**Практическая работа №2**

**Оглавление**

[Практическая работа № 2. МЕТОДЫ 3](#_Toc161928746)

[Содержание пояснительной записки 9](#_Toc161928747)

[Используемое программное обеспечение 10](#_Toc161928748)

[Список литературы 11](#_Toc161928749)

Практическая работа № 2.  
МЕТОДЫ

**Цель практической работы**

Закрепление теоретических знаний по обработке текстов с помощью регулярных выражений.

**Постановка задачи**

**Метод –** блок кода, содержащий ряд инструкций.

[тип возвращаемого значения] [Имя]([аргументы])

{

// Тело метода

// Напримр:

float a = 10.2f;

float b = 3.4f;

return a+b;

}

При передаче аргументов в метод они копируются. Если это тип значения, то копируются сами значения, если тип ссылочный, то копируется ссылка.

**Пример.**

float getSum(float a, float b)

{

return a+b;

}

**void** – специальное слово, означающее, что метод **не возвращает значения**;

**return** – специальное слово, **завершающее** выполнение метода и **возвращающее** результат (при его наличии).

**Пример** создания методов.

var var1 = Convert.ToDouble(Console.ReadLine());

var var2 = Convert.ToDouble(Console.ReadLine());

WriteSum(var1, var2);

var var3 = GetSum(var1, var2);

void WriteSum(double a, double b)

{

Console.WriteLine(a + b);

}

double GetSum(double a, double b)

{

return a + b;

}

**Пример создания метода для решения квадратного уравнения:**

calculateSquareEquation();

Console.WriteLine();

calculateSquareEquation();

void calculateSquareEquation()

{

Console.WriteLine("Введеите через пробел коэффициенты A, B, C квадратного уравнения Ax^2+Bx+C=0");

var input = Console.ReadLine().Split(" ");

var A = double.Parse(input[0]);

var B = double.Parse(input[1]);

var C = double.Parse(input[2]);

var D = B \* B - 4 \* A \* C;

var x1 = (-B + Math.Sqrt(D)) / (2 \* A);

var x2 = (-B - Math.Sqrt(D)) / (2 \* A);

Console.WriteLine($"Корни уравнения {A}x^2+{B}x+{C}=0");

Console.WriteLine($"x1={x1}\tx2={x2}");

}

var a1 = 5;

var arr1 = new int[5];

Console.Write(a1);

Console.Write(arr1[0]);

a1 = Method1(a1);

Method2(arr1);

Console.Write(a1);

Console.Write(arr1[0]);

int Method1(int a1)

{

a1 = 10;

return 15;

}

void Method2(int[] arr)

{

arr[0] = 11;

}

**Рекурсия – вызов метода из самого метода.**

Console.WriteLine(Factorial(5));

double Factorial(double d)

{

if (d <= 1)

return 1;

return d \* Factorial(d - 1);

}

Факториал натурального числа n определяется как произведение всех натуральных чисел от 1 до n включительно.

Любой рекурсивный алгоритм можно переделать в не рекурсивный, например, с помощью бесконечных циклов или специальных коллекций данных.

**ref** – ключевое слово, указывает, что значение передается по ссылке.

Используется, когда необходимо изменить значение переменной внутри метода (если тип значение).

**out** – инициирует передачу аргументов по ссылке.

Используется, когда нужно вернуть несколько значений из метода.

var a1 = 5;

Console.Write($"{a1} ");

Method1(ref a1);

Method2(out int a2);

Console.Write($"{a1} ");

Console.Write($"{a2} ");

void Method1(ref int a)

{

a = 10;

}

void Method2(out int a)

{

a = 7;

}

**params** – ключевое слово, означающее, что метод принимает переменное число аргументов (одного типа). Используется для удобства написания кода.

Method1(1, 2, 3, 4, 5);

void Method1(params int[] arguments)

{

foreach (var argument in arguments)

{

Console.Write($"{argument} ");

}

}

**Задание на практическую работу**

1. Разработать консольное приложение для решения задачи согласно варианту.
2. Предусмотреть понятный интерфейс ввода и вывода данных на консоль.
3. Предусмотреть выбор файла с данными.
4. Защита работы включает демонстрацию работы программы.

**Варианты заданий на практическую работу**

Задания берем с сайта «Школа программиста» <https://acmp.ru/>

|  |  |
| --- | --- |
| Номер варианта | Ссылка на задание |
| 1 | 778 <https://acmp.ru/index.asp?main=task&id_task=778> |
| 2 | 817 <https://acmp.ru/index.asp?main=task&id_task=817> |
| 3 | 850 <https://acmp.ru/index.asp?main=task&id_task=850> |
| 4 | 294 <https://acmp.ru/index.asp?main=task&id_task=294> |
| 5 | 86 <https://acmp.ru/index.asp?main=task&id_task=86> |
| 6 | 327 <https://acmp.ru/index.asp?main=task&id_task=327> |
| 7 | 791 <https://acmp.ru/index.asp?main=task&id_task=791> |
| 8 | 264 <https://acmp.ru/index.asp?main=task&id_task=264> |
| 9 | 148 <https://acmp.ru/index.asp?main=task&id_task=148> |
| 10 | 678 <https://acmp.ru/index.asp?main=task&id_task=678> |
| 11 | 263 <https://acmp.ru/index.asp?main=task&id_task=263> |
| 12 | 894 <https://acmp.ru/index.asp?main=task&id_task=894> |
| 13 | 147 <https://acmp.ru/index.asp?main=task&id_task=147> |
| 14 | 331 <https://acmp.ru/index.asp?main=task&id_task=331> |
| 15 | 43 <https://acmp.ru/index.asp?main=task&id_task=43> |
| 16 | 935 <https://acmp.ru/index.asp?main=task&id_task=935> |
| 17 | 62 <https://acmp.ru/index.asp?main=task&id_task=62> |
| 18 | 297 <https://acmp.ru/index.asp?main=solution&id_task=297> |
| 19 | 943 <https://acmp.ru/index.asp?main=task&id_task=943> |
| 20 | 844 <https://acmp.ru/index.asp?main=task&id_task=844> |
| 21 | 496 <https://acmp.ru/index.asp?main=task&id_task=496> |
| 22 | 929 <https://acmp.ru/index.asp?main=task&id_task=929> |
| 23 | 10 <https://acmp.ru/index.asp?main=task&id_task=10> |
| 24 | 284 <https://acmp.ru/index.asp?main=task&id_task=284> |
| 25 | 637 <https://acmp.ru/index.asp?main=task&id_task=637> |
| 26 | 131 <https://acmp.ru/index.asp?main=task&id_task=131> |
| 27 | 952 <https://acmp.ru/index.asp?main=task&id_task=952> |
| 28 | 26 <https://acmp.ru/index.asp?main=task&id_task=26> |
| 29 | 822 <https://acmp.ru/index.asp?main=task&id_task=822> |
| 30 | 63 <https://acmp.ru/index.asp?main=task&id_task=63> |
| 31 | 22 <https://acmp.ru/index.asp?main=task&id_task=22> |

# Содержание пояснительной записки

1. Постановка задачи. Приводится теоретический материал, использованный при написании приложения.

2. Формулировка задания и вариант. Приводится задание на лабораторную работу и вариант этого задания.

3. Описание выполняемых действий. Необходимо привести описание последовательности разработки программы, реализации используемых методов, алгоритмов, блок-схем.

4. Анализ результатов. Привести анализ входных и выходных данных. Показать результаты выполнения программного кода. Предоставить скриншоты обработки тестовых примеров. Сделать выводы.

5. Листинг программы. Привести листинг разработанного программного кода, содержание файлов входных и выходных данных.

# Используемое программное обеспечение

1. Среда программирования MS Visual Studio Community 2022 (Свободно распространяемое программное обеспечение (в учебных целях));
2. Microsoft Office Standard 2007 (Open License: 42267924);
3. Open Office (Свободно распространяемое программное обеспечение).
4. Браузер (Свободно распространяемое программное обеспечение).

# Список литературы

* + - 1. Мейер Б. Объектно-ориентированное программирование и программная инженерия [Электронный ресурс]/ Мейер Б. – Электрон. текстовые данные. – М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. – 285 c.
      2. Биллиг, В. A. Основы объектного программирования на С# (C# 3.0, Visual Studio 2008) [Электронный ресурс] : учебное пособие / В. A. Биллиг. — Электрон. текстовые данные. — Москва, Саратов : Интернет-Университет Информационных Технологий (ИНТУИТ), Вузовское образование, 2017. — 583 c. — 978-5-4487-0145-0. — Режим доступа: http://www.iprbookshop.ru/72339.html
      3. Павловская, Т. А. Программирование на языке высокого уровня C# [Электронный ресурс] / Т. А. Павловская. — 2-е изд. — Электрон. текстовые данные. — М. : Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. — 245 c. — 2227-8397. — Режим доступа: http://www.iprbookshop.ru/73713.html
      4. Агапов, В. П. Основы программирования на языке С# [Электронный ресурс] : учебное пособие / В. П. Агапов. — Электрон. текстовые данные. — М. : Московский государственный строительный университет, ЭБС АСВ, 2012. — 128 c. — 978-5-7264-0576-6. — Режим доступа: http://www.iprbookshop.ru/16366.html
      5. Медведев, М. А. Программирование на СИ# [Электронный ресурс] : учебное пособие / М. А. Медведев, А. Н. Медведев ; под ред. А. В. Присяжный. — Электрон. текстовые данные. — Екатеринбург : Уральский федеральный университет, ЭБС АСВ, 2015. — 64 c. — 978-5-7996-1561-1. — Режим доступа: http://www.iprbookshop.ru/69667.html
      6. Казанский А.А. Объектно-ориентированное программирование на языке Microsoft Visual С# в среде разработки Microsoft Visual Studio 2008 и .NET Framework. 4.3 [Электронный ресурс]: учебное пособие и практикум/ Казанский А.А.— Электрон. текстовые данные.— М.: Московский государственный строительный университет, ЭБС АСВ, 2011.— 180 c
      7. Уйманова Н.А. Основы объектно-ориентированного программирования [Электронный ресурс]: практикум/ Уйманова Н.А., Таспаева М.Г.— Электрон. текстовые данные.— Оренбург: Оренбургский государственный университет, ЭБС АСВ, 2017.— 156 c.— Режим доступа: http://www.iprbookshop.ru/78808.html.— ЭБС «IPRbooks»
      8. Новиков П.В. Объектно-ориентированное программирование [Электронный ресурс]: учебно-методическое пособие к лабораторным работам/ Новиков П.В.— Электрон. текстовые данные.— Саратов: Вузовское образование, 2017.— 124 c.— Режим доступа: http://www.iprbookshop.ru/64650.html.— ЭБС «IPRbooks»