

Санкт-Петербургский Национальный Исследовательский  
Университет Информационных Технологий, Механики и Оптики

Факультет инфокоммуникационных технологий

**Лабораторная работа №4**  
**Создание и использование методов**

Выполнил  
Стафеев И.А.

Группа  
К3221

Проверил  
Иванов С.Е.

Санкт-Петербург,  
2024

## СОДЕРЖАНИЕ

	Стр.
<b>ВВЕДЕНИЕ .....</b>	<b>3</b>
<b>1 Упражнение 1.....</b>	<b>4</b>
<b>2 Упражнение 2.....</b>	<b>5</b>
<b>3 Упражнение 3.....</b>	<b>7</b>
<b>4 Упражнение 4.....</b>	<b>10</b>
<b>5 Упражнение 5.....</b>	<b>14</b>
<b>ЗАКЛЮЧЕНИЕ .....</b>	<b>17</b>

## ВВЕДЕНИЕ

Цель работы: изучение и приобретение навыков работы с методами классов в C#.

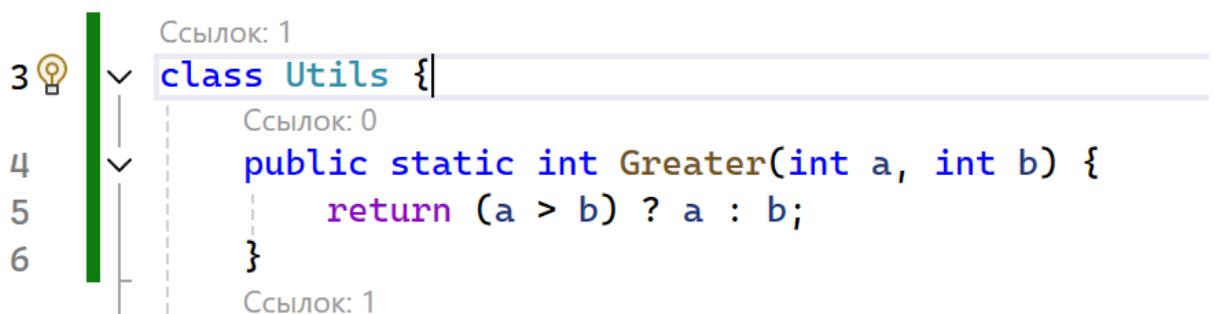
Для достижения цели необходимо выполнить следующие упражнения:

1. Использование параметров в методах, возвращающих значения
2. Использование в методах параметров, передаваемых по ссылке
3. Использование возвращаемых параметров в методах
4. Расчет площади треугольника с помощью метода
5. Вычисление корней квадратного уравнения

## 1 Упражнение 1

Задача: В этом упражнении вы создадите класс **Utils**, в котором определите метод **Greater**. Этот метод будет принимать два целочисленных параметра и возвращать больший из них. Для тестирования работы данного класса Вы будете использовать класс Program, в котором у пользователя будут запрашиваться два числа, далее будет вызываться метод **Utils.Greater**, после чего на экран консоли будет выводиться результат.

Код программы приведен на рисунке 1. В методе класса **Utils** использован тернарный оператор. Поскольку метод статический, то его можно вызывать без создания объекта класса, что и делается в методе Main.



```
3  class Utils {  
4      public static int Greater(int a, int b) {  
5          return (a > b) ? a : b;  
6      }  
    }
```

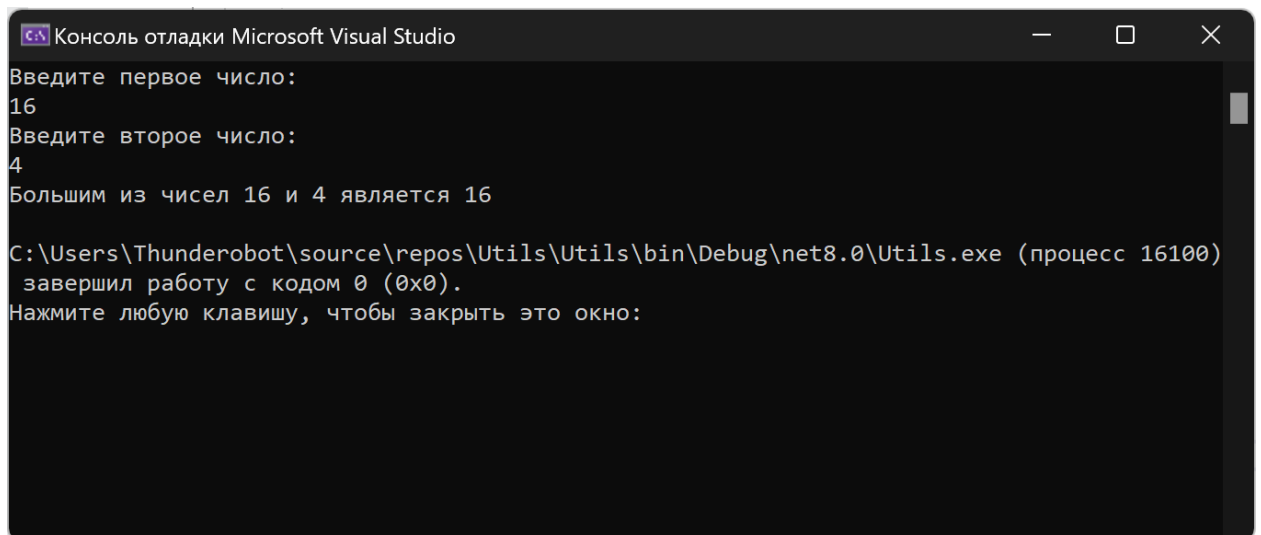
Ссылка: 1

Ссылка: 0

Ссылка: 1

Рисунок 1 — Код программы

Пример выполнения программы приведен на рисунке 2.



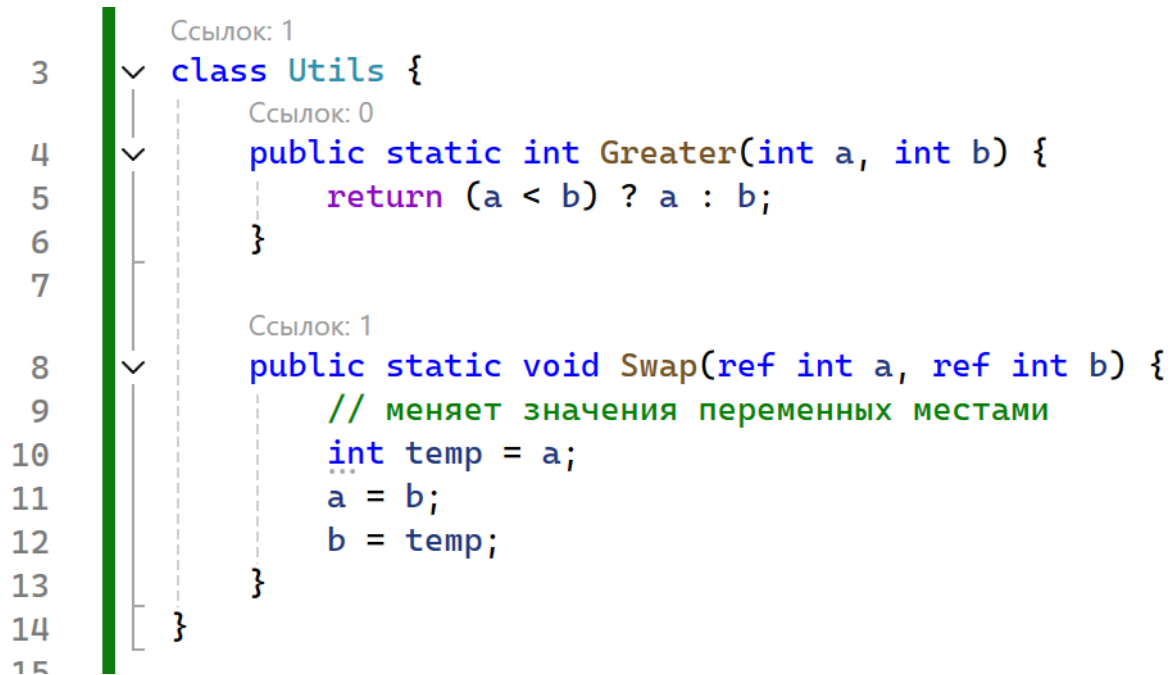
```
Консоль отладки Microsoft Visual Studio  
Введите первое число:  
16  
Введите второе число:  
4  
Большим из чисел 16 и 4 является 16  
  
C:\Users\Thunderobot\source\repos\Utils\Utils\bin\Debug\net8.0\Utils.exe (процесс 16100)  
завершил работу с кодом 0 (0x0).  
Нажмите любую клавишу, чтобы закрыть это окно:
```

Рисунок 2 — Пример выполнения программы

## 2 Упражнение 2

Задача: В этом упражнении вы создадите метод Swap, который поменяет местами значения параметров. При этом вы будете использовать параметры, передаваемые по ссылке

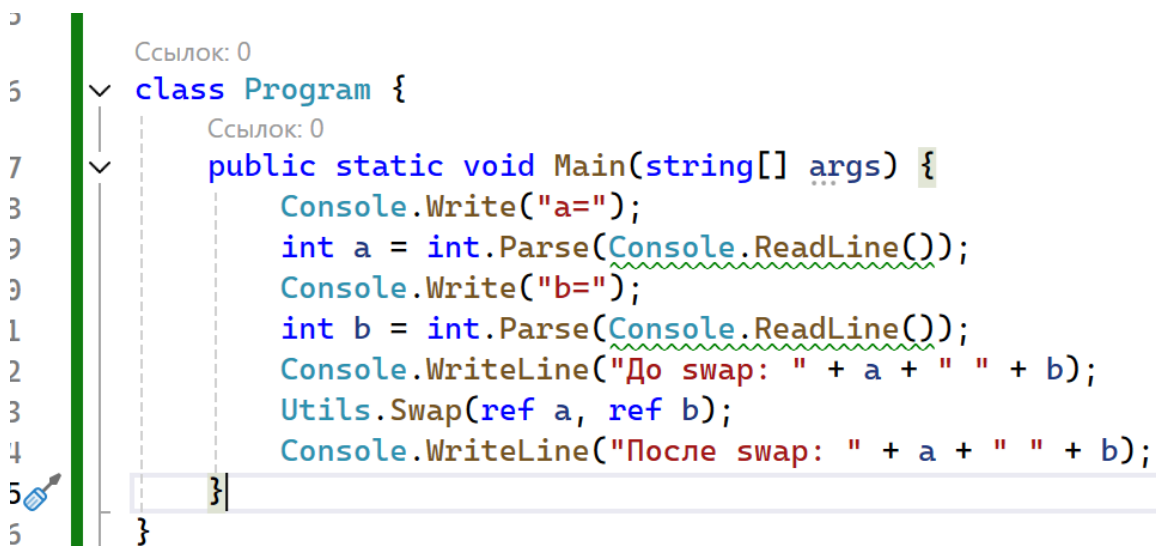
Код класса Utils приведен на рисунке 3.



```
3  class Utils {
4      public static int Greater(int a, int b) {
5          return (a < b) ? a : b;
6      }
7
8      public static void Swap(ref int a, ref int b) {
9          // меняет значения переменных местами
10         int temp = a;
11         a = b;
12         b = temp;
13     }
14 }
15
```

Рисунок 3 — Код класса Utils

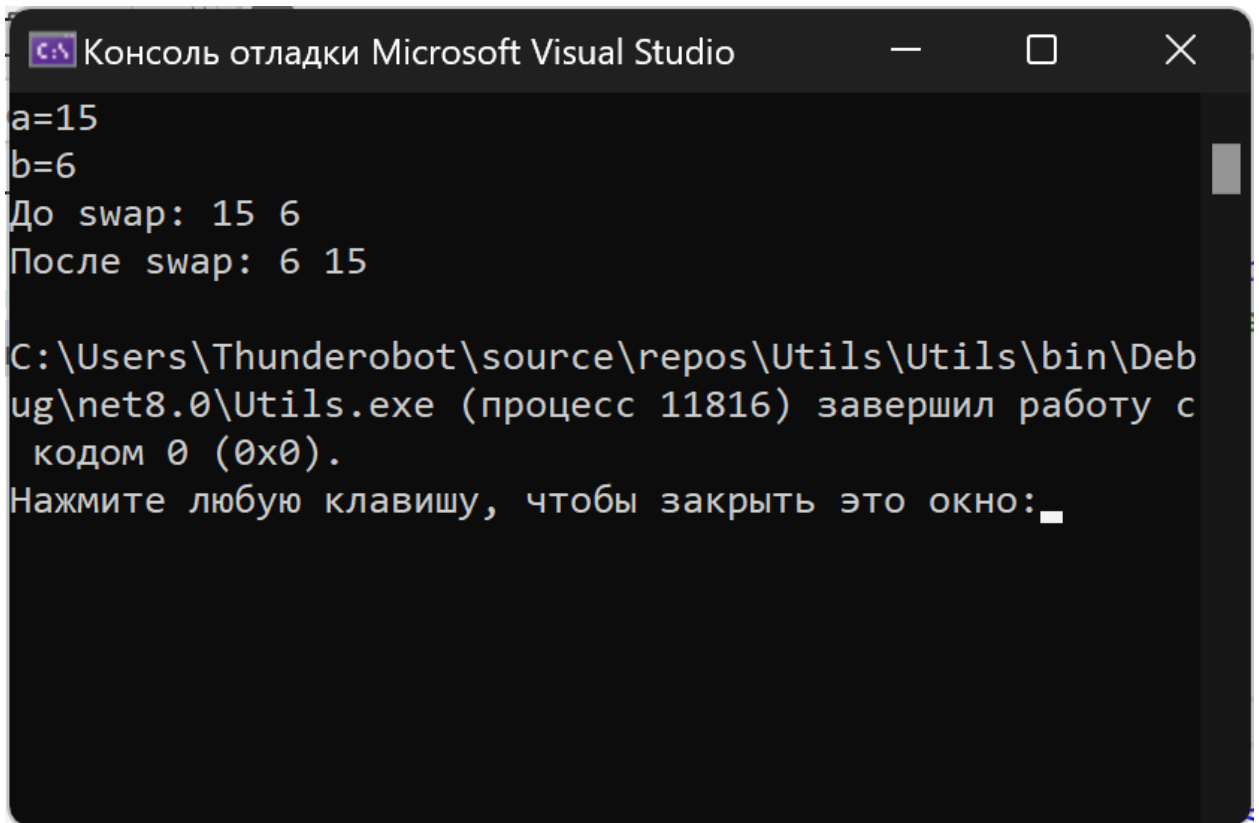
Код метода Main приведен на рисунке 4.



```
5  class Program {
6      public static void Main(string[] args) {
7          Console.Write("a=");
8          int a = int.Parse(Console.ReadLine());
9          Console.Write("b=");
10         int b = int.Parse(Console.ReadLine());
11         Console.WriteLine("До swap: " + a + " " + b);
12         Utils.Swap(ref a, ref b);
13         Console.WriteLine("После swap: " + a + " " + b);
14     }
15 }
```

Рисунок 4 — Код метода Main

Пример выполнения программы на рисунке 5.

The image shows a screenshot of the 'Консоль отладки Microsoft Visual Studio' (Microsoft Visual Studio Debug Console) window. The window has a dark background and a title bar with standard Windows window controls. The text inside the console is as follows:  
a=15  
b=6  
До swap: 15 6  
После swap: 6 15  
  
C:\Users\Thunderobot\source\repos\Utils\Utils\bin\Debug\net8.0\Utils.exe (процесс 11816) завершил работу с кодом 0 (0x0).  
Нажмите любую клавишу, чтобы закрыть это окно:   
The text is displayed in a monospaced font, with the first four lines in white and the subsequent lines in a light blue color. A small white cursor is visible at the end of the last line.

```
Консоль отладки Microsoft Visual Studio
a=15
b=6
До swap: 15 6
После swap: 6 15

C:\Users\Thunderobot\source\repos\Utils\Utils\bin\Debug\net8.0\Utils.exe (процесс 11816) завершил работу с кодом 0 (0x0).
Нажмите любую клавишу, чтобы закрыть это окно: 
```

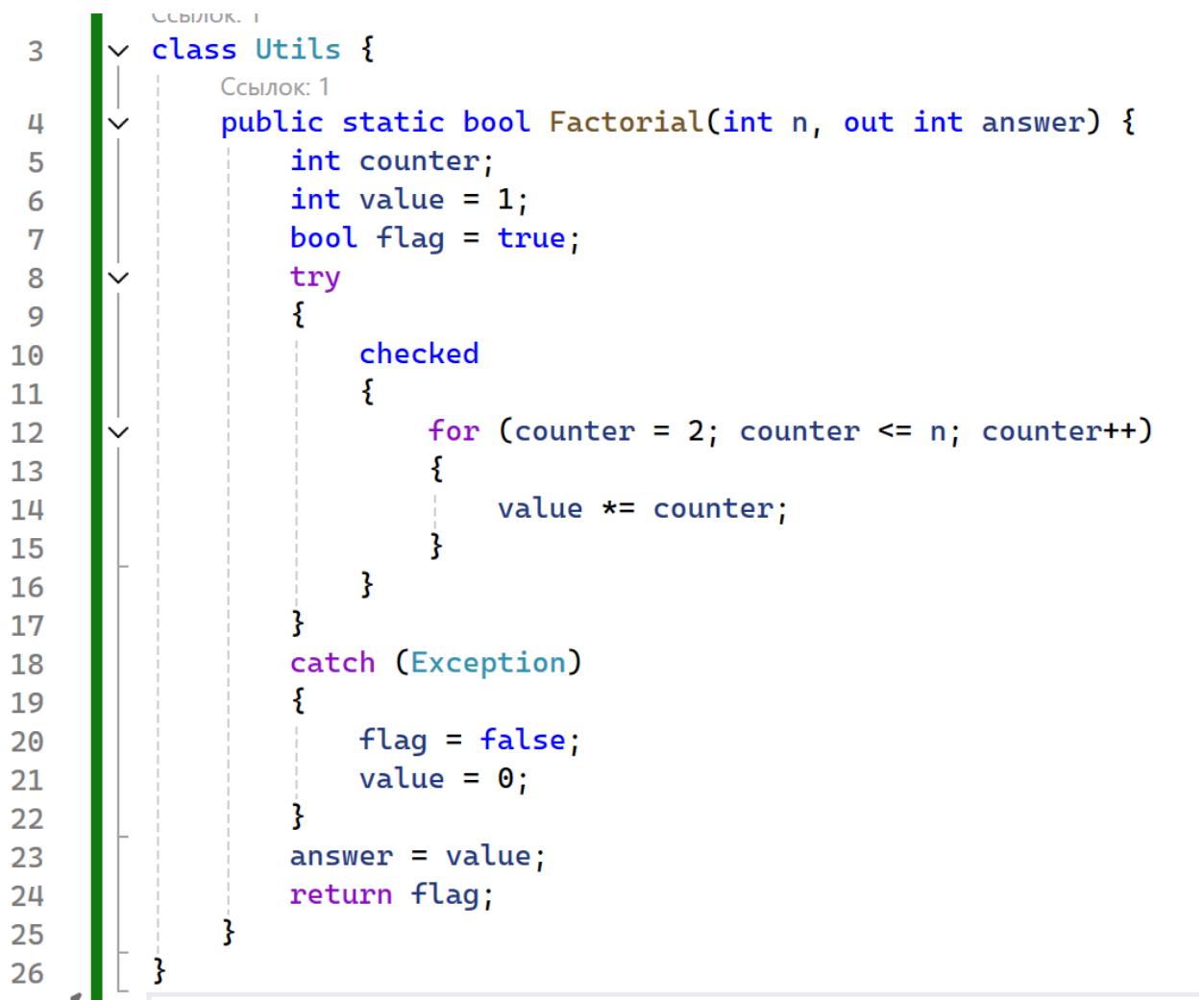
Рисунок 5 — Пример выполнения программы

### 3 Упражнение 3

Задача: Добавьте в класс **Utils** статический открытый метод **Factorial**, имеющий следующие особенности реализации:

- Метод **Factorial** будет использовать два параметра **n** и **answer**: первый параметр типа **int** передается по значению (это число, для которого рассчитывается факториал), второй параметр типа **out int** используется для возвращения результата.
- Метод **Factorial** должен возвращать значение типа **bool**, отражающее успешность выполнения метода, так как может произойти переполнение и выброс исключения

Код класса **Utils** с методом **Factorial** приведен на рисунке 6. Для проверки на переполнение использовано ключевое слово *checked*, вызывающую ошибку при переполнении.



```
3  class Utils {
4      public static bool Factorial(int n, out int answer) {
5          int counter;
6          int value = 1;
7          bool flag = true;
8          try
9          {
10             checked
11             {
12                 for (counter = 2; counter <= n; counter++)
13                 {
14                     value *= counter;
15                 }
16             }
17         }
18         catch (Exception)
19         {
20             flag = false;
21             value = 0;
22         }
23         answer = value;
24         return flag;
25     }
26 }
```

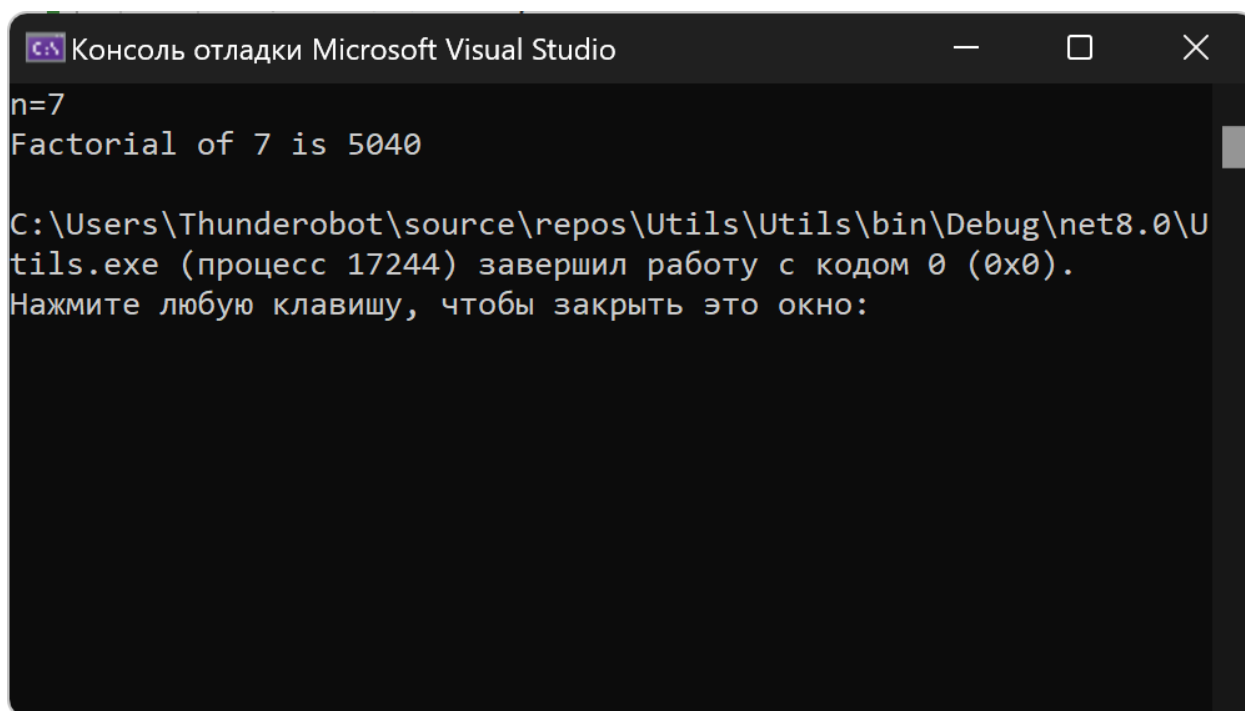
Рисунок 6 — Код класса **Utils**

Код метода Main приведен на рисунке 7.

```
28 class Program {
29     public static void Main(string[] args) {
30         bool flag;
31         int res;
32         Console.Write("n=");
33         int n = int.Parse(Console.ReadLine());
34         flag = Utils.Factorial(n, out res);
35         if (flag) { Console.WriteLine("Factorial of {0} is {1}", n, res); }
36         else { Console.WriteLine("Cannot calculate factorial of {0}", n); }
37     }
38 }
```

Рисунок 7 — Код метода Main

Пример успешного выполнения и выполнения с переполнением можно увидеть на рисунках 8 и 9.

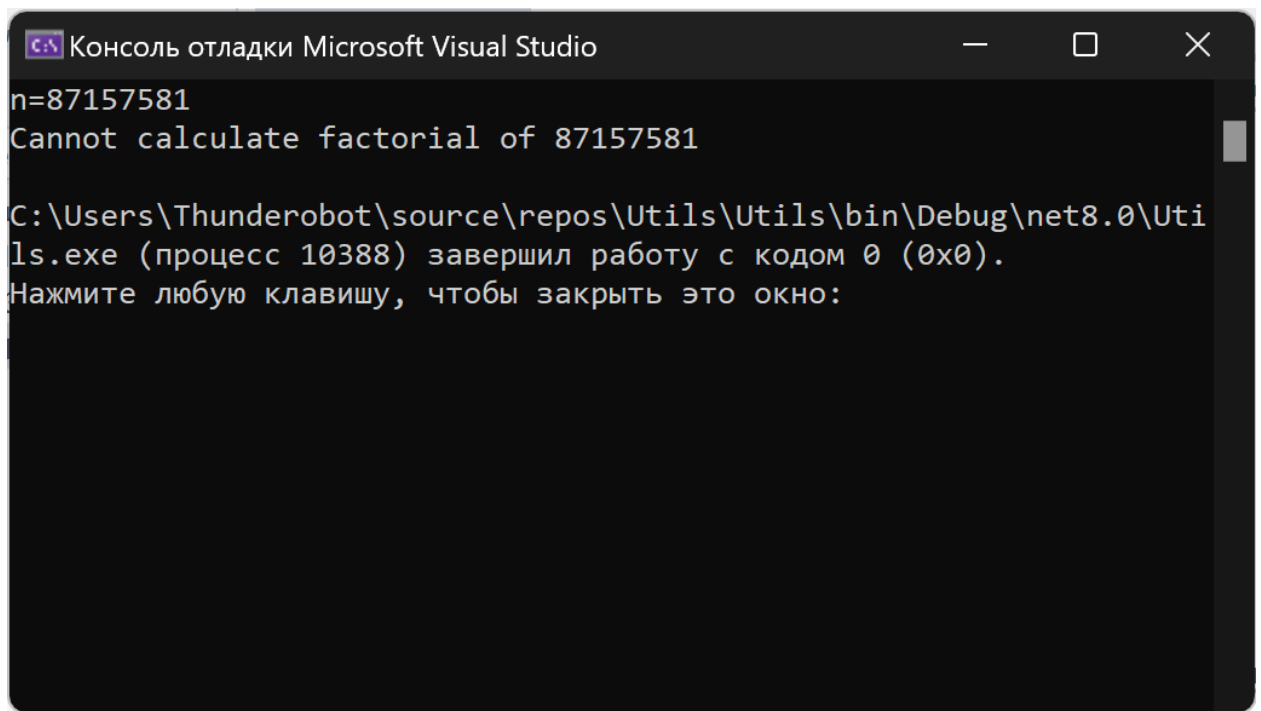


```
Консоль отладки Microsoft Visual Studio
n=7
Factorial of 7 is 5040

C:\Users\Thunderobot\source\repos\Utils\Utils\bin\Debug\net8.0\U
tils.exe (процесс 17244) завершил работу с кодом 0 (0x0).
Нажмите любую клавишу, чтобы закрыть это окно:
```

Рисунок 8 — Пример успешного выполнения





Консоль отладки Microsoft Visual Studio

```
n=87157581
Cannot calculate factorial of 87157581

C:\Users\Thunderobot\source\repos\Utils\Utils\bin\Debug\net8.0\Utils.exe (процесс 10388) завершил работу с кодом 0 (0x0).
Нажмите любую клавишу, чтобы закрыть это окно:
```

Рисунок 9 — Пример выполнения с переполнением

## 4 Упражнение 4

Задача: Требуется создать класс **Operation**, в котором определите:

- статический метод расчета площади треугольника по формуле Герона. Этот метод должен принимать три параметра (стороны треугольника) и возвращать значение площади;
- статический закрытый метод проверки наличия треугольника. Этот метод должен возвращать значение логического типа;
- перегруженный статический метод, который будет принимать один параметр – сторону и вычислять площадь равностороннего треугольника.

Код класса **Operation** показан на рисунке 10. Метод **IsTriangle** проверяет, что треугольник с введенными сторонами существует, иначе вызывает исключение, которое обрабатывается далее в программе. Метод **Square** непосредственно вычисляет площадь.

```
3  class Operation
4  {
5      static void IsTriangle(double a, double b, double c) {
6          // проверка на треугольник
7          if (!(a < b + c && b < a + c && c < a + b)) {
8              throw new ArgumentException();
9          }
10     }
11     public static double Square(double a, double b, double c)
12     {
13         // расчет площади
14         Operation.IsTriangle(a, b, c);
15         double p = (a + b + c) / 2;
16         return Math.Sqrt(p * (p - a) * (p - b) * (p - c));
17     }
18
19     public static double Square(double a) {
20         // расчет площади равностороннего треугольника
21         Operation.IsTriangle(a, a, a);
22         return a * a * Math.Sqrt(3) / 4;
23     }
24 }
25
```

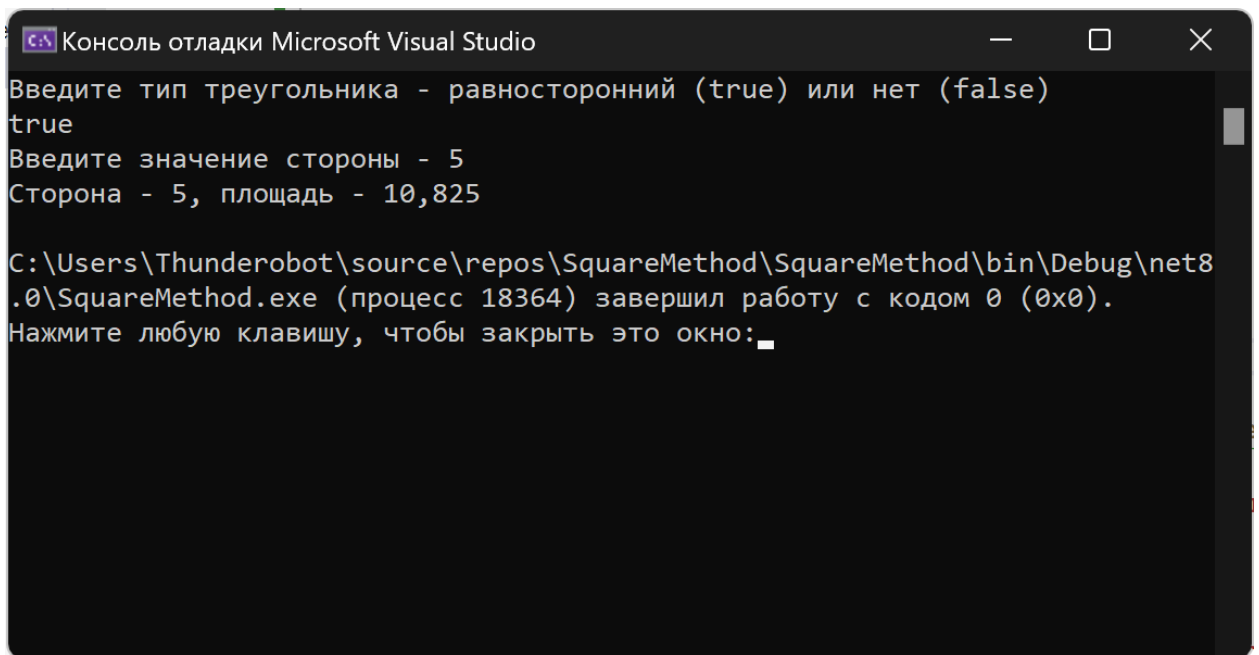
Рисунок 10 — Код класса Operation

Код метода Main приведен на рисунке 11. Пользователь должен ввести булево значение для обозначения того, равнобедренный треугольник или нет. В зависимости от ответа запрашиваются одна или три стороны, после чего вычисляется площадь треугольника. При некорректных входных данных (неподдержка формата или несуществование треугольника с введенными сторонами) выводится соответствующее сообщение.

```
26 class Program
27 {
28     Ссылка: 0
29     public static void Main(string[] args)
30     {
31         double square;
32         try
33         {
34             Console.WriteLine("Введите тип треугольника - равнобедренный (true) или нет (false)");
35             bool type = bool.Parse(Console.ReadLine());
36             if (type) {
37                 Console.WriteLine("Введите значение стороны - ");
38                 double a = double.Parse(Console.ReadLine());
39                 square = Operation.Square(a);
40                 Console.WriteLine("Сторона - {0}, площадь - {1:f3}", a, square);
41             }
42             else {
43                 Console.WriteLine("Введите значение первой стороны - ");
44                 double a = double.Parse(Console.ReadLine());
45                 Console.WriteLine("Введите значение второй стороны - ");
46                 double b = double.Parse(Console.ReadLine());
47                 Console.WriteLine("Введите значение третьей стороны - ");
48                 double c = double.Parse(Console.ReadLine());
49                 square = Operation.Square(a, b, c);
50                 Console.WriteLine("Стороны равны {0}, {1} и {2}. Площадь равна {3:f3}", a, b, c, square);
51             }
52         }
53         catch (ArgumentException e) { Console.WriteLine("Не существует треугольника с указанными сторонами"); }
54         catch (FormatException e) { Console.WriteLine("Данные введены в неверном формате: {0}", e.Message); }
55     }
56 }
```

Рисунок 11 — Код метода Main

Пример выполнения программы для равнобедренного треугольника показан на рисунке 12.

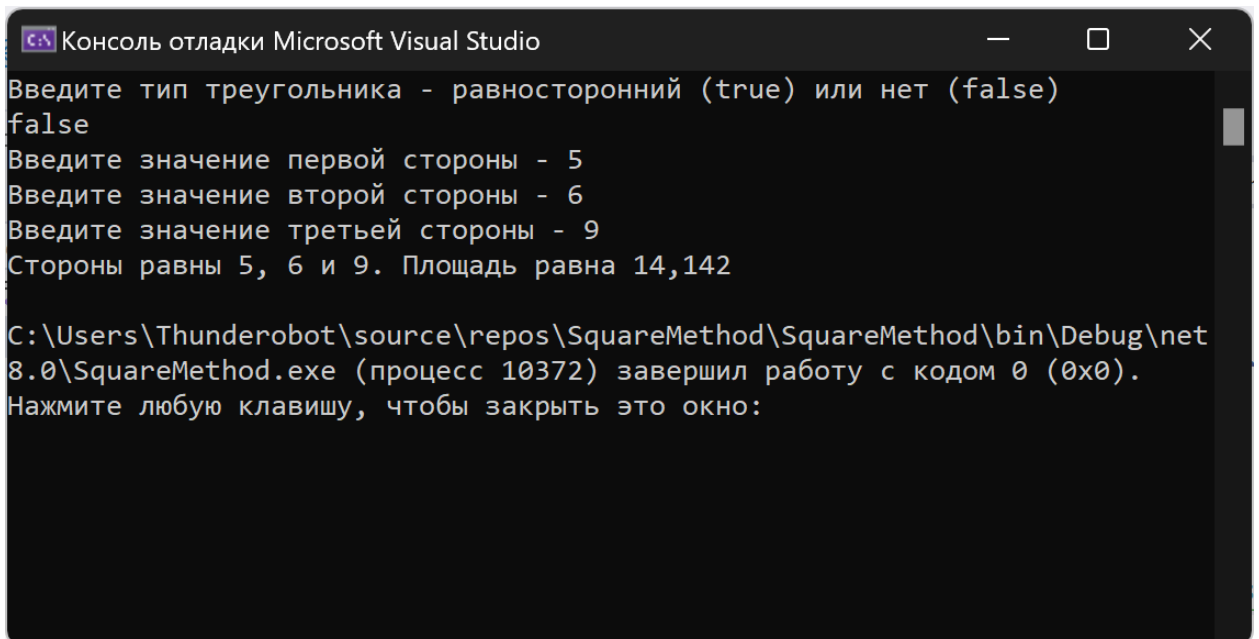


```
Консоль отладки Microsoft Visual Studio
Введите тип треугольника - равносторонний (true) или нет (false)
true
Введите значение стороны - 5
Сторона - 5, площадь - 10,825

C:\Users\Thunderobot\source\repos\SquareMethod\SquareMethod\bin\Debug\net8
.0\SquareMethod.exe (процесс 18364) завершил работу с кодом 0 (0x0).
Нажмите любую клавишу, чтобы закрыть это окно:■
```

Рисунок 12 — Пример выполнения 1

Пример выполнения программы для разностороннего треугольника показан на рисунке 13.



```
Консоль отладки Microsoft Visual Studio
Введите тип треугольника - равносторонний (true) или нет (false)
false
Введите значение первой стороны - 5
Введите значение второй стороны - 6
Введите значение третьей стороны - 9
Стороны равны 5, 6 и 9. Площадь равна 14,142

C:\Users\Thunderobot\source\repos\SquareMethod\SquareMethod\bin\Debug\net
8.0\SquareMethod.exe (процесс 10372) завершил работу с кодом 0 (0x0).
Нажмите любую клавишу, чтобы закрыть это окно:■
```

Рисунок 13 — Пример выполнения 2

Пример выполнения программы для несуществующего треугольника показан на рисунке 14.

```
Консоль отладки Microsoft Visual Studio
Введите тип треугольника - равносторонний (true) или нет (false)
false
Введите значение первой стороны - 1
Введите значение второй стороны - 2
Введите значение третьей стороны - 3
Не существует треугольника с указанными сторонами

C:\Users\Thunderobot\source\repos\SquareMethod\SquareMethod\bin\Debug\net8.0\SquareMethod.exe (процесс 14452) завершил работу с кодом 0 (0x0).
Нажмите любую клавишу, чтобы закрыть это окно:
```

Рисунок 14 — Пример выполнения 3

Пример выполнения программы при некорректных данных показан на рисунке 15.

```
Консоль отладки Microsoft Visual Studio
Введите тип треугольника - равносторонний (true) или нет (false)
false
Введите значение первой стороны - aksjfgakgsfk
Данные введены в неверном формате: The input string 'aksjfgakgsfk' was not in a correct format.

C:\Users\Thunderobot\source\repos\SquareMethod\SquareMethod\bin\Debug\net8.0\SquareMethod.exe (процесс 12204) завершил работу с кодом 0 (0x0).
Нажмите любую клавишу, чтобы закрыть это окно:
```

Рисунок 15 — Пример выполнения 4

## 5 Упражнение 5

Задача: Требуется реализовать метод вычисления корней квадратного уравнения:

- функция должна возвращать значение 1, если корни найдены, значение нуля, если оба корня совпадают, и значение -1, если корней не существует.
- значения корней уравнений должны возвращаться в качестве аргументов функции, передаваемых по ссылке

Для решения был создан класс **Solver** с методом **GetRoots** (16), принимающим коэффициенты уравнения и по ссылке принимающий корни.

```
4 class Solver {
5
6     public static int GetRoots(double a, double b, double c, ref double x1, ref double x2) {
7         double D = b * b - 4 * a * c;
8         if (D < 0) return -1;
9         x1 = (-b - Math.Sqrt(D)) / (2 * a);
10        x2 = (-b + Math.Sqrt(D)) / (2 * a);
11        return (x1 == x2) ? 0 : 1;
12    }
13 }
14
```

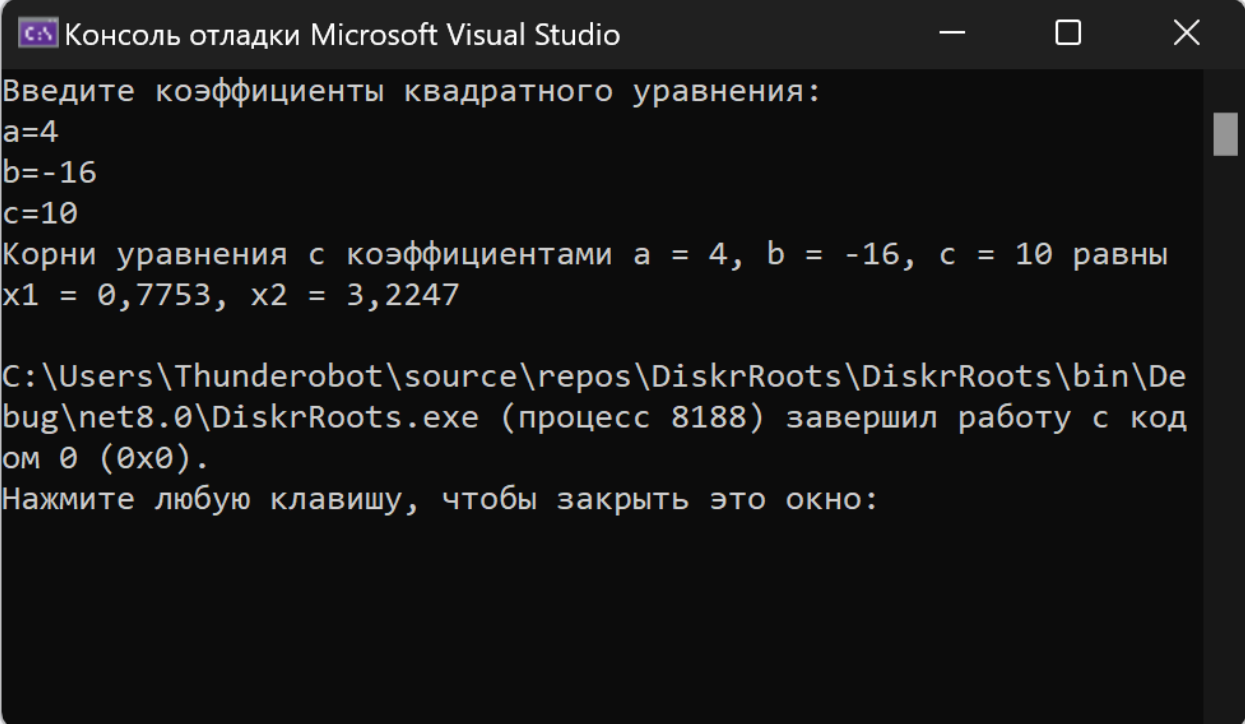
Рисунок 16 — Код класса Solver

Код метода Main показан на рисуне 17. В этом методе запрашиваются коэффициенты уравнения, затем вычисляются корни, и в зависимости от их количества выводится соответствующее сообщение.

```
15 class Program
16 {
17     public static void Main(string[] args)
18     {
19         double x1 = 0, x2 = 0;
20         Console.WriteLine("Введите коэффициенты квадратного уравнения:");
21         Console.Write("a=");
22         double a = double.Parse(Console.ReadLine());
23         Console.Write("b=");
24         double b = double.Parse(Console.ReadLine());
25         Console.Write("c=");
26         double c = double.Parse(Console.ReadLine());
27         int res = Solver.GetRoots(a, b, c, ref x1, ref x2);
28         if (res == -1) { // корней нет
29             Console.WriteLine("Корней уравнения с коэффициентами a = {0}, b = {1}, c = {2} нет.", a, b, c);
30         }
31         else if (res == 0) { // один корень
32             Console.WriteLine("Корень уравнения с коэффициентами a = {0}, b = {1}, c = {2} равен x1 = x2 = {3:f4}",
33                 a, b, c, x1);
34         }
35         else { // два корня
36             Console.WriteLine("Корни уравнения с коэффициентами a = {0}, b = {1}, c = {2} равны x1 = {3:f4}, x2 = {4:f4}",
37                 a, b, c, x1, x2);
38         }
39     }
40 }
```

Рисунок 17 — Код метода Main

Примеры выполнения программы для уравнения с двумя корнями, одним корнем и без корней приведены на рисунках 18, 19 и 20.

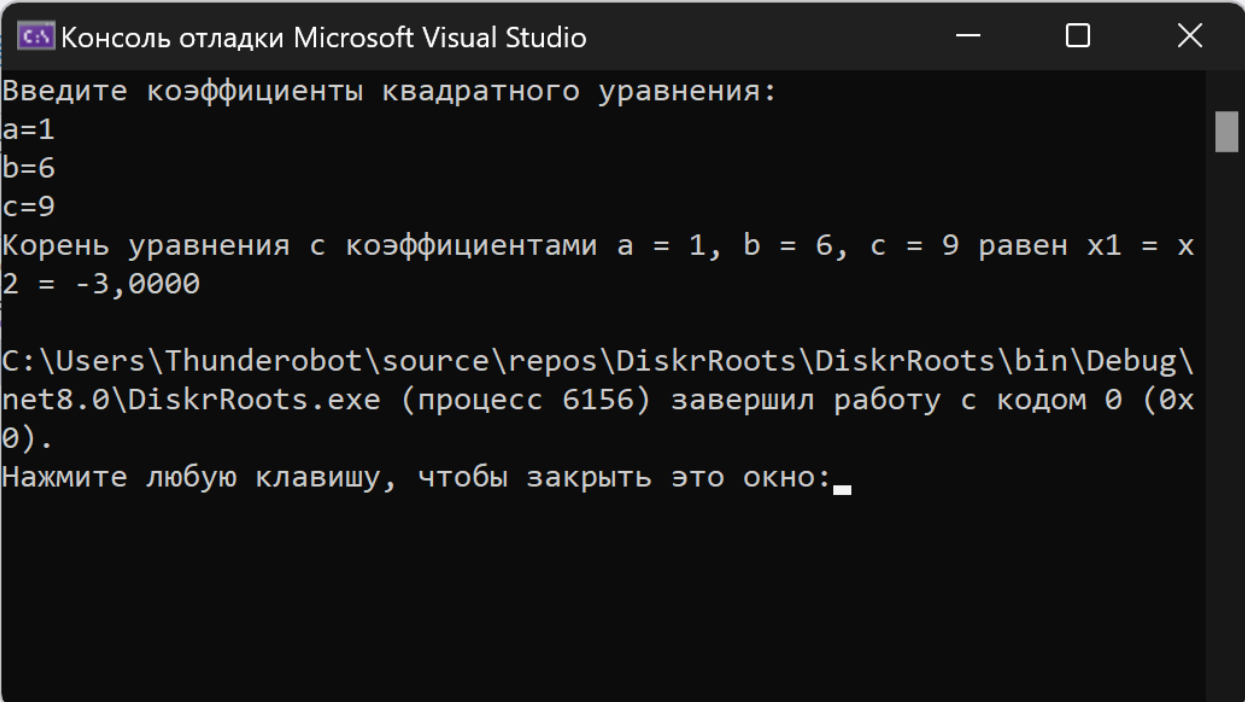


```
Консоль отладки Microsoft Visual Studio

Введите коэффициенты квадратного уравнения:
a=4
b=-16
c=10
Корни уравнения с коэффициентами a = 4, b = -16, c = 10 равны
x1 = 0,7753, x2 = 3,2247

C:\Users\Thunderobot\source\repos\DiskrRoots\DiskrRoots\bin\De
bug\net8.0\DiskrRoots.exe (процесс 8188) завершил работу с код
ом 0 (0x0).
Нажмите любую клавишу, чтобы закрыть это окно:
```

Рисунок 18 — Пример с двумя корнями

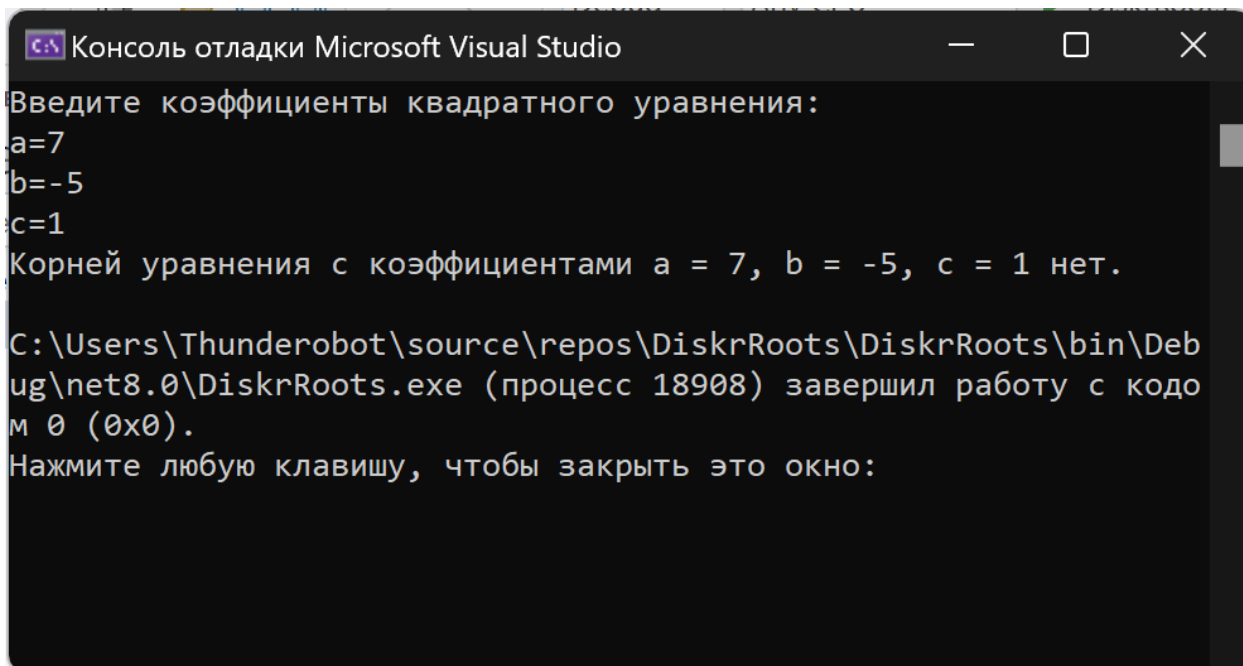


```
Консоль отладки Microsoft Visual Studio

Введите коэффициенты квадратного уравнения:
a=1
b=6
c=9
Корень уравнения с коэффициентами a = 1, b = 6, c = 9 равен x1 = x
2 = -3,0000

C:\Users\Thunderobot\source\repos\DiskrRoots\DiskrRoots\bin\Debug\
net8.0\DiskrRoots.exe (процесс 6156) завершил работу с кодом 0 (0x
0).
Нажмите любую клавишу, чтобы закрыть это окно: _
```

Рисунок 19 — Пример с одним корнем



```
Консоль отладки Microsoft Visual Studio

Введите коэффициенты квадратного уравнения:
a=7
b=-5
c=1
Корней уравнения с коэффициентами a = 7, b = -5, c = 1 нет.

C:\Users\Thunderobot\source\repos\DiskrRoots\DiskrRoots\bin\Debug\net8.0\DiskrRoots.exe (процесс 18908) завершил работу с кодом 0 (0x0).
Нажмите любую клавишу, чтобы закрыть это окно:
```

Рисунок 20 — Пример без корней



## ЗАКЛЮЧЕНИЕ

В ходе выполнения лабораторной работы были выполнены все требуемые упражнения. Цель работы достигнута. Получены знания и навыки по работе с методами классов в C#.