# Практическое занятие № 3. Программирование разветвляющихся алгоритмов. Условный оператор

**1 Цель занятия**

Получить практические навыки разработки приложений Windows Forms с разветвляющимися алгоритмами.

**2 Перечень оборудования и программного обеспечения**

Персональный компьютер

Microsoft Office (Word)

Microsoft Visual Studio

**3 Краткие теоретические сведения**

**3.1 Оператор условия if**

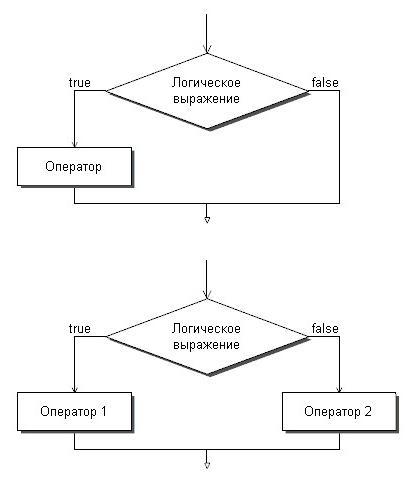
Условные операторы позволяют осуществить ветвление алгоритма и делают возможность выбрать для выполнения один из операторов.

Синтаксис оператора if можно представить следующим образом:

if (<лог\_выражение>) <оператор>; //сокращенная форма

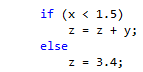
if (<лог\_выражение>) <оператор1> else <оператор2>; //полная форма

Блок-схема оператора if (показаны сокращённая и полная формы):

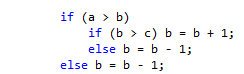


В выражении должен получаться результат, имеющий логический тип. Если результатом выражения является истинное значение true, то выполняется оператор, следующий за условием в скобках. Если результатом выражения является значение false и присутствует ключевое слово else, то выполнятся оператор, следующий за ключевым словом else. Если ключевое слово else отсутствует, то никакой оператор не выполняется. В предшествующем else операторе точка с запятой указывается. В общем случае ключевое слово else связывается с ближайшим ключевым словом if, которое еще не связано с ключевым словом else. Если вместо указанных <оператор1> и <оператор2> требуется выполнить несколько операторов, используются операторные скобки {}.

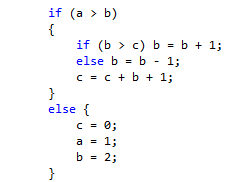
**Пример 1:**



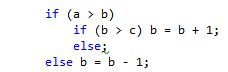
**Пример 3:**



**Пример 2:**



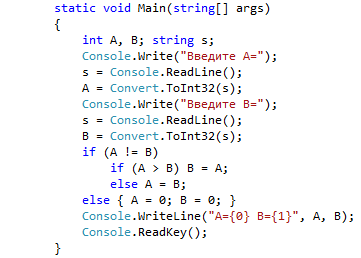
**Пример 4:**



**Пример 5:**



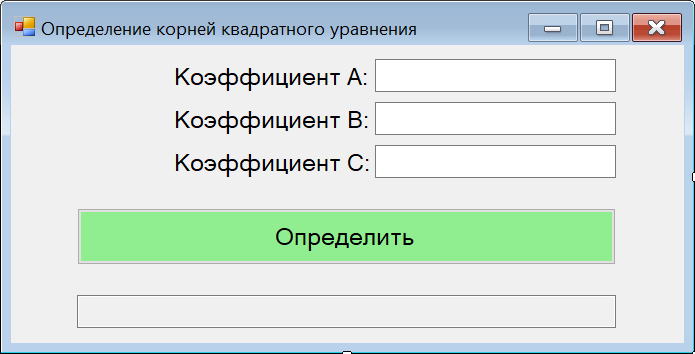
**Пример 6.** Даны две переменные целого типа *A* и *B*. Если их значения не равны, то присвоить каждой переменной большее из этих значений, а если равны, то присвоить переменным нулевые значения. Вывести новые значения переменных *A* и *B*:



**Задание.** Даны *a*, *b*, *c* - коэффициенты квадратного уравнения. Получить корни уравнения.

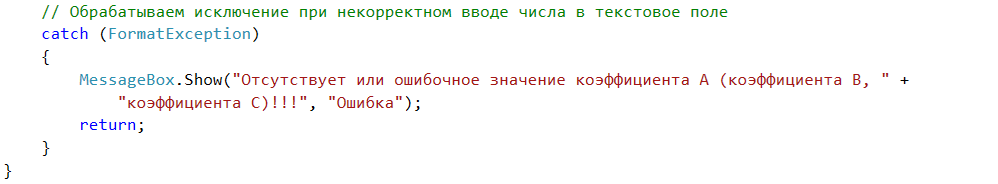
**Решение.**

а) Интерфейс приложения Windows Forms:

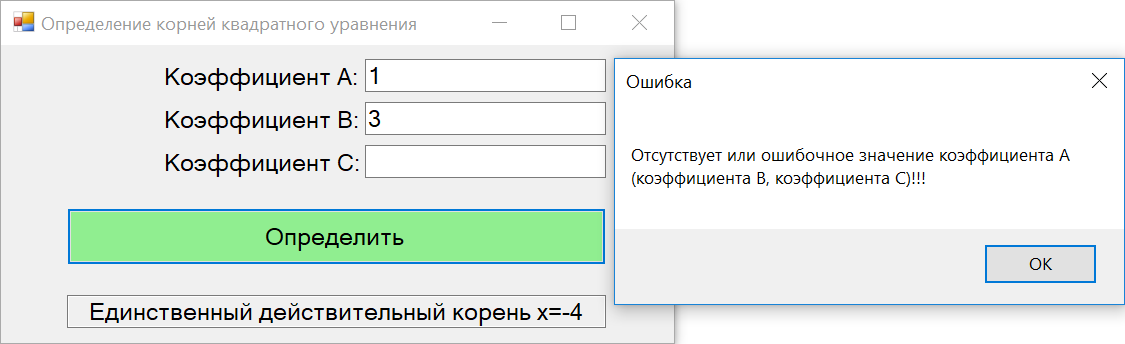


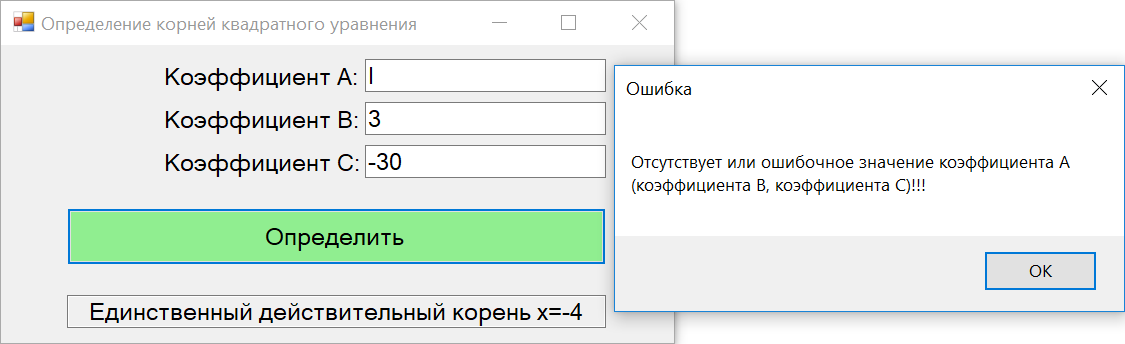
Метод button1\_Click(), содержащий код приложения:

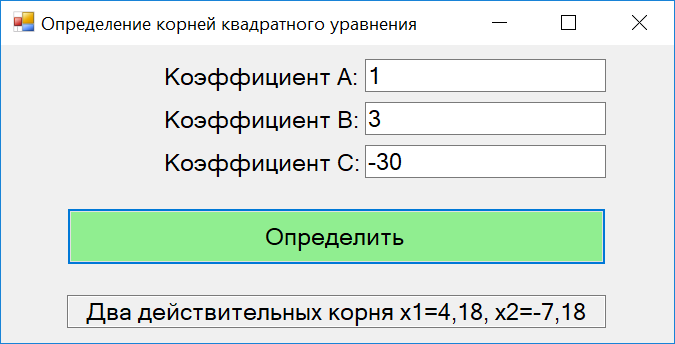


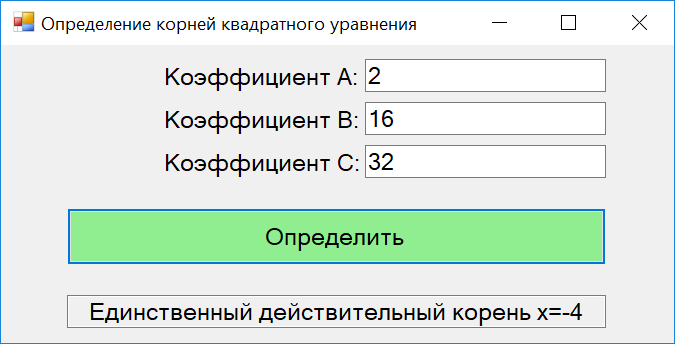


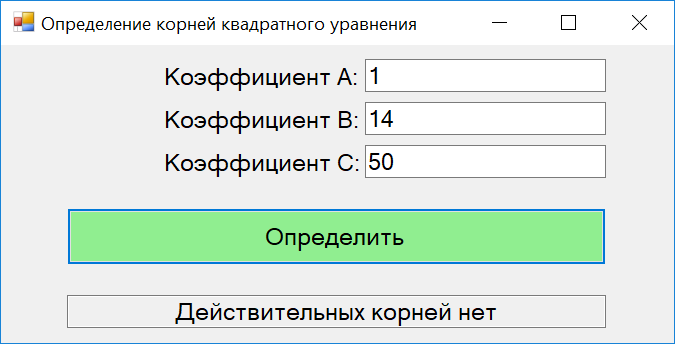
Скрины отлаженной программы:



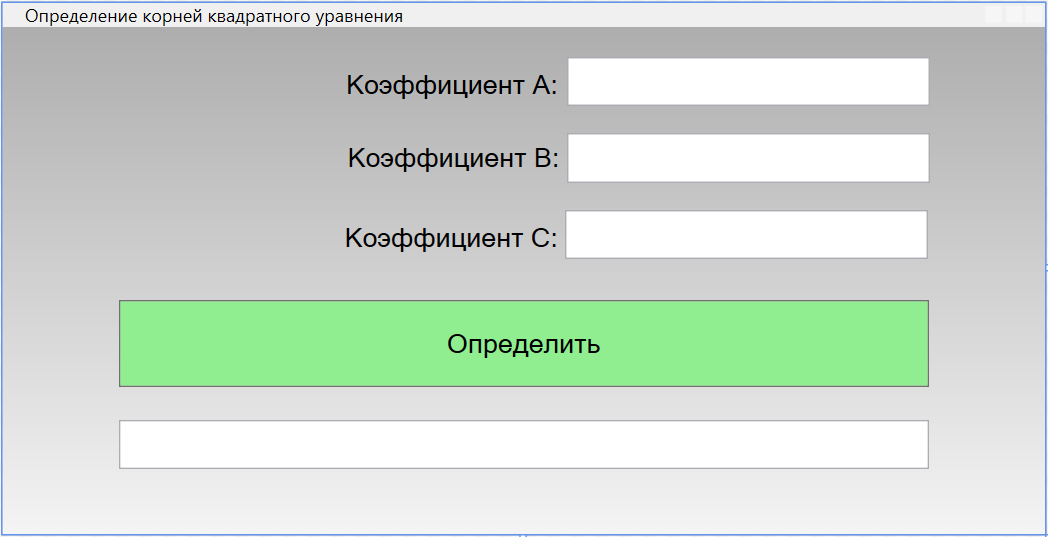




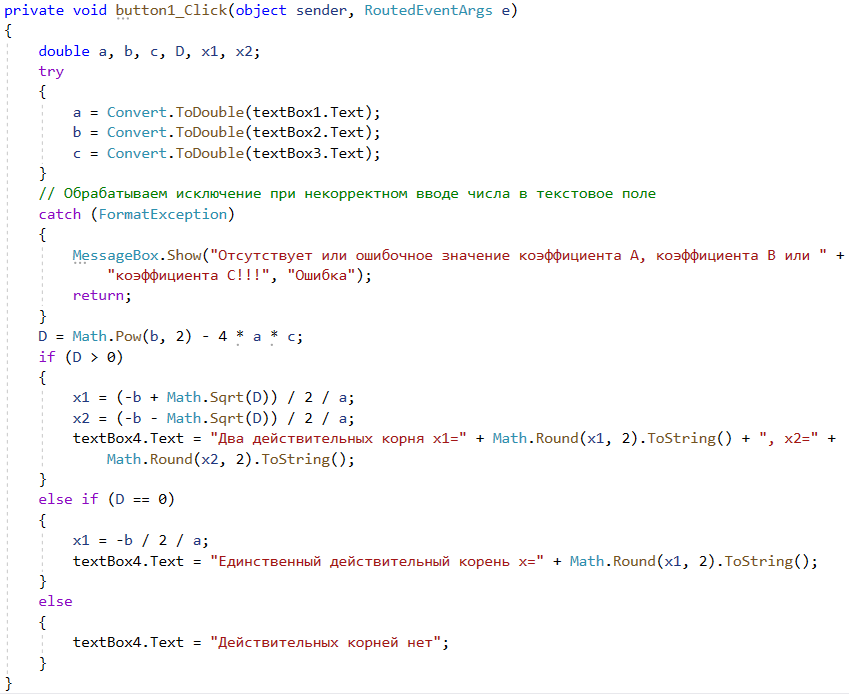




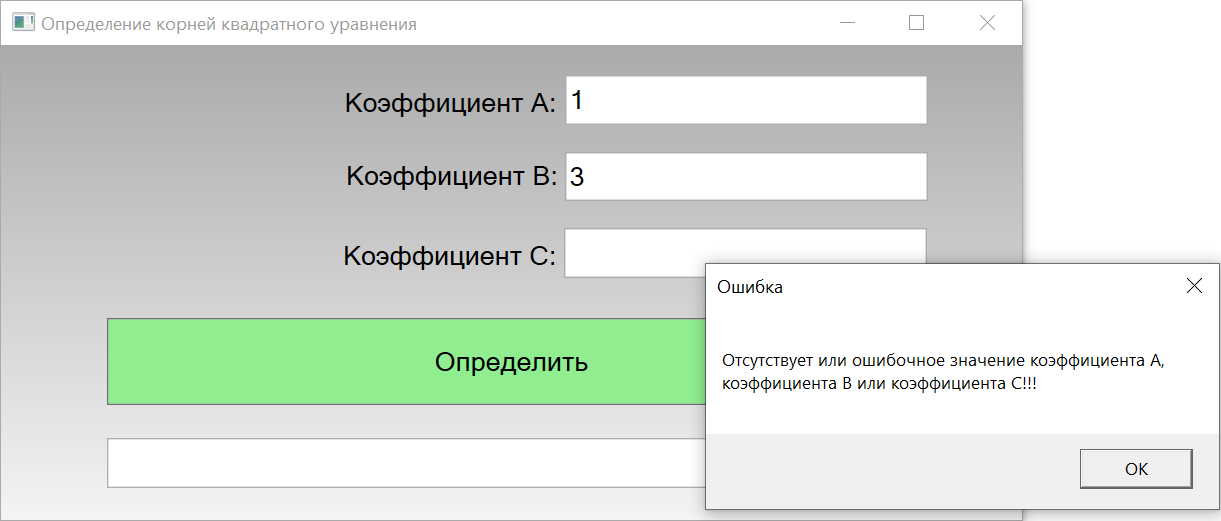
б) Интерфейс приложения WPF:

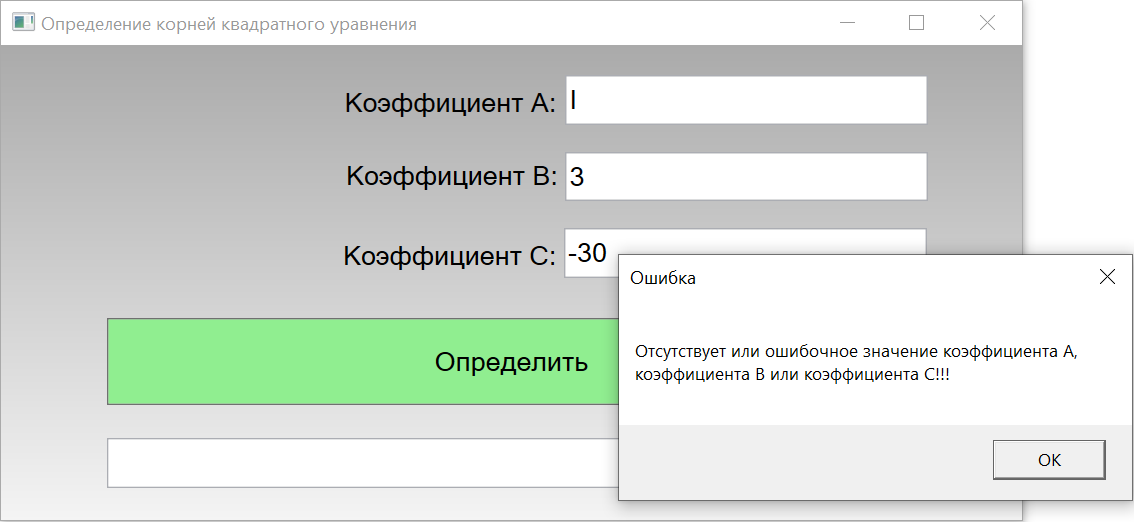


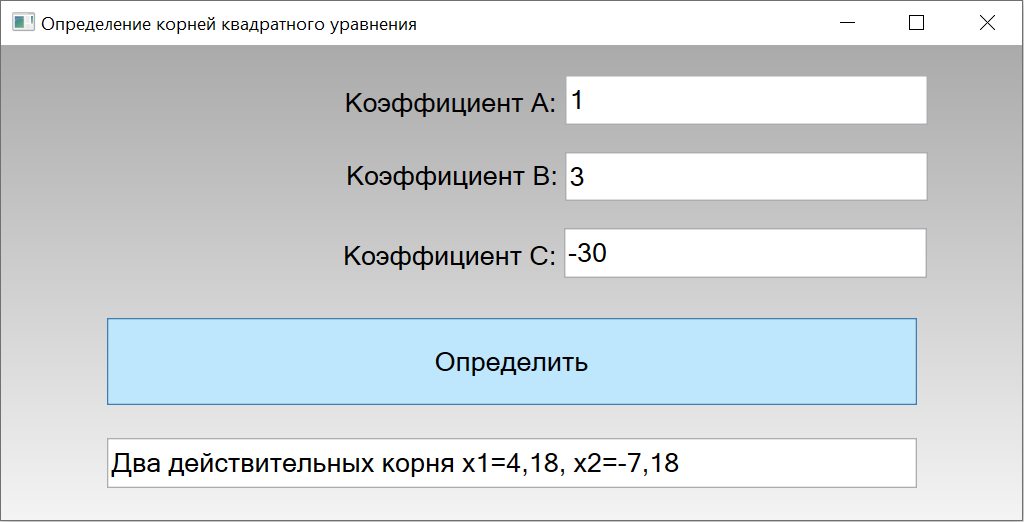
Метод button1\_Click(), содержащий код приложения:

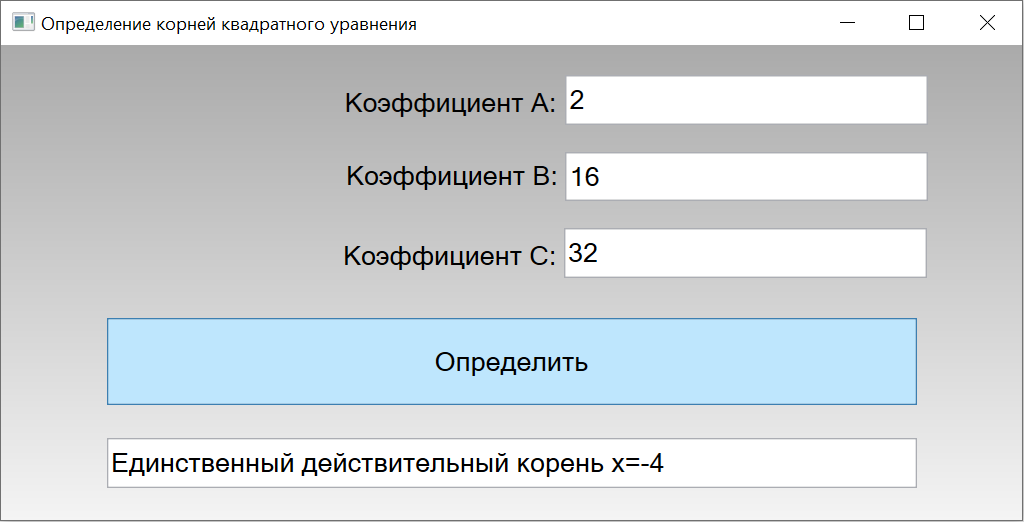


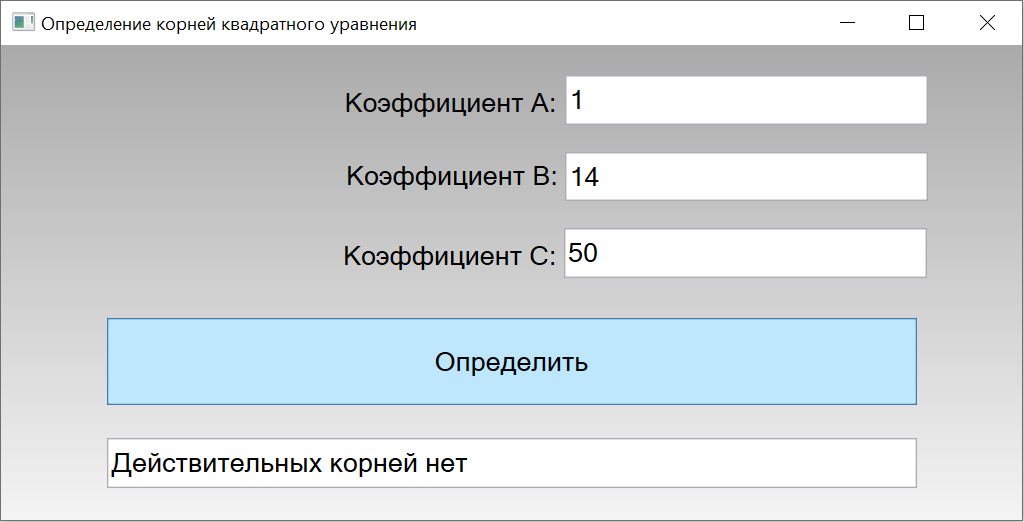
Скрины отлаженной программы:



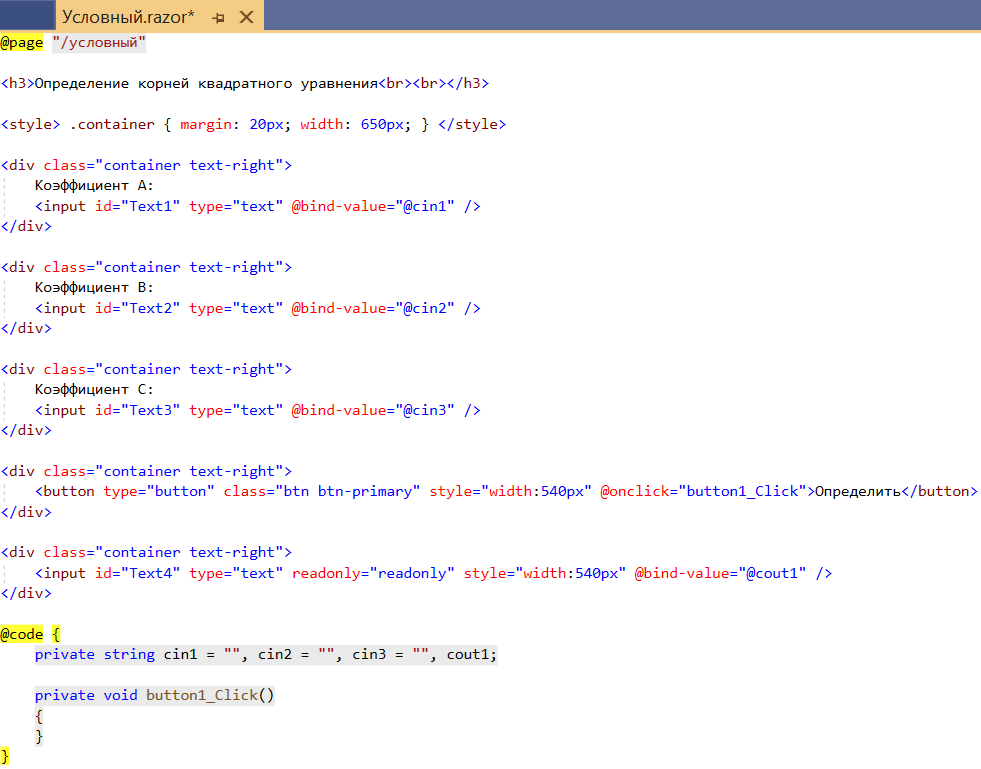


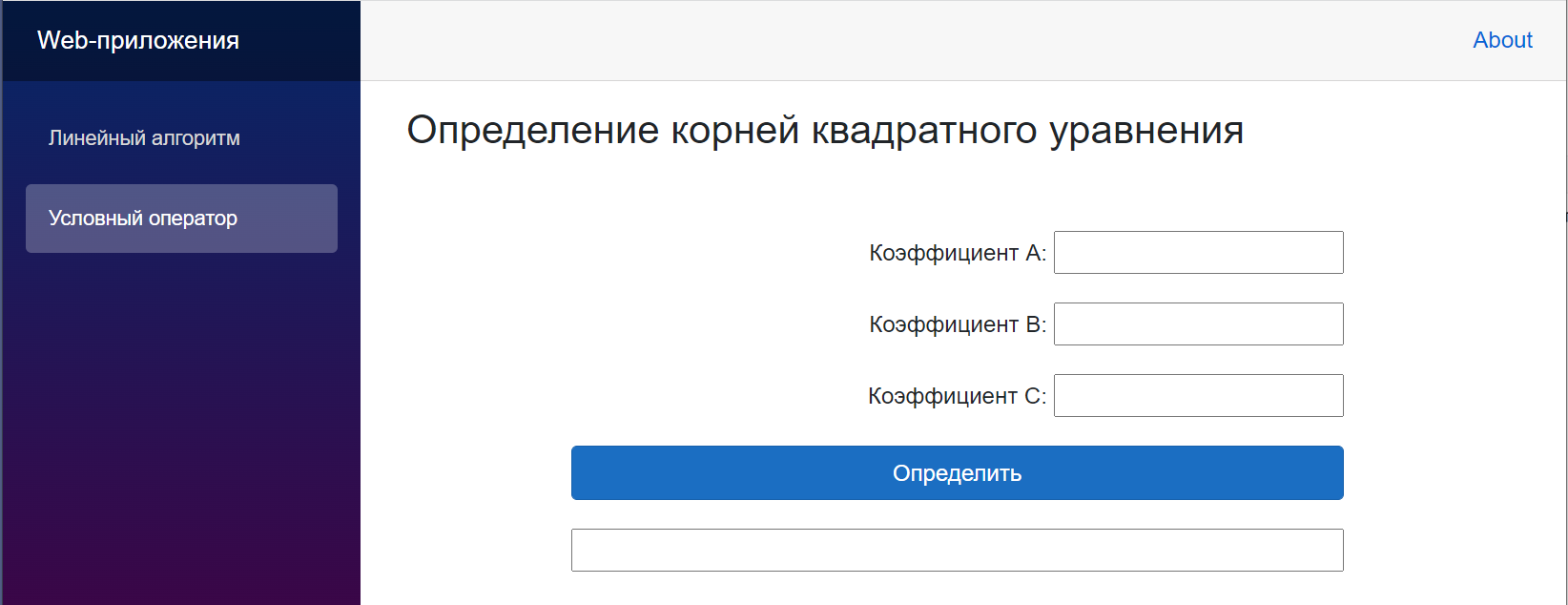




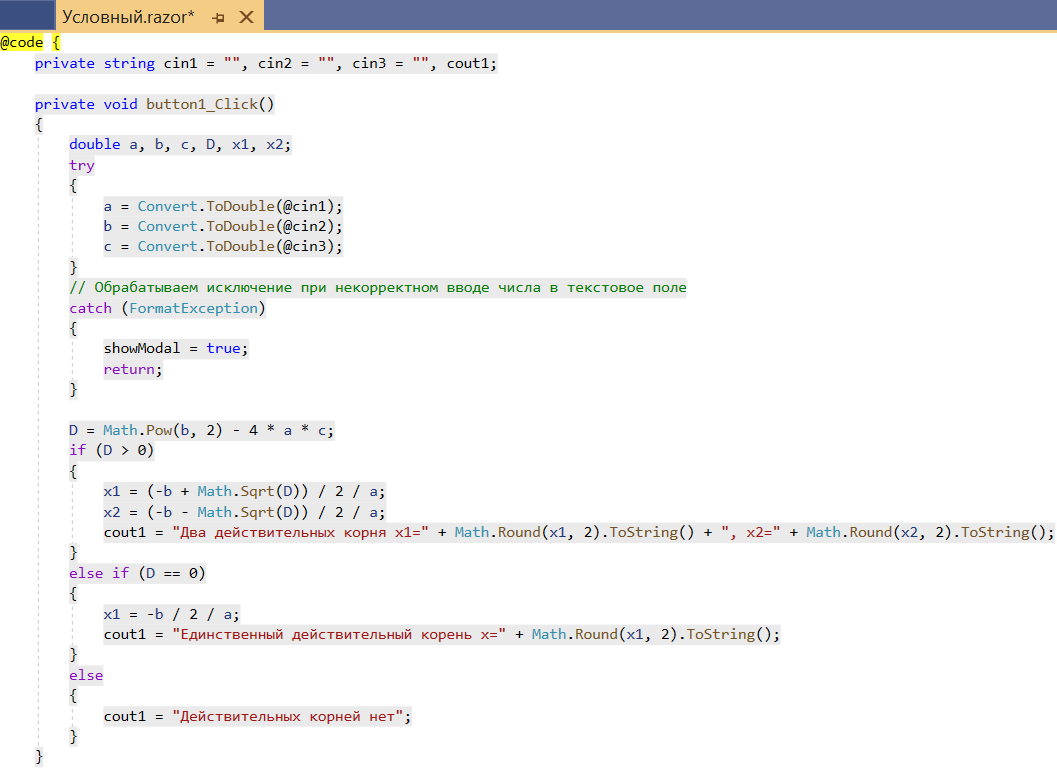


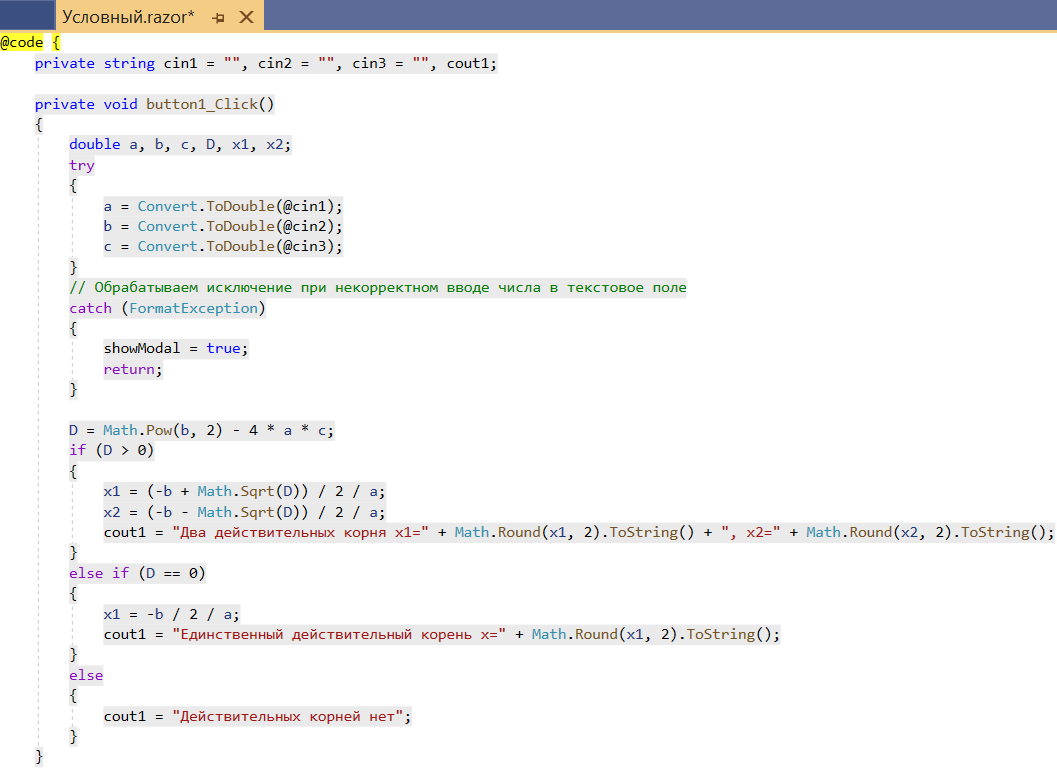
в) Интерфейс приложения WebAssembly Blazor:





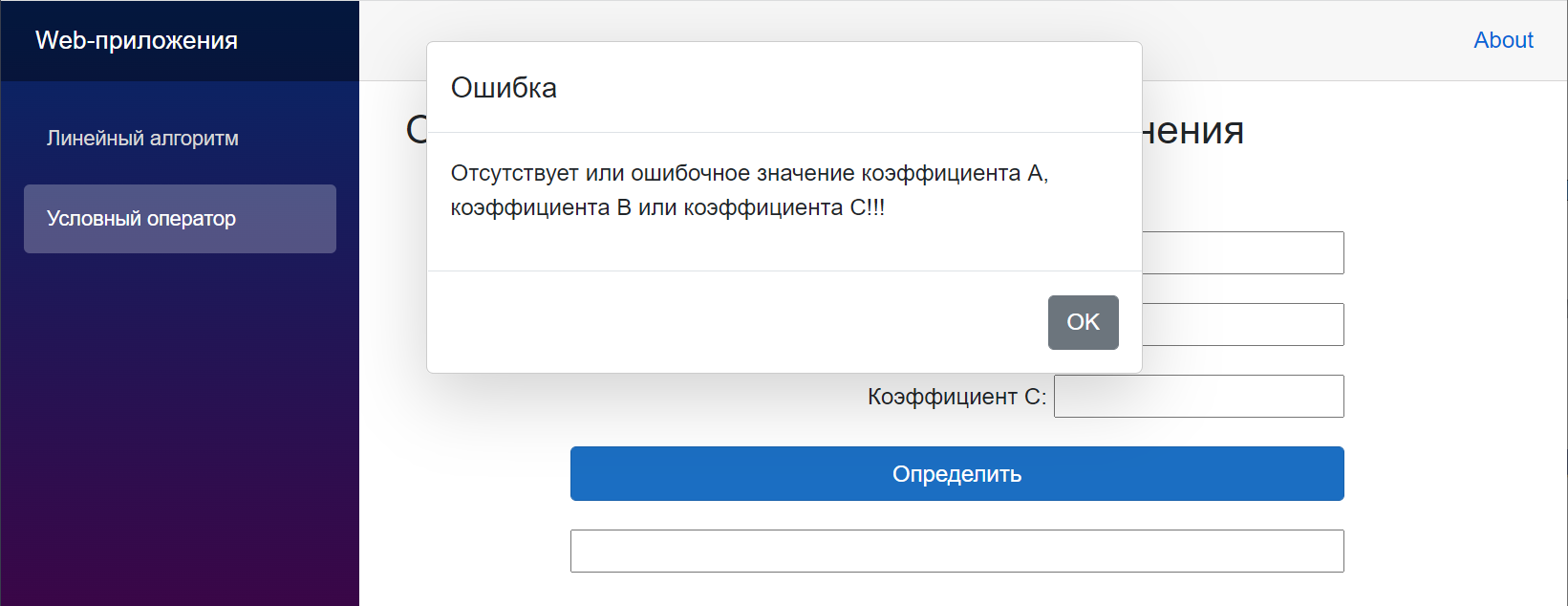
Коды приложения:

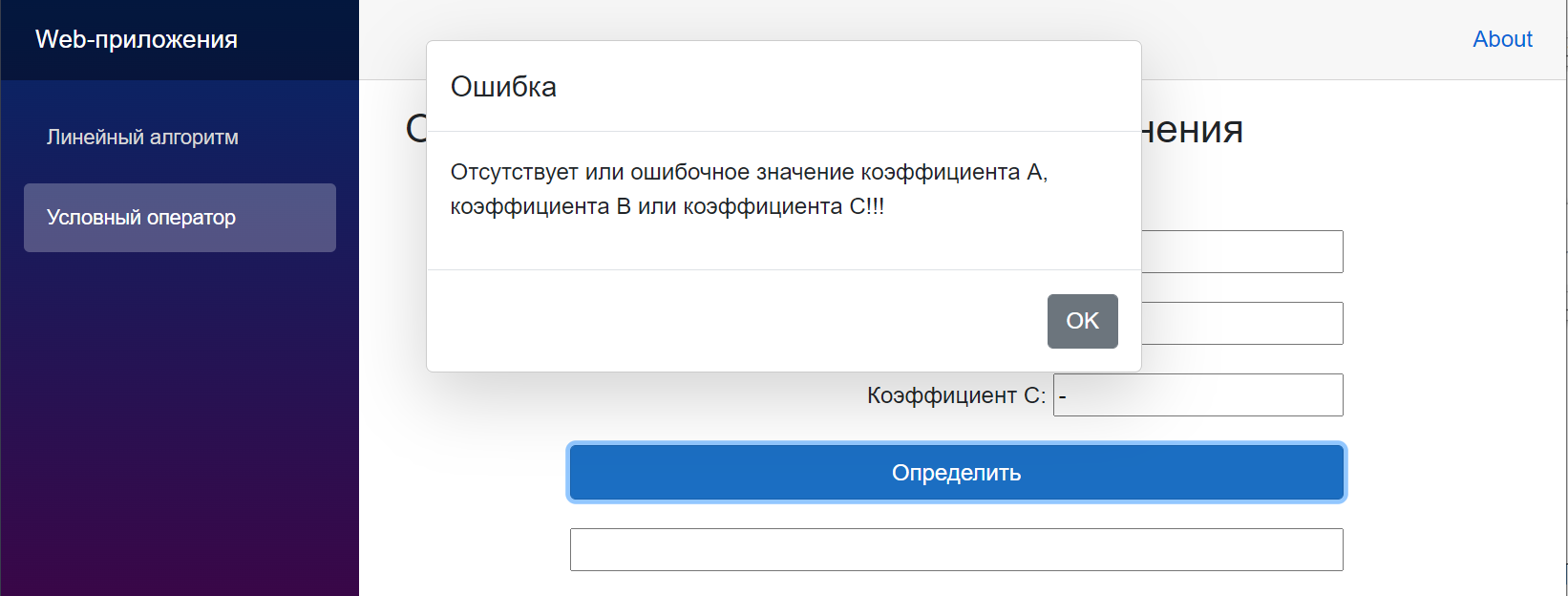


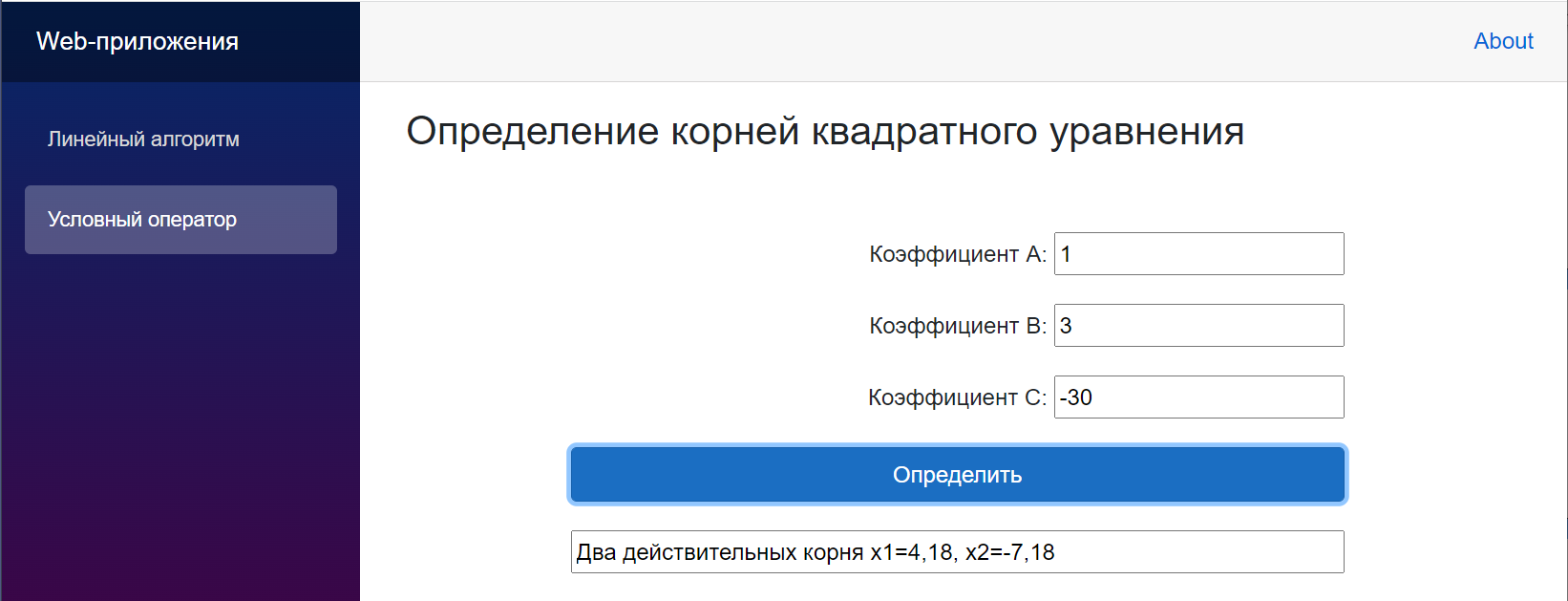


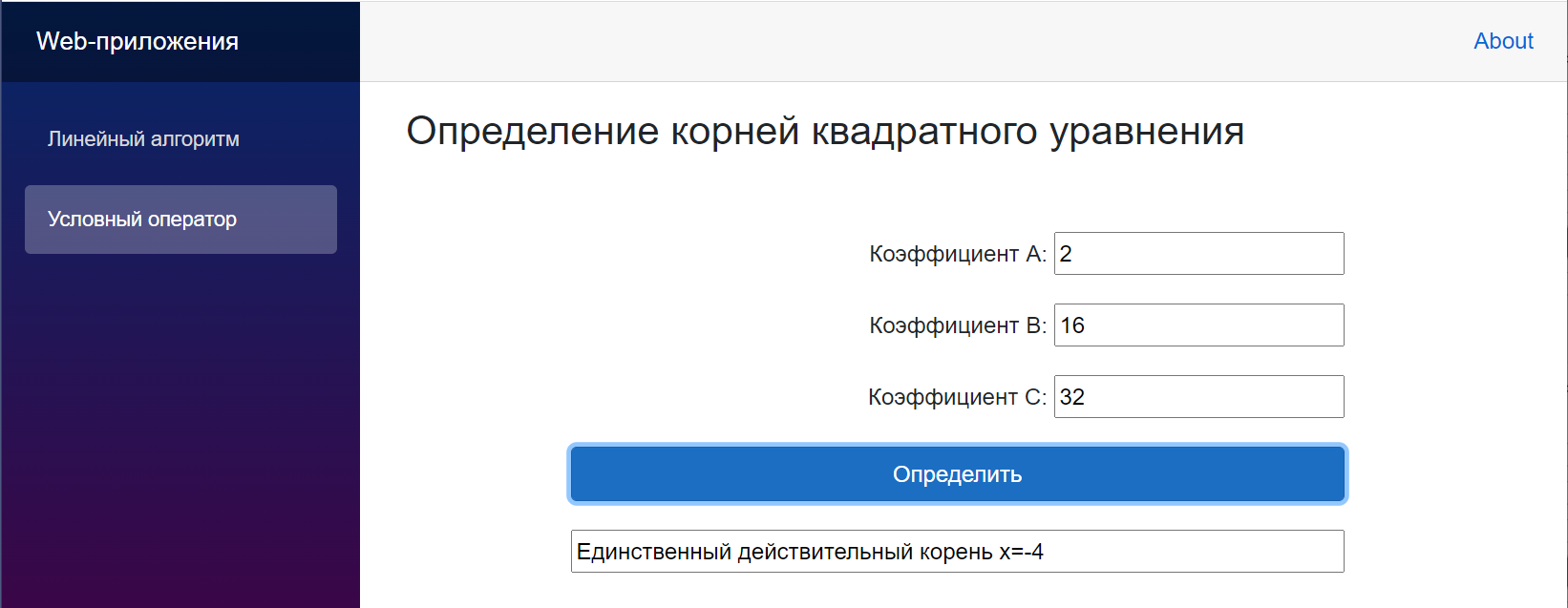


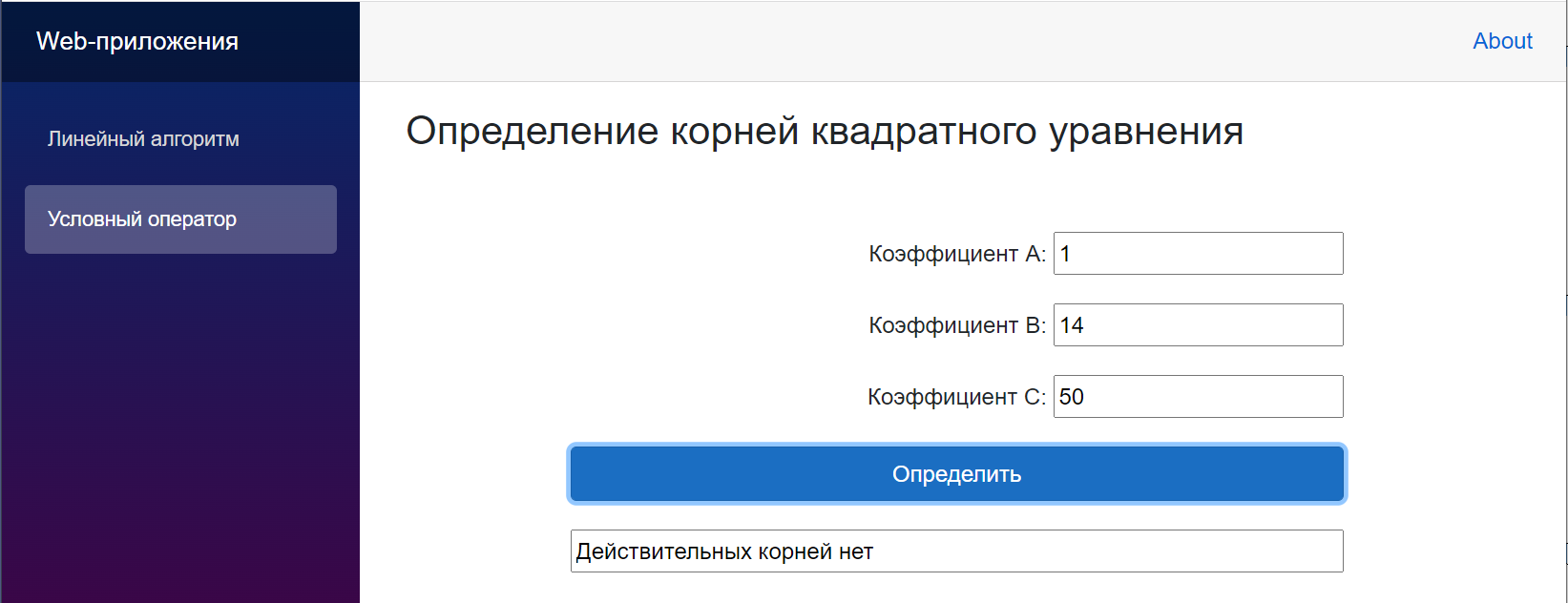
Скрины отлаженной программы:











**4 Порядок выполнения работы**

4.1 Изучить теоретические сведения и задание к работе.

4.2 В соответствии с вариантом задания создать форму приложения Windows Forms (WPF, WebAssembly Blazor).

4.3 В соответствии с вариантом задания разработать и отладить код приложения Windows Forms (WPF, WebAssembly Blazor), используя оператор условия if и исключение, выбрасываемое, если форматы входных значений не соответствуют спецификациям параметров обрабатывающих их методов.

**5 Содержание отчета**

5.1 Название работы

5.2 Цель работы

5.3 Программный код и скрины работы разработанного приложения Windows Forms в соответствии с вариантом задания.

**6 Варианты заданий**

6.1 На числовой оси расположены три точки: *A*, *B*, *C*. Определить, какая из двух последних точек (*B* или *C*) расположена ближе к *A*, и вывести эту точку и ее расстояние от точки *A*. Если пользователь введёт координаты *B* и *C* так, что они будут равноотстоящими от *А*, совпадать с *А* или между собой - выдать соответствующее сообщение.

6.2 Ввести номер года (положительное целое число). Выдать количество дней в этом году, учитывая, что обычный год насчитывает 365 дней, а високосный — 366 дней. Високосным считается год, делящийся на 4, за исключением тех годов, которые делятся на 100 и не делятся на 400 (например, годы 300, 1300 и 1900 не являются високосными, а 1200 и 2000 — являются).

6.3 Ввести три числа *А*, *В*, *С*. Если среди них имеется хотя бы одно четное вывести максимальное из них, иначе - минимальное. Если пользователь введёт числа так, что среди них нельзя будет определить лишь одно максимальное/минимальное - выдать соответствующее сообщение.

6.4 Ввести четыре целых числа *A*, *B*, *C*, *D*, одно из которых отлично от трех других, равных между собой. Определить порядковый номер числа, отличного от остальных. Если пользователь введёт числа так, что они не будут соответствовать условию задачи - выдать сообщение об ошибке.

6.5 Дано целое число. Вывести его строку-описание вида «отрицательное четное число», «нулевое число», «положительное нечетное число» и т. д.

6.6 Ввести три числа *A*, *B*, *C*. Вывести среднее по величине из них (то есть число, расположенное между наименьшим и наибольшим). Если пользователь введёт числа так, что среди них нельзя будет определить среднее - выдать соответствующее сообщение.

6.7 Ввести три переменные вещественного типа: *A*, *B*, *C*. Если их значения упорядочены по возрастанию или убыванию, то удвоить их; в противном случае заменить знак каждой переменной на противоположный. Вывести новые значения переменных *A*, *B*, *C*.

6.8 Ввести три числа *A*, *B*, *C*. Найти сумму двух наибольших из них. Если пользователь введёт числа так, что среди них нельзя будет определить два наибольших - выдать соответствующее сообщение.

6.9 Ввести три целых числа *A*, *B*, *C*. Найти количество положительных и количество отрицательных чисел в исходном наборе.

6.10 Ввести целочисленные координаты трех вершин прямоугольника, стороны которого параллельны координатным осям. Выдать координаты его четвертой вершины. Если пользователь введёт координаты точек так, что нельзя получить прямоугольник со сторонами, параллельными координатным осям, вывести соответствующее сообщение.

6.11 Даны две переменные целого типа: *A* и *B*. Если их значения не равны, то присвоить каждой переменной сумму этих значений, а если равны, то присвоить переменным нулевые значения. Вывести новые значения переменных *A* и *B*.

6.12 Дано целое число, лежащее в диапазоне 1–999. Вывести его строку-описание вида «четное двузначное число», «нечетное трехзначное число» и т. д.

6.13 Ввести три числа *A*, *B*, *C*. Вывести вначале наименьшее, а затем наибольшее из данных чисел. Если пользователь введёт числа так, что среди них нельзя будет определить одно наименьшее/наибольшее - выдать соответствующее сообщение.

6.14 Ввести два целых числа *A* и *B*. Проверить делится ли первое на второе. Вывести на экран сообщение об этом, а также остаток (если он есть) и частное (в любом случае).

6.15 Ввести с клавиатуры целочисленные координаты точки. Определить четверть координатной плоскости, которой принадлежит точка.

6.16 По длинам трех введенных отрезков определить возможность существования треугольника, составленного из этих отрезков. Если такой треугольник существует, то определить, является ли он разносторонним, равнобедренным или равносторонним.

6.17 Ввести координаты точки *X* и *Y* и радиус круга *R*. Определить принадлежит ли данная точка кругу, если его центр находится в начале координат.

**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.