



ОТЧЁТ ПО ЛАБРАТОРНОЙ РАБОТЕ №4

Объектно-ориентированное программирование

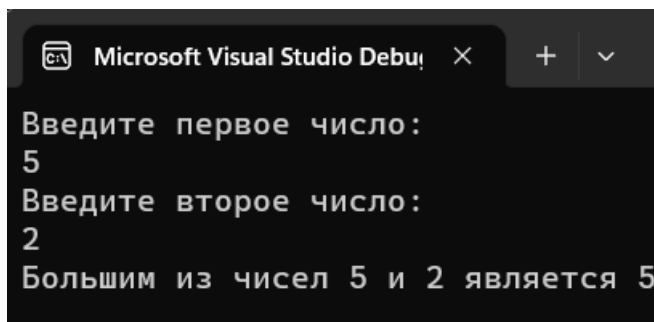
Подготовил Шкода Глеб Ярославович
Студент 2 курса факультета ИКТ Университета ИТМО
Группа K32211
Преподаватель Иванов Сергей Евгеньевич

Упражнение 1

Реализуем описанную в тексте задания программу:

```
1  using System;
2
3  1 reference
4  class Units {
5      1 reference
6      public static int Greater(int a, int b) {
7          if (a > b)
8              return a;
9          else
10             return b;
11         }
12     }
13     0 references
14 class Program {
15     0 references
16     static void Main() {
17
18         int x, y;
19         Console.WriteLine("Введите первое число:");
20         x = int.Parse(Console.ReadLine());
21         Console.WriteLine("Введите второе число:");
22         y = int.Parse(Console.ReadLine());
23         int greater = Units.Greater(x, y);
24         Console.WriteLine("Большим из чисел {0} и {1} является {2}", x, y, greater);
25     }
26 }
```

Протестируем её, чтобы убедиться в правильности выполнения задания:



```
Microsoft Visual Studio Debug Console
Введите первое число:
5
Введите второе число:
2
Большим из чисел 5 и 2 является 5
```

Упражнение 2

Добавим метод Swap к созданному классу Units и протестируем его.

```

1  using System;
2
3  class Units {
4      public static int Greater(int a, int b) {
5          if (a > b)
6              return a;
7          else
8              return b;
9      }
10
11     public static void Swap(ref int a, ref int b) {
12         int tmp = b;
13         b = a;
14         a = tmp;
15     }
16 }
17
18 class Program {
19     static void Main() {
20         int x, y;
21         Console.WriteLine("Введите первое число:");
22         x = int.Parse(Console.ReadLine());
23         Console.WriteLine("Введите второе число:");
24         y = int.Parse(Console.ReadLine());
25         Console.WriteLine("До swap: \t" + x + " " + y);
26         Units.Swap(ref x, ref y);
27         Console.WriteLine("После swap: \t" + x + " " + y);
28     }
29 }
30

```

```

Microsoft Visual Studio Debug Console
Введите первое число:
2
Введите второе число:
5
До swap:          2 5
После swap:       5 2

```

Упражнение 3

Реализуем метод расчёта факториала числа с защитой от переполнения int переменной.

```

1  using System;
2
3  1 reference
4  class Units {
5      0 references
6      public static int Greater(int a, int b) {
7          if (a > b)
8              return a;
9          else
10             return b;
11         }
12
13     0 references
14     public static void Swap(ref int a, ref int b) {
15         int tmp = b;
16         b = a;
17         a = tmp;
18     }
19
20     1 reference
21     public static bool Factorial(int n, out int ans) {
22         ans = 1;
23         bool ok = true;
24         try {
25             checked {
26                 for (int i = 2; i <= n; i++)
27                     ans *= i;
28             }
29         }
30         catch (Exception) {
31             ans = -1;
32             ok = false;
33         }
34         return ok;
35     }
36 }
37
38 0 references
39 class Program {
40     0 references
41     static void Main() {
42
43         Console.WriteLine("Number for factorial:");
44         int x = int.Parse(Console.ReadLine());
45         int f;
46         if (Units.Factorial(x, out f))
47             Console.WriteLine("Factorial({0}) = {1}", x, f);
48         else
49             Console.WriteLine("Cannot compute factorial");
50     }
51 }

```

Проверим работу программы в штатном режиме:

```

Microsoft Visual Studio Debug Console
Number for factorial:
5
Factorial(5) = 120

```

А также проверим защиту от переполнений:

```

Microsoft Visual Studio Debug Console
Number for factorial:
50
Cannot compute factorial

```

Упражнение 4

Для расчёта площади треугольника будем использовать формулу Герона. Перед расчётом проверим входные данные по условию вырожденного треугольника, при этом равносторонний треугольник вырожденным быть не может, поэтому в этом случае проверку производить не следует. На вход у пользователя сначала попросим указать тип треугольника, а затем нужное количество сторон. В случае, если входные данные не верны, укажем на ошибку в случае, если введённые данные соответствуют вырожденному треугольнику, сообщим об этом, иначе же выведем искомую площадь.

```

1 using System;
2
3
4 2 references
5
6 class Operation {
7     1 reference
8     private static bool Check(double a, double b, double c) {
9         if (a + b + c - Math.Min(a, Math.Min(b, c)) <= Math.Max(a, Math.Max(b, c)))
10             return false;
11             return true;
12     }
13
14     1 reference
15     public static double Calc_Square(double a, double b, double c) {
16         if (Check(a, b, c)) {
17             double p = (a + b + c) / 2;
18             return (Math.Sqrt(p * (p - a) * (p - b) * (p - c)));
19         }
20         return -1;
21     }
22
23     1 reference
24     public static double Calc_Square(double a) {
25         return (Math.Sqrt(3) * a * a / 4);
26     }
27 }

```

```

23 class Program {
24     0 references
25     static void Main() {
26         Console.WriteLine("Укажите тип треугольника (1 - равносторонний, 2 - неравносторонний)");
27         int type = int.Parse(Console.ReadLine());
28         switch (type) {
29             case 1:
30                 Console.WriteLine("Введите сторону равностороннего треугольника");
31                 double a = double.Parse(Console.ReadLine());
32                 Console.WriteLine("Площадь треугольника: {0}", Math.Round(Operation.Calc_Square(a), 2));
33                 break;
34             case 2:
35                 Console.WriteLine("Введите стороны треугольника");
36                 var input = Console.ReadLine().Split();
37                 a = double.Parse(input[0]);
38                 double b = double.Parse(input[1]), c = double.Parse(input[2]);
39                 double sq = Operation.Calc_Square(a, b, c);
40                 if (sq > 0)
41                     Console.WriteLine("Площадь треугольника: {0}", Math.Round(sq, 2));
42                 else
43                     Console.WriteLine("Вырожденный треугольник");
44                 break;
45             default:
46                 Console.WriteLine("Неверный входные данные");
47                 break;
48         }
49     }
50 }
51

```

Тестирование:

```
Microsoft Visual Studio Debug Console
Укажите тип треугольника (1 – равносторонний, 2 – неравносторонний)
1
Введите сторону равностороннего треугольника
2
Площадь треугольника: 1,73
```

```
Microsoft Visual Studio Debug Console
Укажите тип треугольника (1 – равносторонний, 2 – неравносторонний)
2
Введите стороны треугольника
3 4 5
Площадь треугольника: 6
```

```
Microsoft Visual Studio Debug Console
Укажите тип треугольника (1 – равносторонний, 2 – неравносторонний)
2
Введите стороны треугольника
2 5 100
Вырожденный треугольник
```

```
Microsoft Visual Studio Debug Console
Укажите тип треугольника (1 – равносторонний, 2 – неравносторонний)
3
Неверный входные данные
```

Упражнение 5

Попросим у пользователя на вход коэффициенты уравнения. Далее расчёт корней будем осуществлять с помощью дискриминанта. Как известно, если дискриминант меньше 0, то вещественных корней у уравнения нет, если равен 0, то корни совпадают, а если больше, то различны. В случае, если вещественные корни существуют их можно рассчитать также с помощью выделенного дискриминанта по известным формулам. После этого, остаётся только вывести полученный ответ на экран.

```

1  using System;
2  1 reference
3  class Operation {
4
5      1 reference
6      public static int CalcRoots(double a, double b, double c, out double x1, out double x2) {
7          x1 = 0;
8          x2 = 0;
9          double d = b * b - 4 * a * c;
10         if (d > 0) {
11             x1 = (-b + Math.Sqrt(d)) / (2 * a);
12             x2 = (-b - Math.Sqrt(d)) / (2 * a);
13             return 1;
14         }
15         else if (d < 0)
16             return -1;
17         else {
18             x1 = -b / (2 * a);
19             return 0;
20         }
21     }
22     0 references
23 }
24
25 class Program {
26     0 references
27     static void Main() {
28
29         Console.WriteLine("Введите коэффициенты квадратного уравнения");
30         var input = Console.ReadLine().Split();
31         double a = double.Parse(input[0]), b = double.Parse(input[1]), c = double.Parse(input[2]);
32         int op = Operation.CalcRoots(a, b, c, out double x1, out double x2);
33         switch (op) {
34             case -1:
35                 Console.WriteLine("Корней уравнения с коэффициентами a = {0}, b = {1}, c = {2} нет", a, b, c);
36                 break;
37             case 0:
38                 Console.WriteLine("Корни уравнения с коэффициентами a = {0}, b = {1}, c = {2} равны x1 = x2 = {3}", a, b, c, x1);
39                 break;
40             case 1:
41                 Console.WriteLine("Корни уравнения с коэффициентами a = {0}, b = {1}, c = {2} равны x1 = {3} и x2 = {4}", a, b, c, x2, x1);
42                 break;
43         }
44     }
45 }

```

Тестирование:

```

Microsoft Visual Studio Debug Console
Введите коэффициенты квадратного уравнения
1 -2 1
Корни уравнения с коэффициентами a = 1, b = -2, c = 1 равны x1 = x2 = 1

```

```

Microsoft Visual Studio Debug Console
Введите коэффициенты квадратного уравнения
1 4 6
Корней уравнения с коэффициентами a = 1, b = 4, c = 6 нет

```

```

Microsoft Visual Studio Debug Console
Введите коэффициенты квадратного уравнения
1 -2 0
Корни уравнения с коэффициентами a = 1, b = -2, c = 0 равны x1 = 0 и x2 = 2

```