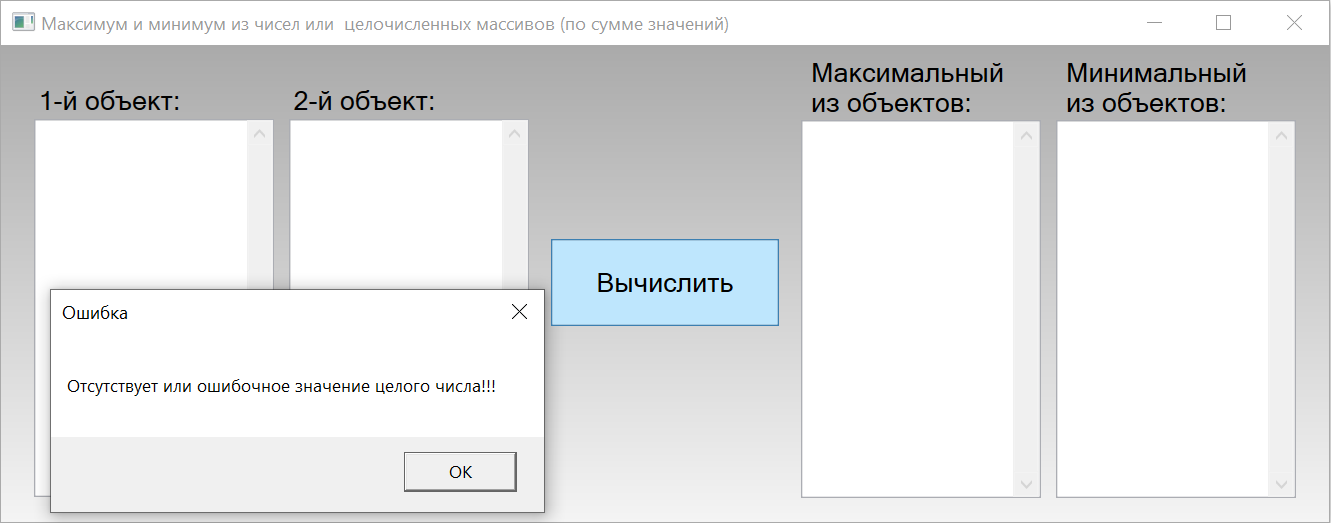
Код, выполняемый при щелчке на кнопку «Вычислить»:

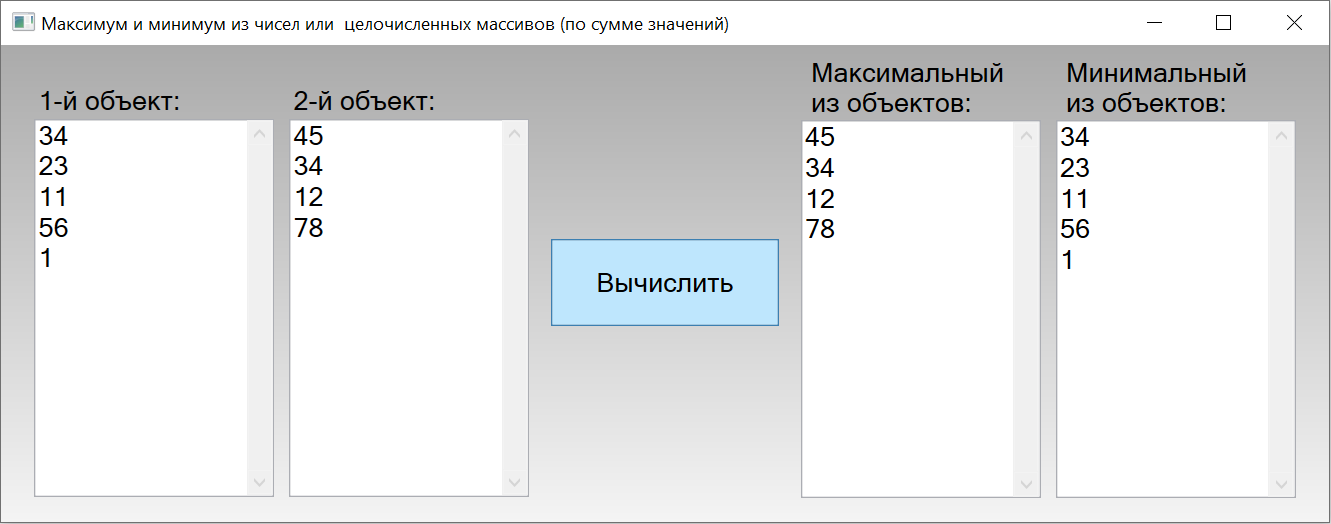


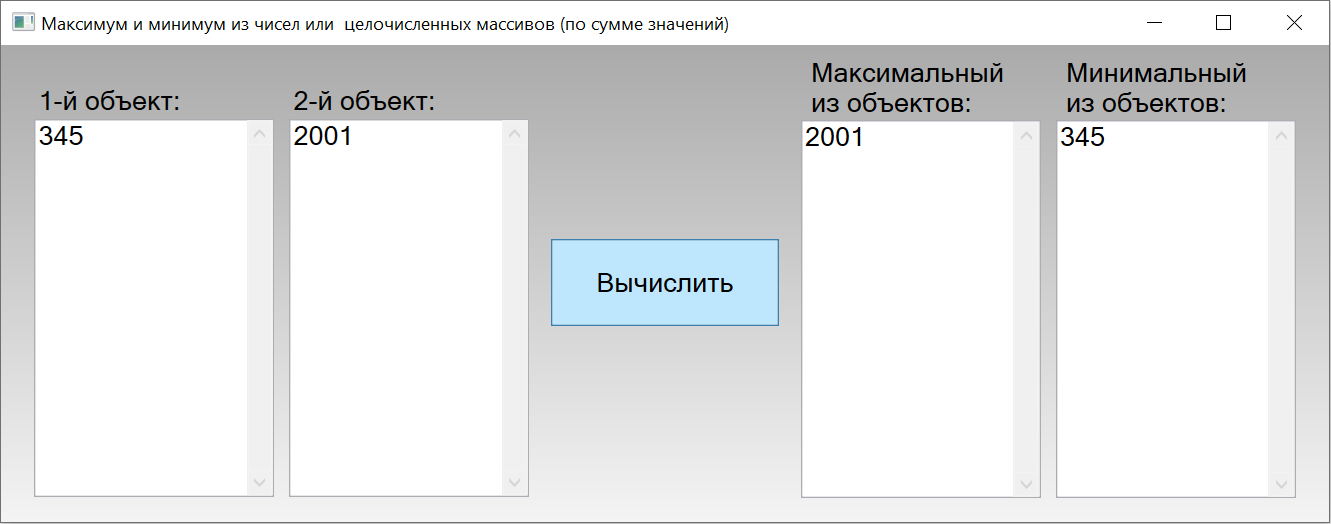


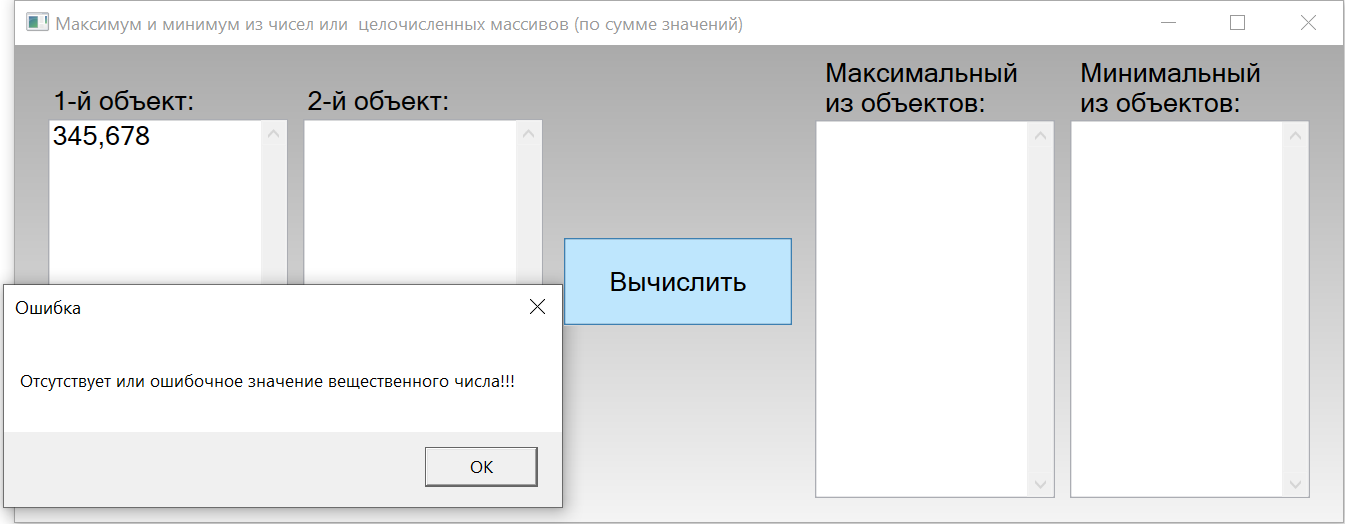


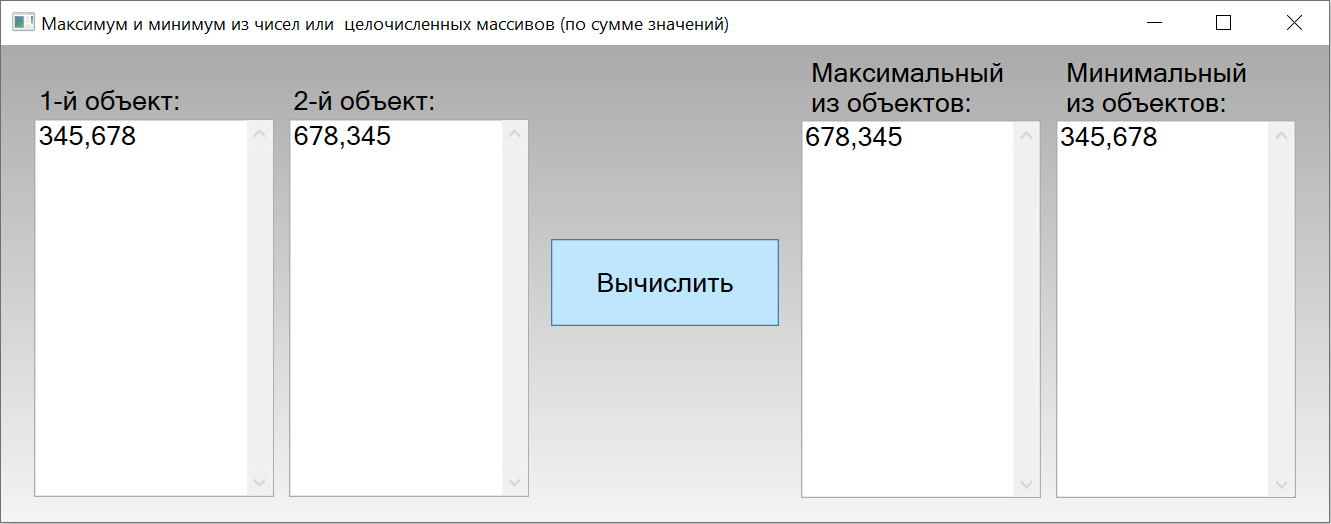
Скрины отлаженной программы:



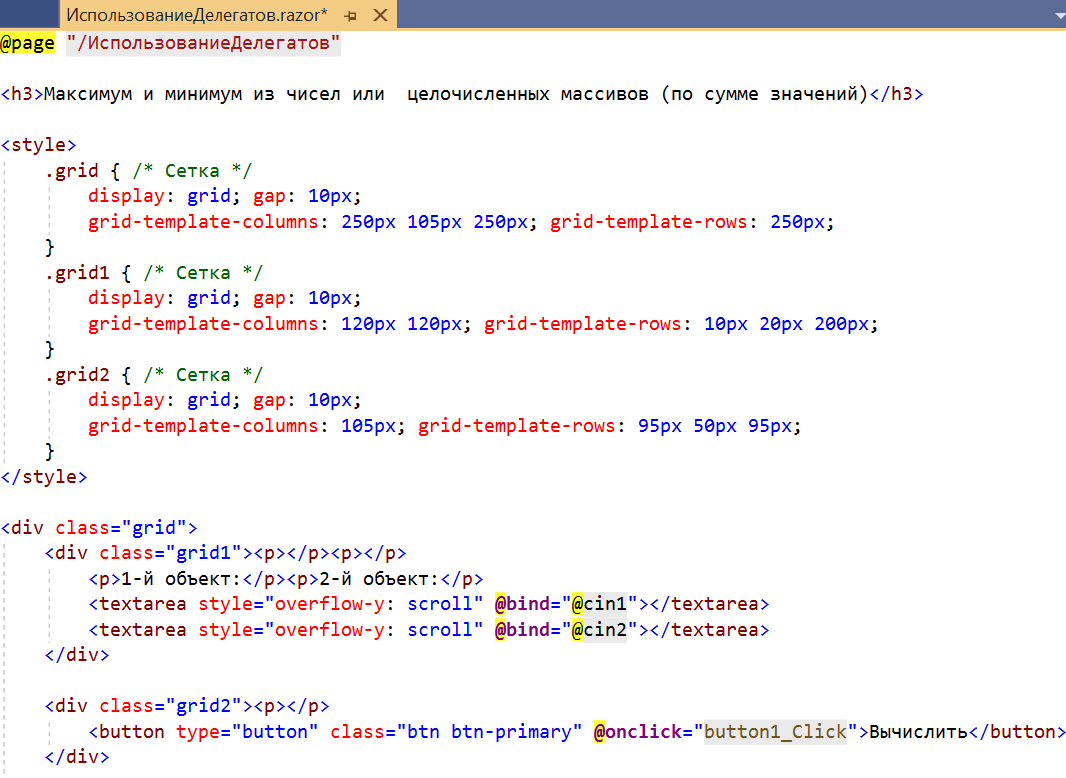




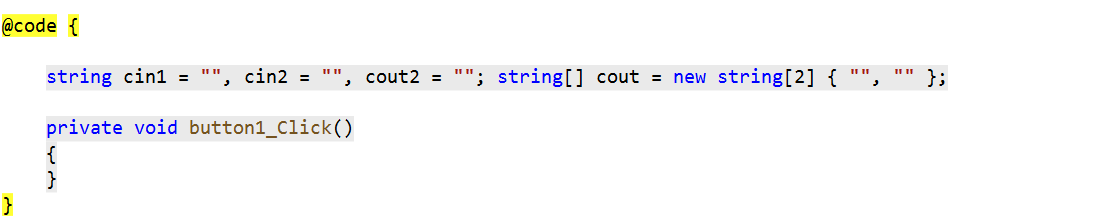


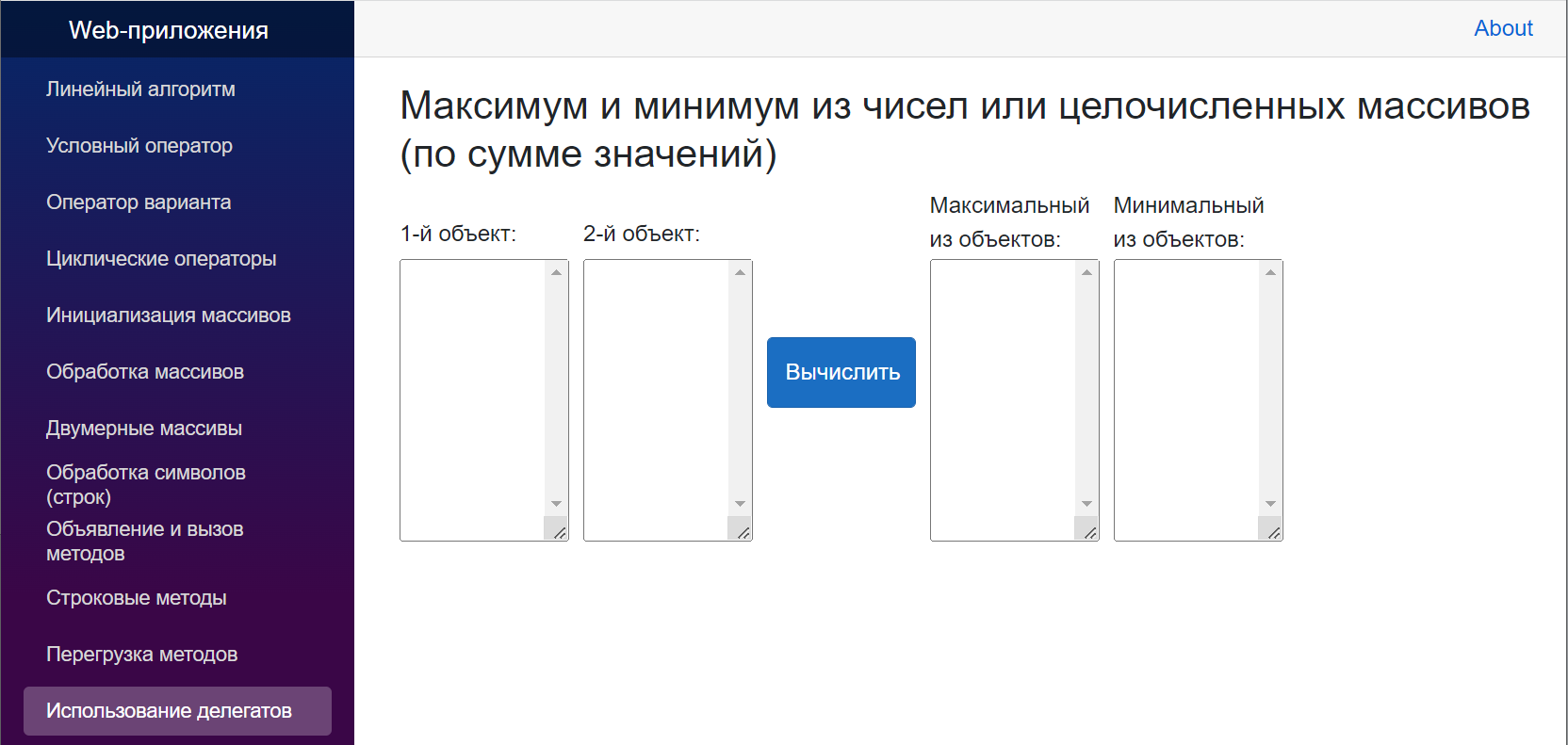


в) Интерфейс приложения WebAssembly Blazor:

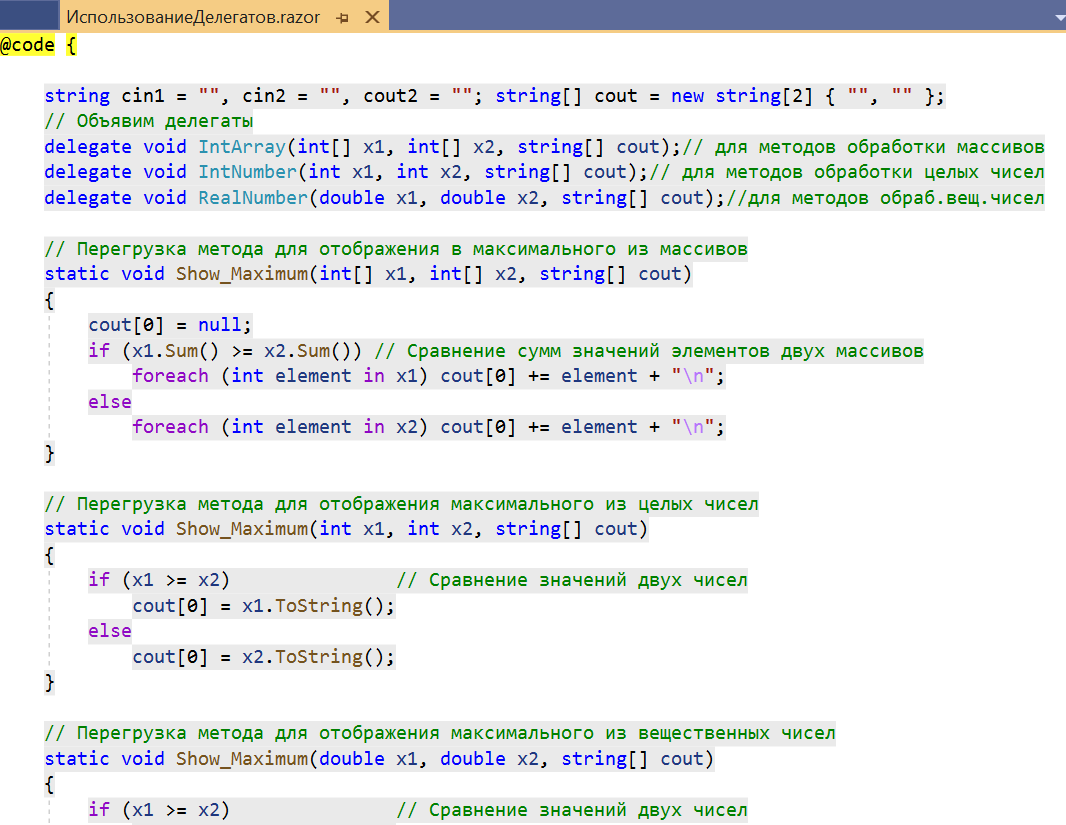


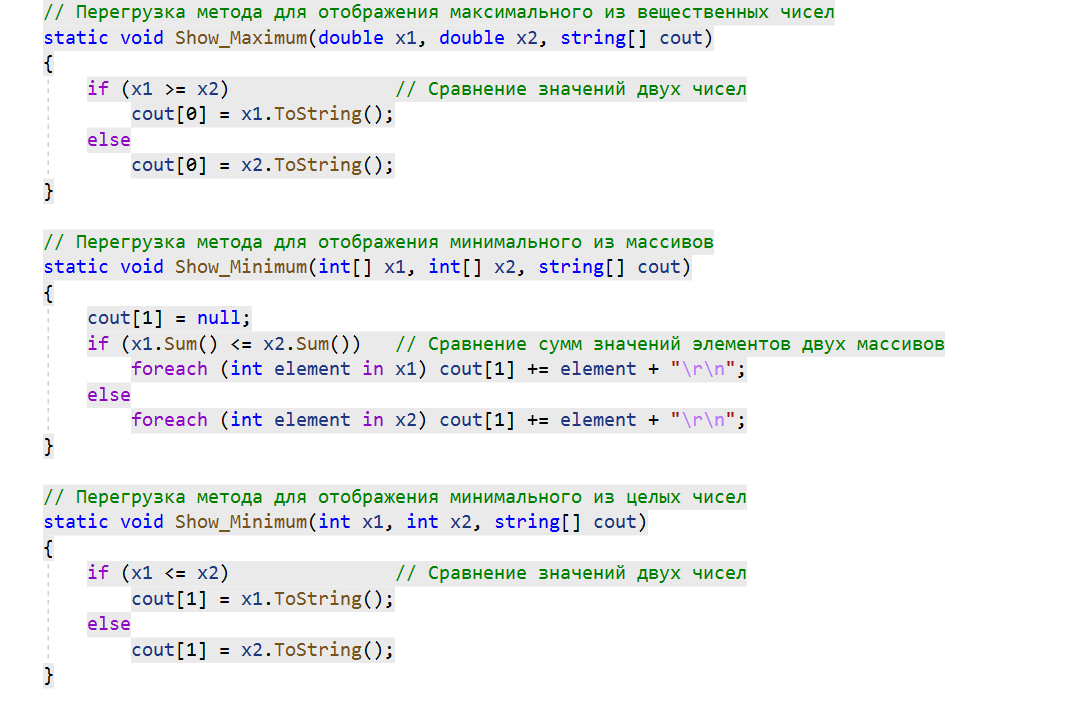


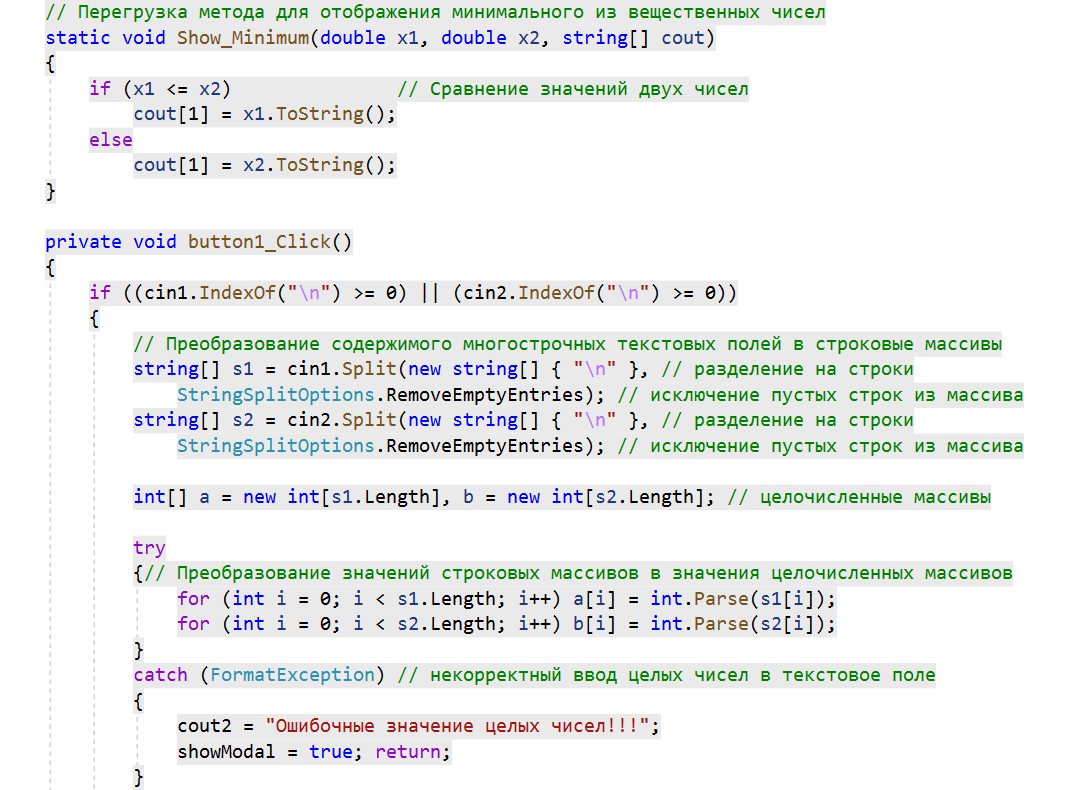


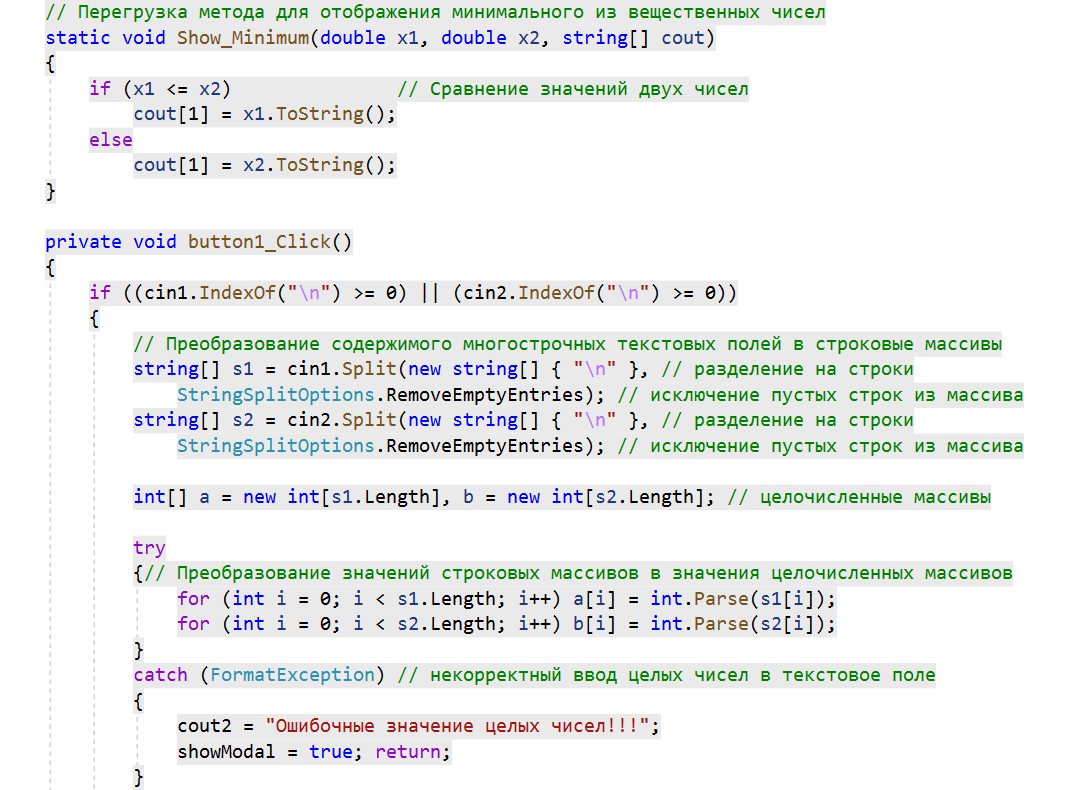


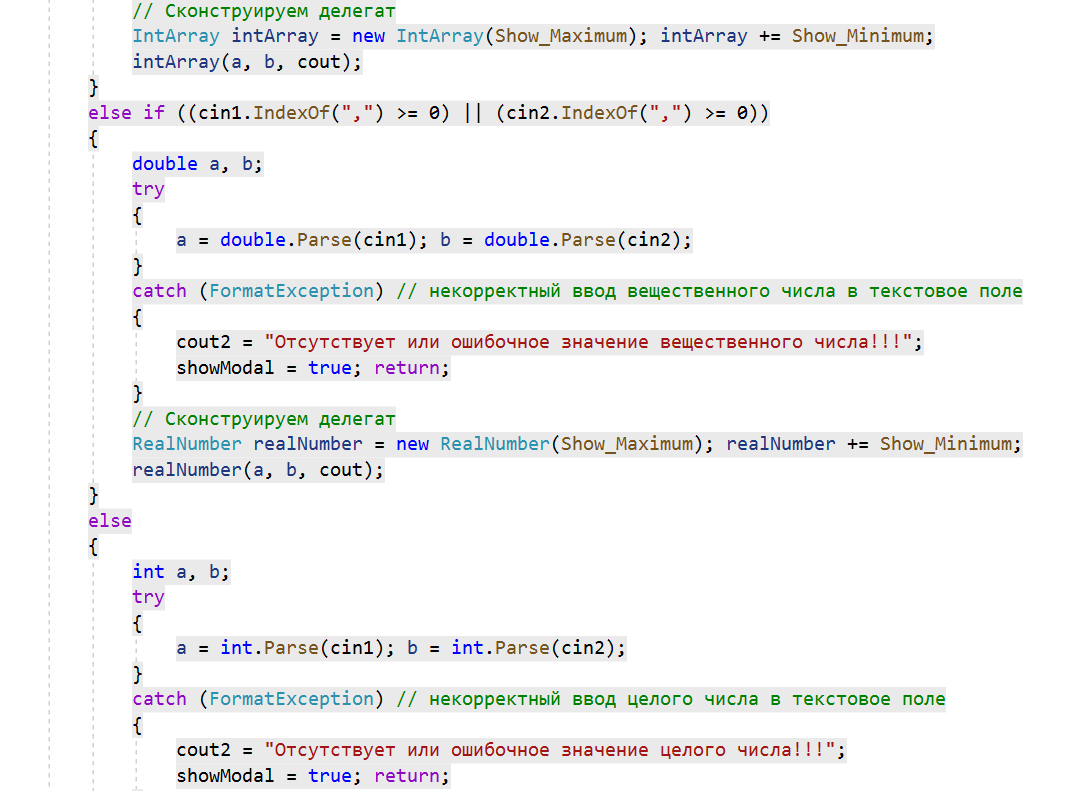
Коды приложения:

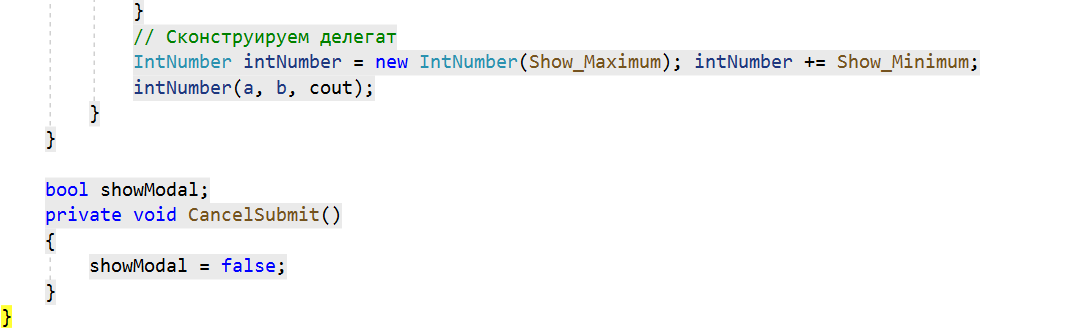


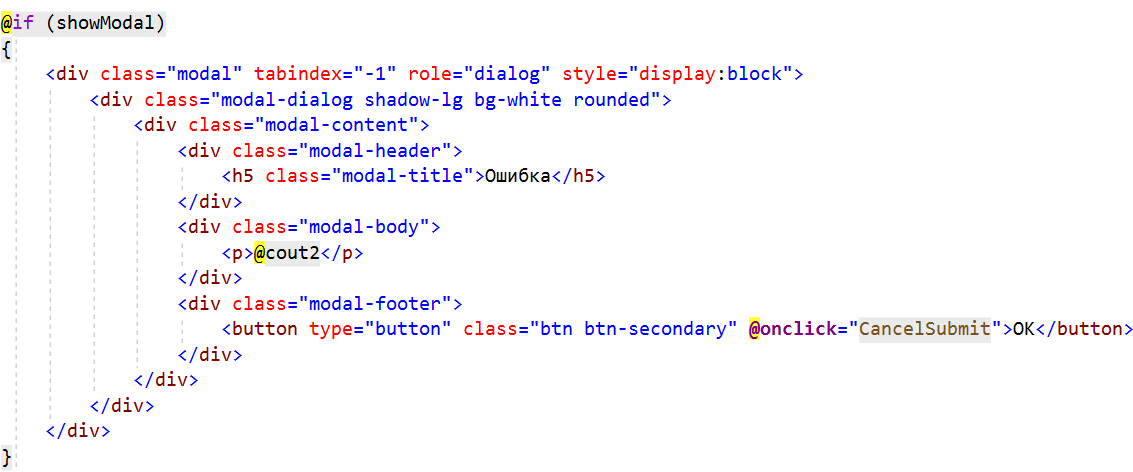




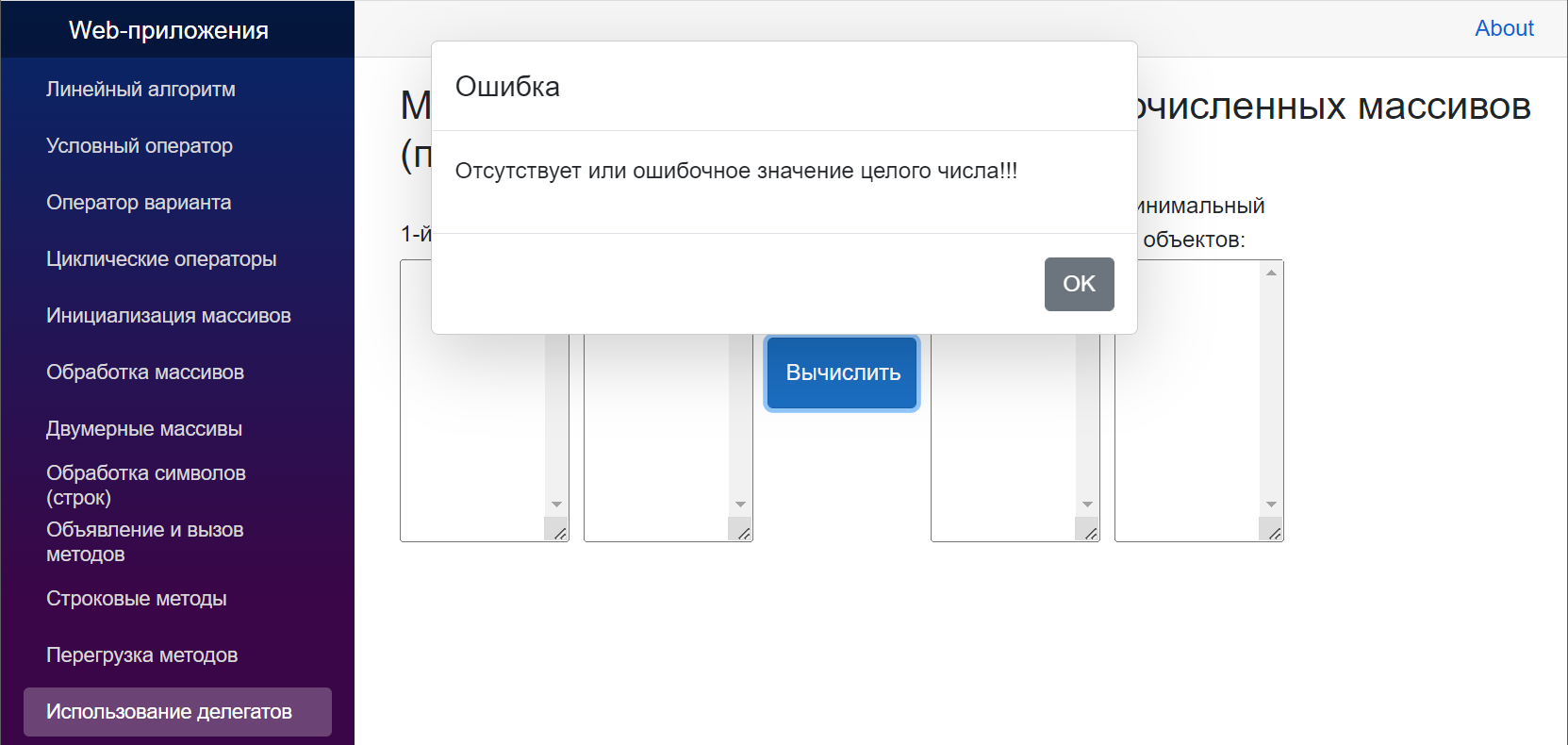


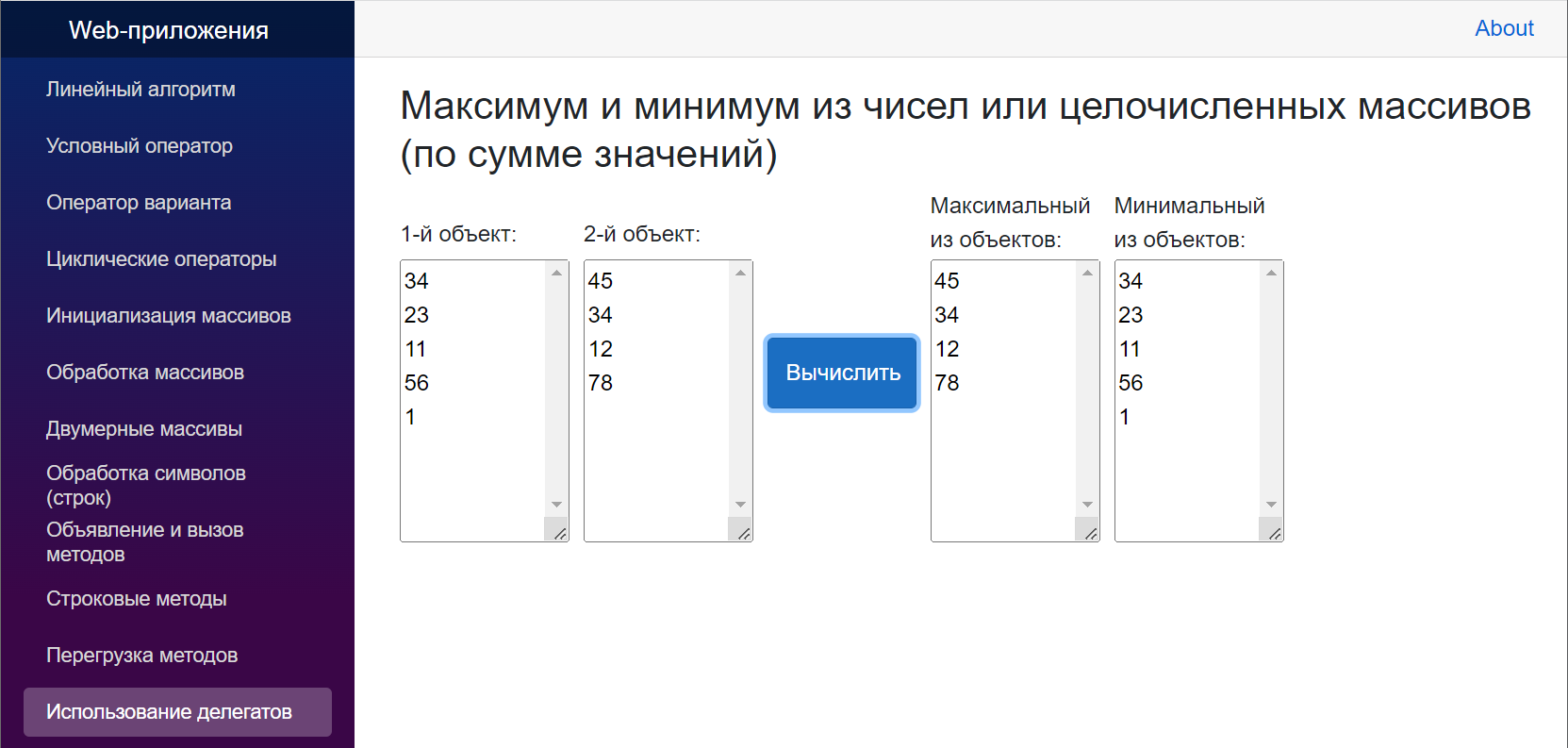


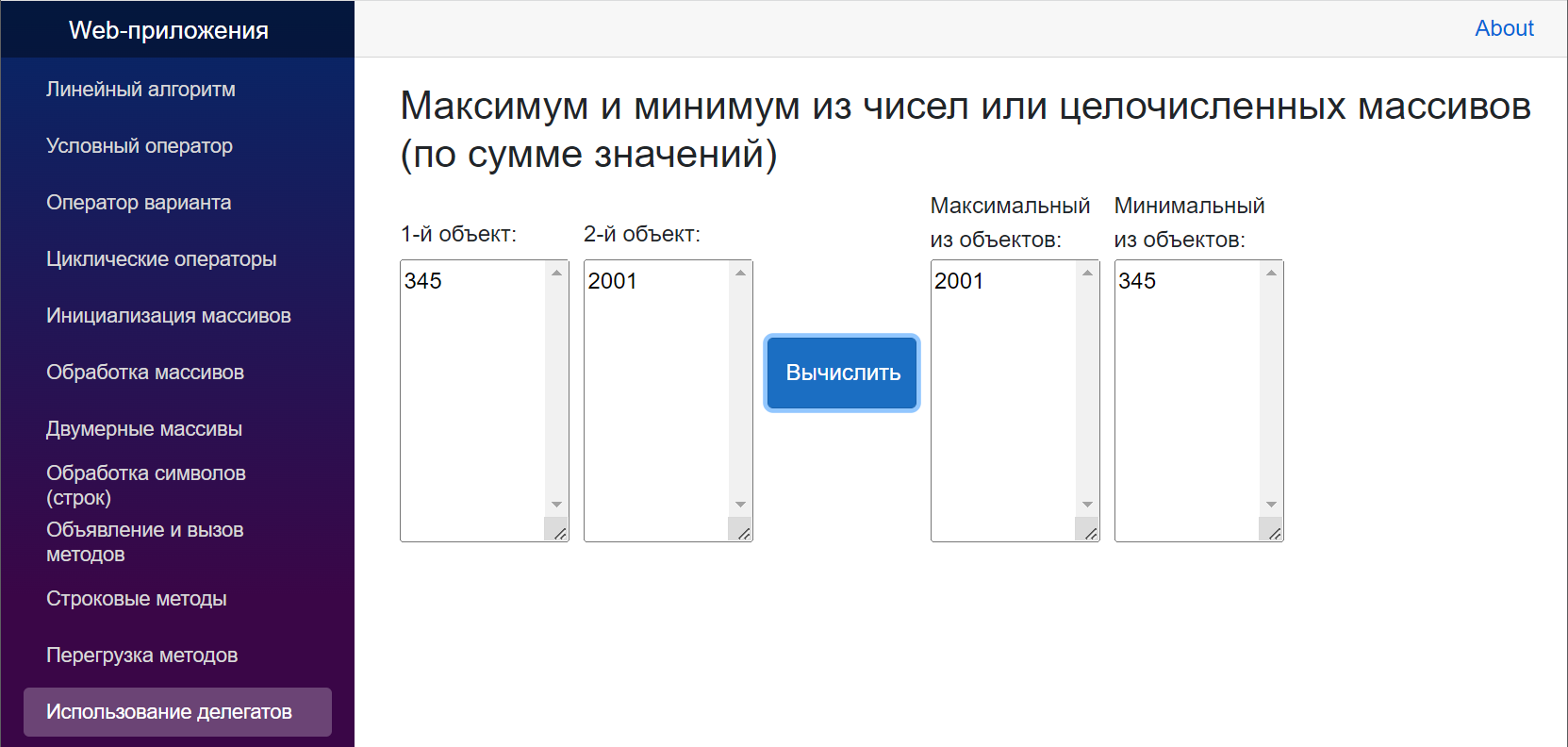


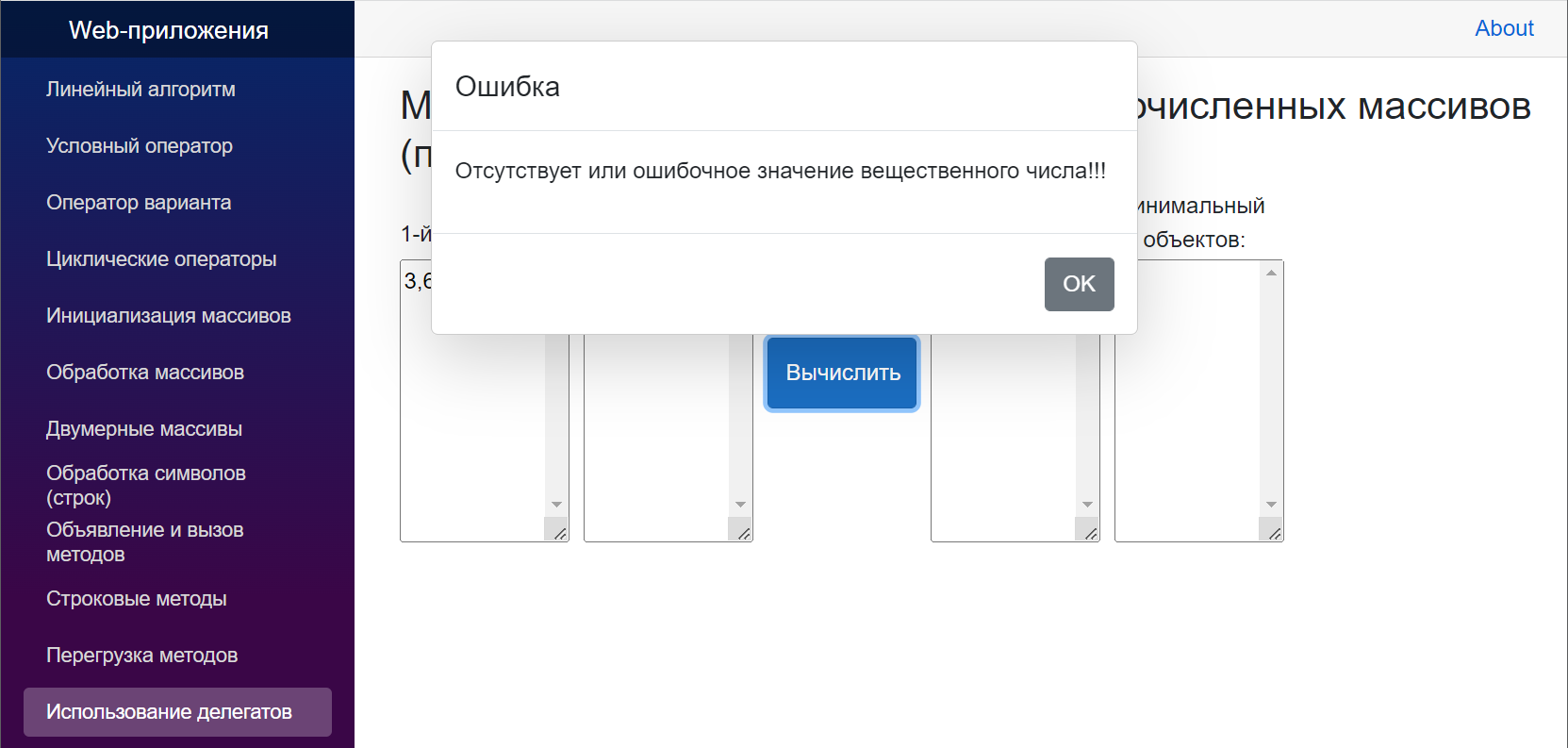


Скрины отлаженной программы:











**4 Порядок выполнения работы**

4.1 Изучить теоретические сведения и задания к работе.

4.2 В соответствии с вариантом заданий создать формы двух приложений Windows Forms (WPF, WebAssembly Blazor).

4.3 В соответствии с вариантом заданий создать перегруженные методы, различающиеся типом возвращаемых значений и типами аргументов.

4.4 В соответствии с вариантом 1-го задания разработать и отладить коды приложения Windows Forms (WPF, WebAssembly Blazor), осуществляющего обращение к созданным перегрузкам методов с использованием исключений, выбрасываемых, если значения входных данных не соответствуют спецификациям параметров обрабатывающих их методов.

4.5 В соответствии с вариантом 2-го задания разработать и отладить код приложения Windows Forms (WPF, WebAssembly Blazor), применяющего делегаты для вызовов списков разработанных методов с использованием исключений, выбрасываемых, если значения входных данных не соответствуют спецификациям параметров обрабатывающих их методов.

**5 Содержание отчета**

5.1 Название работы

5.2 Цель работы

5.3 Программные коды и скрины работы разработанных приложений Windows Forms в соответствии с вариантом заданий.

**6 Варианты заданий**

6.1.1 Даны два числа *A* и *B* (*A* < *B*). Найти заданную сумму чисел: а) сумму всех целых чисел от *A* до *B* включительно для целых значений *A* и *B*; б) сумму чисел от *A* до *B* включительно с заданным шагом *H* для вещественных значений *A* и *B*.

6.1.2 Даны два числа *A* и *B* (*A* < *B*). Отобразить (вывести) результаты двух заданных сумм чисел: а) суммы всех целых чисел от *A* до *B* включительно и суммы квадратов всех целых чисел от *A* до *B* включительно для целых значений *A* и *B*; б) суммы чисел от *A* до *B* включительно с заданным шагом *H* и суммы квадратов чисел от *A* до *B* включительно с заданным шагом *H* для вещественных значений *A* и *B*.

6.2.1 Даны два числа *A* и *B* (*A* < *B*). Найти заданное произведение чисел: а) произведение всех целых чисел от *A* до *B* включительно для целых значений *A* и *B*; б) произведение чисел от *A* до *B* включительно с заданным шагом *H* для вещественных значений *A* и *B*.

6.2.2 Даны два числа *A* и *B* (*A* < *B*). Отобразить (вывести) результаты двух заданных выражений: а) произведения всех целых чисел от *A* до *B* включительно и суммы попарных произведений всех целых чисел от *A* до *B* включительно для целых значений *A* и *B*; б) произведения чисел от *A* до *B* включительно с заданным шагом *H* и суммы попарных произведений чисел от *A* до *B* включительно с заданным шагом *H* для вещественных значений *A* и *B*.

6.3.1 Дано число *N* (> 0). Найти заданную сумму : а) для целого значения *N*; б) для вещественного значения *N*.

6.3.2 Дано число *N* (> 0). Отобразить (вывести) результаты двух заданных сумм чисел – суммы и суммы : а) для целого значения *N*; б) для вещественного значения *N*.

6.4.1 Дано натуральное число *N* и число *A*. Сформировать заданный массив размера *N* c элементами *A* + 1, *A* + 3, *A* + 5, …: а) для целого значения *A*; б) для вещественного значения *A*.

6.4.2 Дано натуральное число *N* и число *A*. Отобразить (вывести) два заданных массива размера *N* – массив с элементами *A* + 1, *A* + 3, *A* + 5, … и массив с элементами *A* + 2, *A* + 4, *A* + 6, …: а) для целого значения *A*; б) для вещественного значения *A*.

6.5.1 Дано целое число *N* (> 1), а также первый член *A* и разность *D* арифметической прогрессии. Сформировать заданный массив размера *N*, содержащий *N* первых членов данной прогрессии *A*, *A* + *D*, *A* + 2×*D*, *A* + 3×*D*, …: а) для целых значений *A* и *D*; б) для вещественных значений *A* и *D*.

6.5.2 Дано целое число *N* (> 1), а также первый член *A* и разность (знаменатель) *D* числовой прогрессии. Отобразить (вывести) два заданных массива размера *N* – массив, содержащий *N* первых членов арифметической прогрессии *A*, *A* + *D*, *A* + 2×*D*, *A* + 3×*D*, … и массив, содержащий *N* первых членов геометрической прогрессии *A*, *A* × *D*, *A* × *D*2, *A* × *D*3, …: а) для целых значений *A* и *D*; б) для вещественных значений *A* и *D*.

6.6.1 Дано натуральное число *N*. Найти минимальную площадь для прямоугольников, входящих в набор из заданных *N* прямоугольников. Стороны прямоугольников могут быть заданы: а) парами целых чисел *a* и *b*; б) парами вещественных чисел *a* и *b*.

6.6.2 Дано натуральное число *N*. Отобразить (вывести) минимальную площадь и максимальный периметр для прямоугольников, входящих в набор из заданных *N* прямоугольников. Стороны прямоугольников могут быть заданы: а) парами целых чисел *a* и *b*; б) парами вещественных чисел *a* и *b*.

6.7.1 Задан один из месяцев года (1 – январь, 2 – февраль, и т. д.). Найти название соответствующего времени года («зима», «весна» и т. д.). Месяц может быть задан: а) номером (целым числом); б) названием (строковым значением).

6.7.2 Задан один из месяцев года (1 – январь, 2 – февраль, и т. д.). Отобразить (вывести) название соответствующего времени года («зима», «весна» и т. д.) и число дней в этом месяце для не високосного года (т. е. в феврале 28 дней). Месяц может быть задан: а) номером (целым числом); б) названием (строковым значением).

6.8.1 Дано натуральное число *n* и число *a*. Найти заданное произведение : а) для целого значения *a*; б) для вещественного значения *a*.

6.8.2 Дано натуральное число *n* и число *a*. Отобразить (вывести) результаты двух заданных произведений чисел – произведения и произведения : а) для целого значения *n*; б) для вещественного значения *n*.

6.9.1 Дано натуральное число *N* и число *A*. Найти заданную сумму первых *N* слагаемых : а) для целого значения *A*; б) для вещественного значения *A*.

6.9.2 Дано натуральное число *N* и число *A*. Отобразить (вывести) результаты двух заданных выражений – суммы первых *N* слагаемых и произведения первых *N* сомножителей : а) для целого значения *A*; б) для вещественного значения *A*.

6.10.1 Дано натуральное число *n* и число *x*. Найти заданную сумму : а) для целого значения *x*; б) для вещественного значения *x*.

6.10.2 Дано натуральное число *n* и число *x*. Отобразить (вывести) результаты двух заданных сумм чисел – суммы и суммы а) для целого значения *x*; б) для вещественного значения *x*.

6.11.1 Найти сумму каждой пары заданных объектов: а) двух одномерных целочисленных массивов (поэлементно); б) двух целых чисел.

6.11.2 Отобразить (вывести) результаты суммы и разности каждой пары заданных объектов: а) двух одномерных целочисленных массивов (поэлементно); б) двух целых чисел.

6.12.1 Найти произведение каждой пары заданных объектов: а) двух одномерных целочисленных массивов (поэлементно); б) двух целых чисел.

6.12.2 Отобразить (вывести) результаты произведения и частного каждой пары заданных объектов: а) двух одномерных целочисленных массивов (поэлементно); б) двух целых чисел.

6.13.1 Найти объединение (логическая операция ИЛИ) каждой пары заданных объектов: а) двух одномерных целочисленных массивов; б) двух логических значений.

6.13.2 Отобразить (вывести) результаты объединения (логическая операция ИЛИ) и пересечения (операция И) каждой пары заданных объектов: а) двух одномерных целочисленных массивов; б) двух логических значений.

6.14.1 Найти сумму каждой пары заданных объектов: а) двух векторов, заданных проекциями на оси *X* и *Y*; б) двух вещественных чисел.

6.14.2 Отобразить (вывести) результаты суммы и разности каждой пары заданных объектов: а) двух векторов, заданных проекциями на оси *X* и *Y*; б) двух вещественных чисел.

6.15.1 Найти соединение (конкатенацию) каждой пары заданных объектов: а) двух одномерных целочисленных массивов; б) двух строк.

6.15.2 Отобразить (вывести) результаты соединения (конкатенации) и вычитания (разности) каждой пары заданных объектов: а) двух одномерных целочисленных массивов; б) двух строк.

6.16.1 Найти инверсию каждого заданного объекта: а) целого числа (обратное число); б) строки (инвертированная строка).

6.16.2 Отобразить (вывести) результаты инверсии и удваивания каждого заданного объекта: а) целого числа; б) строки.

6.17.1 Найти площадь объединения каждой пары заданных объектов: а) двух прямоугольников, заданных координатами своих противоположных вершин; б) двух кругов, заданных координатами центральной точки и радиусом.

6.17.2 Отобразить (вывести) результаты площади объединения и площади пересечения каждой пары заданных объектов: а) двух прямоугольников, заданных координатами своих противоположных вершин; б) двух кругов, заданных координатами центральной точки и радиусом.

**7 Используемая литература**

7.1 Г. Гагарина, Е. В. Кокорева, Б. Д. Виснадул Технология разработки программного обеспечения. Форум, Инфра-М, 2009

7.2 Эндрю Троелсен Язык программирования С# 2010 и платформа .NET 4.0, М., ВИЛЬЯМС, 2011

7.3 А. Марченко C#. Введение в программирование, М, Вильямс, 2009

7.4 http://msdn.microsoft.com/ru-ru/library/67ef8sbd.aspx.