

Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики

Кафедра программных систем

**Лабораторная работа №4**

**Создание и использование методов**

Выполнил: Мороз

Илья Олегович

Группа № K3120

Проверила: Казанова П.П.

Санкт-Петербург

2019

**Цель работы:**

Целью работы является знакомство с методами в языке C# и их применение на практике.

**Ход работы:**

**Упражнение 1.**

1. Был создан новый проект Utils. В данном проекте был создан новый класс Utils.
2. В классе Utils был реализован метод Greater, который возвращает наибольшее из двух чисел.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Utils

{

class Utils

{

public static int Greater (int a,int b)

{

if (a > b)

{

return a;

}

else

{

return b;

}

}

}

}

Листинг 1 – класс Utils

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Utils

{

class Program

{

static void Main(string[] args)

{

Console.Write("Введите первое число: ");

int x = int.Parse(Console.ReadLine());

Console.Write("Введите второе число: ");

int y = int.Parse(Console.ReadLine());

int greater = Utils.Greater(x, y);

Console.WriteLine("Большим из чисел {0} и {1} является число {2}", x, y, greater);

}

}

}

Листинг 2 – класс Program

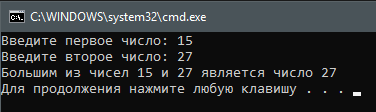


Рисунок 1 – результат работы программы

**Упражнение 2.**

1. В класс Utils был добавлен метод Swap, который меняет значения двух переменных, передаваемых в метод по ссылке.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Utils

{

class Utils

{

public static int Greater(int a, int b)

{

if (a > b)

{

return a;

}

else

{

return b;

}

}

public static void Swap(ref int a, ref int b)

{

int temp = a;

a = b;

b = temp;

}

}

}

Листинг 3 – класс Utils

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Utils

{

class Program

{

static void Main(string[] args)

{

Console.Write("Введите первое число: ");

int x = int.Parse(Console.ReadLine());

Console.Write("Введите второе число: ");

int y = int.Parse(Console.ReadLine());

int greater = Utils.Greater(x, y);

Console.WriteLine("Большим из чисел {0} и {1} является число {2}", x, y, greater);

Console.WriteLine("До swap: \t{0} {1}", x, y);

Utils.Swap(ref x, ref y);

Console.WriteLine("После swap: \t{0} {1}", x, y);

}

}

}

Листинг 4 – класс Program

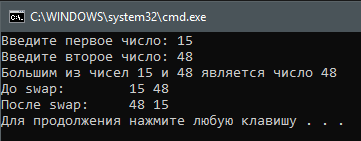


Рисунок 2 – результат работы программы

**Упражнение 3.**

1. В классе Utils был реализован метод для нахождения факториала числа. При этом результат возвращается в переменную-аргумент метода.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Utils

{

class Utils

{

public static int Greater(int a, int b)

{

if (a > b)

{

return a;

}

else

{

return b;

}

}

public static void Swap(ref int a, ref int b)

{

int temp = a;

a = b;

b = temp;

}

public static bool Factorial(int n, out int answer)

{

int k;

int f = 1;

bool ok = true;

try

{

checked

{

for (k = 2; k <= n; k++)

{

f \*= k;

}

}

}

catch (Exception)

{

f = 0;

ok = false;

}

answer = f;

return ok;

}

}

}

Листинг 5 – класс Utils

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Utils

{

class Program

{

static void Main(string[] args)

{

Console.Write("Введите первое число: ");

int x = int.Parse(Console.ReadLine());

Console.Write("Введите второе число: ");

int y = int.Parse(Console.ReadLine());

int greater = Utils.Greater(x, y);

Console.WriteLine("Большим из чисел {0} и {1} является число {2}", x, y, greater);

Console.WriteLine("До swap: \t{0} {1}", x, y);

Utils.Swap(ref x, ref y);

Console.WriteLine("После swap: \t{0} {1}", x, y);

int f;

bool ok;

Console.Write("Number for factorial: ");

x = int.Parse(Console.ReadLine());

ok = Utils.Factorial(x, out f);

if (ok)

{

Console.WriteLine("Factorial({0}) = {1}", x, f);

}

else

{

Console.WriteLine("Cannot compute this factorial");

}

}

}

}

Листинг 6 – класс Program

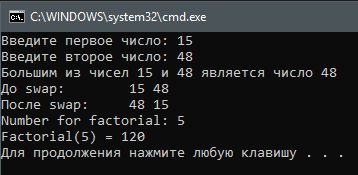


Рисунок 3 – результат работы программы

**Упражнение 4.**

1. Был создан новый проект Triangle. Данная программа позволит расчитывать площать треугольника.
2. Был создан класс Operation. В данном классе был реализован private метод для проверки существования треугольника. Метод был перегружен для работы с равносторонним треугольником.
3. Был реализован public метод GetArea для расчета площади треугольника. Данный метод так же был перегружен для работы с равносторонним треугольником.
4. В классе Program был реализован ввод сторон треугольника с клавиатуры.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Triangle

{

class Operation

{

public static double GetArea(int a, int b, int c)

{

if (IsTriangle(a, b, c))

{

double area;

double p = (double)(a + b + c) / 2;

area = Math.Sqrt(p \* (p - a) \* (p - b) \* (p - c));

return area;

}

else

{

return 0;

}

}

public static double GetArea(int a)

{

if (IsTriangle(a))

{

return (Math.Pow(a, 2) \* Math.Sqrt(3)) / 4;

}

else

{

return 0;

}

}

private static bool IsTriangle(int a, int b, int c)

{

if ((a > 0 && b > 0 && c > 0) && (a + b > c && b + c > a && c + a > b))

{

return true;

}

else

{

return false;

}

}

private static bool IsTriangle(int a)

{

if (a > 0)

{

return true;

}

else

{

return false;

}

}

}

}

Листинг 7 – класс Operation

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Triangle

{

class Program

{

static void Main(string[] args)

{

Console.Write("Введите тип треугольника (1 - равносторонний, 2 - обыкновенный): ");

int i = int.Parse(Console.ReadLine());

switch (i)

{

case 1:

Console.Write("Введите сторону треугольника: ");

int x = int.Parse(Console.ReadLine());

double area = Operation.GetArea(x);

if (area != 0)

{

Console.WriteLine("Площадь треугольника - {0}", area);

}

else

{

Console.WriteLine("Такого трекгольника не существует!");

}

break;

case 2:

Console.Write("a = ");

int a = int.Parse(Console.ReadLine());

Console.Write("b = ");

int b = int.Parse(Console.ReadLine());

Console.Write("c = ");

int c = int.Parse(Console.ReadLine());

area = Operation.GetArea(a, b, c);

if (area != 0)

{

Console.WriteLine("Площадь треугольника - {0}", area);

}

else

{

Console.WriteLine("Такого трекгольника не существует!");

}

break;

default:

Console.WriteLine("Ошибка ввода!");

break;

}

}

}

}

Листинг 8 – класс Program

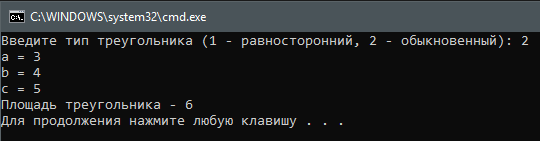


Рисунок 4 – результат работы программы

**Упражнение 5.**

1. Был создан новый проект QuadUr. Данная программа позволит находить корни квадратного уравнения.
2. Был реализован метод FindAnswer, который находит корни уравнения, при этом возвращает их через переменные-параметры метода, которые передавались по ссылкам.
3. В методе Main был реализован ввод коэффициентов с клавиатуры и вывод корней в консоль.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace QuadUr

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("ax^2 + bx + c = 0");

double x1 = 0, x2 = 0;

Console.Write("a = ");

int a = int.Parse(Console.ReadLine());

Console.Write("b = ");

int b = int.Parse(Console.ReadLine());

Console.Write("c = ");

int c = int.Parse(Console.ReadLine());

int countOfAnswers = FindAnswer(a, b, c, ref x1, ref x2);

if (countOfAnswers == -1)

{

Console.WriteLine("Корней уравнения с коэффициентами a = {0}, b = {1}, c = {2} нет.", a, b, c);

}

else if (countOfAnswers == 0)

{

Console.WriteLine("Корни уравнения с коэффициентами a = {0}, b = {1}, c = {2} равны x1 = x2 = {3}.", a, b, c, x1);

}

else

{

Console.WriteLine("Корни уравнения с коэффициентами a = {0}, b = {1}, c = {2} равны x1 = {3}, x2 = {4}.", a, b, c, x1, x2);

}

}

static int FindAnswer(int a, int b, int c, ref double x1, ref double x2)

{

double discr = Math.Pow(b, 2) - 4 \* a \* c;

if (discr < 0)

{

return -1;

}

else if (discr == 0)

{

x1 = (-b + Math.Sqrt(discr)) / (2 \* a);

x2 = x1;

return 0;

}

else

{

x1 = (-b + Math.Sqrt(discr)) / (2 \* a);

x2 = (-b - Math.Sqrt(discr)) / (2 \* a);

return 1;

}

}

}

}

Листинг 9 – исходный код программы QuadUr.cs

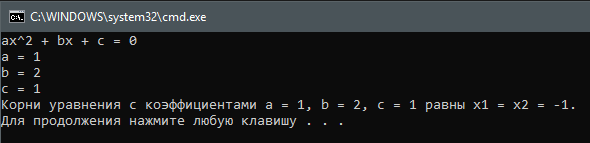


Рисунок 5 – результат работы программы

**Вывод:**

В ходе выполнения данной лабораторной работы были получены новые знания в языке C#. Были рассмотрены методы, возможность передавать переменные по ссылкам, возможность возвращать значения через аргументы. Была рассмотрена перегрузка методов. Полученные навыки были закреплены на практике написанием программ с использованием методов.

**Ответы на контрольные вопросы:**

1. Как передать переменную в метод по ссылке?

Использовать ref.

1. Как вернуть значение в аргумент метода?

Использовать out.