

Московский Государственный Технический Университет
им. Н. Э. Баумана

ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №2
ПО КУРСУ "БКИТ"

Решение квадратных уравнений

Выполнил:
Водка Игорь Эдуардович
27 сентября 2017 г.

Проверил:
Гапанюк Юрий Евгеньевич
«___» _____ 2017 г.

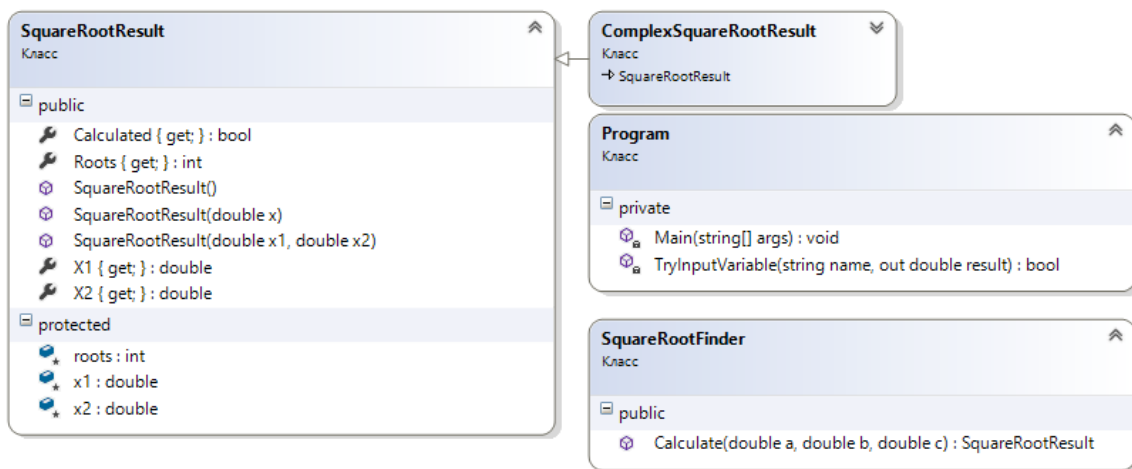
Москва, 2017 г.

1 Описание задания

Разработать программу для решения квадратного уравнения.

1. Программа должна быть разработана в виде консольного приложения на языке C#.
2. Программа осуществляет ввод с клавиатуры коэффициентов A, B, C, вычисляет дискриминант и корни уравнения (в зависимости от дискриминанта).
3. Если коэффициент A, B, C введен некорректно, то необходимо проигнорировать некорректное значение и ввести коэффициент повторно.

2 Диаграмма классов



3 Код программы

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace SharpLab1
8 {
9     class Program
10     {
11         static bool TryInputVariable(string name, out double result)
12         {
13             Console.WriteLine("Введите {0}:", name);
14             return double.TryParse(Console.ReadLine(), out result);
15         }
16
17         static void Main(string[] args)
18         {
19             while (true)
```

```

20     {
21         double a = 0;
22         double b = 0;
23         double c = 0;
24
25         bool parsingResult = false;
26
27         while (parsingResult = !TryInputVariable("a", out a)) ;
28         while (parsingResult = !TryInputVariable("b", out b)) ;
29         while (parsingResult = !TryInputVariable("c", out c)) ;
30
31         var result = SquareRootFinder.Calculate(a, b, c);
32
33         if (result.Roots == 1)
34         {
35             Console.WriteLine("X is {0}", result.X1);
36         }
37         else if (result.Roots == 2)
38         {
39             Console.WriteLine("X1 is {0}", result.X1);
40             Console.WriteLine("X2 is {0}", result.X2);
41         }
42         else if (result is ComplexSquareRootResult)
43         {
44             Console.WriteLine("No real roots found.");
45         }
46
47         Console.ReadKey(true);
48     }
49 }
50 }
51 }

```

Листинг 1: Program.cs

```

1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace SharpLab1
8  {
9      class SquareRootFinder
10     {
11         public static SquareRootResult Calculate(double a, double b, double c)
12         {
13             double D = b * b - 4 * a * c;
14
15             if (D == 0)
16             {
17                 return new SquareRootResult(
18                     (-b) / 2 * a
19                 );
20             }
21             else if (D > 0)
22             {
23                 return new SquareRootResult(
24                     (-b - Math.Sqrt(D)) / 2 * a,
25                     (-b + Math.Sqrt(D)) / 2 * a
26                 );
27             }
28         }
29     }
30 }

```

```

26         );
27     }
28     else
29         return new ComplexSquareRootResult();
30 }
31 }
32 }

```

Листинг 2: SquareRootFinder.cs

```

1  using System;
2
3  namespace SharpLab1
4  {
5      class SquareRootResult
6      {
7          protected int roots = 0;
8          protected double x1 = 0;
9          protected double x2 = 0;
10
11         public SquareRootResult() { }
12
13         public SquareRootResult(double x)
14         {
15             this.x1 = this.x2 = x;
16             this.roots = 1;
17         }
18
19         public SquareRootResult(double x1, double x2)
20         {
21             this.x1 = x1;
22             this.x2 = x2;
23             this.roots = 2;
24         }
25
26         public bool Calculated
27         {
28             get
29             {
30                 return this.roots > 0;
31             }
32         }
33
34         public int Roots
35         {
36             get
37             {
38                 return this.roots;
39             }
40         }
41
42         public double X1
43         {
44             get
45             {
46                 if (!Calculated)
47                     throw new Exception("Square root is not calculated yet.");
48                 return this.x1;
49             }
50         }

```

```

51
52     public double X2
53     {
54         get
55         {
56             if (!Calculated)
57                 throw new Exception("Square root is not calculated yet.");
58             return this.x2;
59         }
60     }
61 }
62 }

```

Листинг 3: SquareRootResult.cs

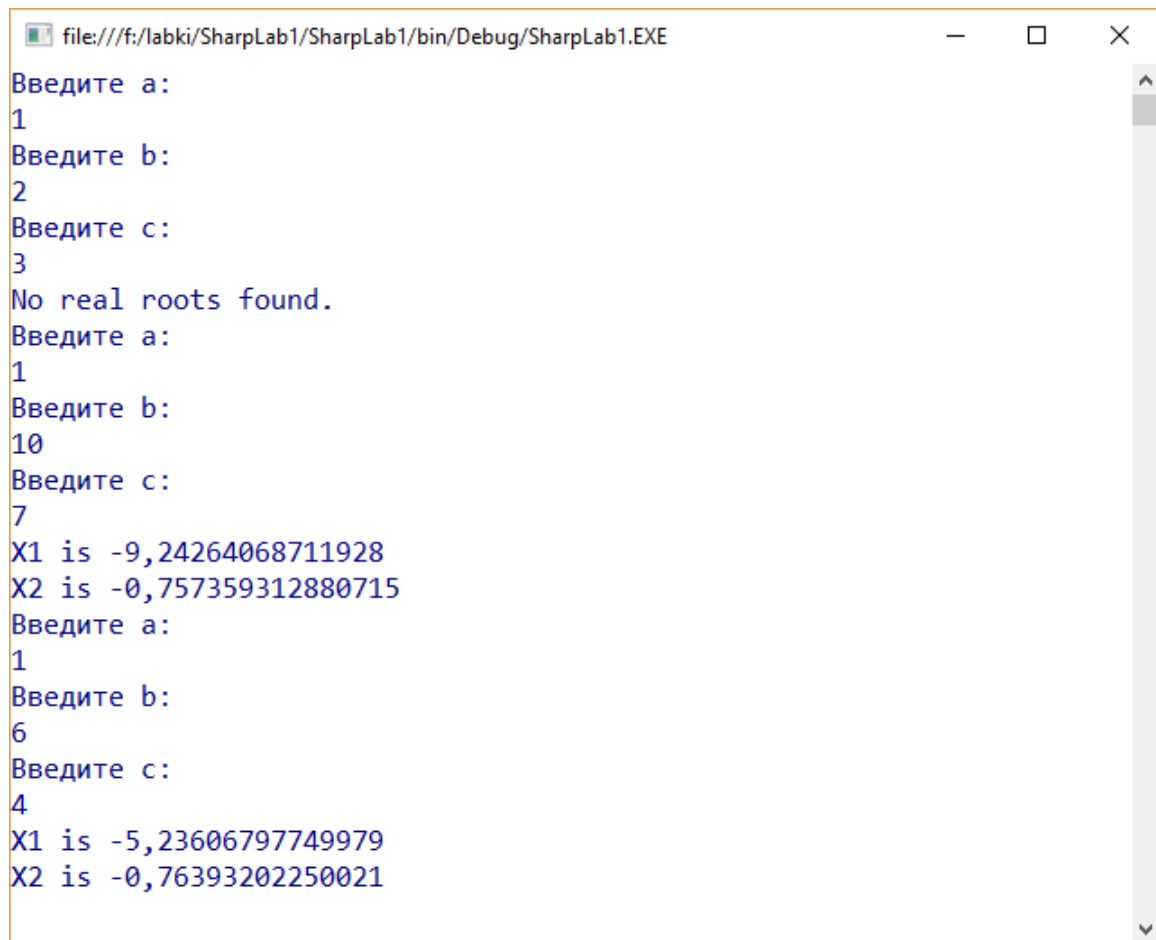
```

1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace SharpLab1
8  {
9      /// <summary></summary>
10     /// <remarks>Вариант> класса для комплексных корней </remarks>
11     class ComplexSquareRootResult : SquareRootResult
12     {
13         public ComplexSquareRootResult()
14         {
15             this.roots = 0;
16         }
17     }
18 }

```

Листинг 4: ComplexSquareRootResult.cs

4 Скриншот работы программы



```
file:///f:/labki/SharpLab1/SharpLab1/bin/Debug/SharpLab1.EXE
Введите a:
1
Введите b:
2
Введите c:
3
No real roots found.
Введите a:
1
Введите b:
10
Введите c:
7
X1 is -9,24264068711928
X2 is -0,757359312880715
Введите a:
1
Введите b:
6
Введите c:
4
X1 is -5,23606797749979
X2 is -0,76393202250021
```