

**УО «МОГИЛЕВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ А.А. КУЛЕШОВА»
СОЦИАЛЬНО-ГУМАНИТАРНЫЙ КОЛЛЕДЖ**



**Дисциплина
«Конструирование программ и языки программирования»**

**Разработка линейных программ
(2 часа)**

Методические рекомендации к лабораторной работе № 2

Могилев 2018

Понятия «Линейная программа», «Типы данных», «Математические функции». Методические указания по лабораторной работе № 2 по дисциплине «Конструирование программ и языки программирования». Для учащихся 3 курса очной формы обучения специальности 2–40 01 01 «Программное обеспечение информационных технологий».

Оглавление

1 Цель работы	4
2 Краткие теоретические сведения	5
2.1 Типы данных.....	5
2.2 Переменные	6
2.3 Математические функции	7
2.4 Организация ввода-вывода данных.....	9
2.4.1 Ввод данных.....	9
2.4.2 Вывод данных	10
2.5 Использование управляющих последовательностей.....	10
3. Задания	13
4 Контрольные вопросы	17

1 Цель работы

выработать умение разрабатывать линейные программы.

2 Краткие теоретические сведения

2.1 Типы данных.

Язык C# имеет набор встроенных типов, которые рассматриваются как псевдонимы типов в пространстве имен System. Например, тип `string` – это псевдоним типа `System.String`, а тип `int` – псевдоним типа `System.Int32`. Все встроенные типы подразделены на группы: целочисленные типы; вещественные типы; логический тип; символьные типы; объектный тип (`object`). Описание типов приведено в таблице 1.

Иерархия классов NET Framework имеет один общий корень - класс `System.Object`. Если `object` является базовым классом для всех остальных типов и упаковка значений простых типов происходит автоматически, то класс `object` можно вполне использовать в качестве "универсального" типа данных.

Таблица 1. Встроенные типы C#

Название	Ключевое слово	Тип .NET	Диапазон значений	Описание	Размер, битов
Логический тип	<code>bool</code>	<code>Boolean</code>	<code>true, false</code>		
Целые типы	<code>sbyte</code>	<code>SByte</code>	От -128 до 127	Со знаком	8
	<code>byte</code>	<code>Byte</code>	От 0 до 255	Без знака	8
	<code>short</code>	<code>Int16</code>	От -32768 до 32767	Со знаком	16
	<code>ushort</code>	<code>UInt16</code>	От 0 до 65535	Без знака	16
	<code>int</code>	<code>Int32</code>	От -2×10^9 до 2×10^9	Со знаком	32
	<code>uint</code>	<code>UInt32</code>	От 0 до 4×10^9	Без знака	32
	<code>long</code>	<code>Int64</code>	От -9×10^{18} до 9×10^{18}	Со знаком	64
	<code>ulong</code>	<code>UInt64</code>	От 0 до 18×10^{18}	Без знака	64
Символьный тип	<code>char</code>	<code>Char</code>	От U+0000 до U+ffff	Unicode-символ	16
Вещественные	<code>float</code>	<code>Single</code>	От 1.5×10^{-45} до 3.4×10^{38}	7 цифр	32
	<code>double</code>	<code>Double</code>	От 5.0×10^{-324} до 1.7×10^{308}	15–16 цифр	64
Финансовый тип	<code>decimal</code>	<code>Decimal</code>	От 1.0×10^{-28} до 7.9×10^{28}	28–29 цифр	128
Строковый тип	<code>string</code>	<code>String</code>	Длина ограничена объемом доступной памяти	Строка из Unicode-символов	
Тип <code>object</code>	<code>object</code>	<code>Object</code>	Можно хранить все, что угодно	Всеобщий предок	

При создании переменной размерного типа под нее в стеке выделяется определенный объем памяти, соответствующий типу этой переменной. При передаче такой переменной в качестве параметра выполняется передача значения, а не ссылки на него. Значение размерного типа не может быть равным `null`. К размерным типам, например, относятся целочисленные и вещественные типы, структуры.

При создании переменной ссылочного типа память под созданный объект выделяется в другой области памяти, называемой кучей. Ссылка всегда указывает на объект заданного типа.

2.2 Переменные

Переменная – именованная область памяти, для хранения данных определенного типа. При выполнении программы значение переменной величины можно изменять. Все переменные должны быть описаны явно, при описании переменной задается ее значение и тип. При объявлении переменной может быть задано начальное значение.

Имя переменной может содержать буквы, цифры и символ подчеркивания. Прописные и строчные буквы различаются. Например, переменные Long, LONG, long - три разных переменные.

Имя переменной может начинаться с буквы или знака подчеркивания, но не цифры. Имя переменной не должно совпадать с ключевыми словами. Не рекомендуется начинать имя с двух подчеркиваний (такие имена зарезервированы для служебного использования).

Правильные имена переменных: MaxLen, iMaxLen, Max_Len

Неправильные имена переменных: 2Len, Le#

Примеры описания переменных:

```
int a = -14; // числовая целая 32 бита
float c = -0.00151f; // числовая вещественная 32 бита
double I = 123.56789; // числовая вещественная 64 бита
bool l = false; // логическая 16 бит
string name = "Petrov"; //строковая
```

Выражение - состоит из одного или более операндов (которые могут быть переменными, константами, функциями или символьными значениями), знаков операций и круглых скобок.

Примеры выражений:

```
2 * 2 + 1 полученное значение 5
1 / 2 - 3 полученное значение -3
1.0 / 2 - 3 полученное значение -2,5
```

Присвоение значения переменной представляет оператор присваивания (знаки основных операций приведены в таблице 2):

$y = 2*x*x + 3*x - 1.$

Таблица 2. Знаки операций

Знак операции	Название
+	Сложение
-	Вычитание
*	Умножение
/	Деление
%	Остаток от деления

Если в арифметических выражениях используются целые числа, то результатом вычислений будет целое число, и любой остаток от деления будет отброшен. Для получения остатка можно использовать соответствующую операцию %, например $10 \% 3$ возвращает

остаток от целочисленного деления, равный 1.

Когда в арифметических выражениях используются числа с плавающей точкой, то результатом деления $10f / 3f$ будет число 3,333333.

Приоритет и ассоциативность операторов C# влияют на группировку и оценку операндов в выражениях. Приоритет оператора имеет смысл только в том случае, если присутствуют другие операторы с более высоким или более низким приоритетом. Сначала оцениваются выражения с операторами с более высоким приоритетом. Приоритет также можно описать словом «привязка». Говорят, что операторы с более высоким приоритетом имеют более жесткую привязку (рисунок 1)

Symbol1	Type of Operation	Associativity
[] () . -> postfix ++ and postfix --	Expression	Left to right
prefix ++ and prefix -- sizeof & * + - ~ !	Unary	Right to left
typecasts	Unary	Right to left
* / %	Multiplicative	Left to right
+ -	Additive	Left to right
<< >>	Bitwise shift	Left to right
< > <= >=	Relational	Left to right
== !=	Equality	Left to right
&	Bitwise-AND	Left to right
^	Bitwise-exclusive-OR	Left to right
	Bitwise-inclusive-OR	Left to right
&&	Logical-AND	Left to right
||	Logical-OR	Left to right
? :	Conditional-expression	Right to left
= *= /= %=	Simple and compound assignment2	Right to left
+= -= <= >= &=		
^= =		
,	Sequential evaluation	Left to right

Рисунок 1. Приоритет операций

2.3 Математические функции

C# содержит большое количество встроенных математических функций, которые реализованы в классе `Math` пространства имен `System`.

Рассмотрим краткое описание некоторых математических функций, подробнее с ни-

ми можно познакомиться в справочной системе VS или технической документации. Особое внимание следует обратить на типы операндов и результатов, т. к. каждая функция может иметь несколько перегруженных версий.

Замечание. Использование нескольких функций с одним и тем же именем, но с различными типами параметров, называется перегрузкой функции. Например, функция `Math.Abs()`, вычисляющая модуль числа, имеет 7 перегруженных версий: `double Math.Abs(double x)`, `float Math.Abs(float x)`, `int Math.Abs(int x)`, и т. д. (таблица 3)

Таблица 3 Математические функции

№	Название	Описание
1.	<code>Math.Abs(выражение)</code>	Модуль
2.	<code>Math.Ceiling(выражение)</code>	Округление до большего целого
3.	<code>Math.Cos(выражение)</code>	Косинус
4.	<code>Math.E</code>	Число e
5.	<code>Math.Exp(выражение)</code>	Экспонента
6.	<code>Math.Floor(выражение)</code>	Округление до меньшего целого
7.	<code>Math.Log(выражение)</code>	Натуральный логарифм
8.	<code>Math.Log10(выражение)</code>	Десятичный логарифм
9.	<code>Math.Max(выражение1, выражение2)</code>	Максимум из двух значений
10.	<code>Math.Min(выражение1, выражение2)</code>	Минимум из двух значений
11.	<code>Math.PI</code>	Число пи
12.	<code>Math.Pow(выражение1, выражение2)</code>	Возведение в степень
13.	<code>Math.Round(выражение)</code> <code>Math.Round(выражение, число)</code>	Простое округление Округление до заданного числа
14.	<code>Math.Sign(выражение)</code>	Знак числа
15.	<code>Math.Sin(выражение)</code>	Синус
16.	<code>Math.Sqrt(выражение)</code>	Корень квадратный

Пример 1. Вычислить значения функции $Y = \frac{\cos \pi x}{1+x^2}$ при $x = 2,5$

```
using System;
namespace Lab_2
{
    class Example2 // начало описание класса Example2
    {
        static void Main(string[] args)
        {
```

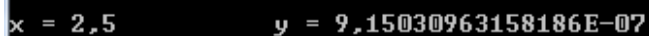


```

        double p = 3.14159;
        double x = 2.5;
        double y = Math.Cos(p * x) / (1 + x * x);
        Console.WriteLine();
        Console.WriteLine("x = {0} \t y = {1}", x, y);
    }
}

```

Эта программа выводит следующее окно с результатом:



```

x = 2,5      y = 9,15030963158186E-07

```

Замечание. Функция выводит на экран пустую строку. Это сделано для более комфортной работы.

2.4 Организация ввода-вывода данных.

Программа при вводе данных и выводе результатов взаимодействует с внешними устройствами. Совокупность стандартных устройств ввода (клавиатура) и вывода (экран) называется консолью. В языке C# нет операторов ввода и вывода. Вместо них для обмена данными с внешними устройствами используются специальные объекты. В частности, для работы с консолью используется стандартный класс `Console`, определенный в пространстве имен `System`.

2.4.1 Ввод данных

Для ввода данных обычно используется метод `ReadLine`, реализованный в классе `Console`. Особенностью данного метода является то, что в качестве результата он возвращает строку (`string`).

```

static void Main(string[] args)
{
    string s = Console.ReadLine();
    Console.WriteLine(s);
    Console.ReadKey();
}

```

Для того чтобы получить числовое значение необходимо воспользоваться преобразованием данных

```

static void Main(string[] args)
{
    string s = Console.ReadLine();
    int x = int.Parse(s); //преобразование строки в число
    Console.WriteLine(x);
    Console.ReadKey();
}

```

```
}
```

Или сокращенный вариант:

```
static void Main(string[] args)
{
    //преобразование строки в число
    int x = int.Parse(Console.ReadLine());
    Console.WriteLine(x);
    Console.ReadKey();
}
```

Для преобразования строкового представления целого числа в тип `int` мы используем метод `int.Parse()`, который реализован для всех числовых типов данных. Таким образом, если нам потребуется преобразовать строковое представление в вещественное, мы можем воспользоваться методом `float.Parse()` или `double.Parse()`. В случае, если соответствующее преобразование выполнить невозможно, то выполнение программы прерывается и генерируется исключение `System.FormatException` (входная строка имела неверный формат).

2.4.2 Вывод данных

В приведенных выше примерах мы уже рассматривали метод `WriteLine`, реализованный в классе `Console`, который позволяет организовывать вывод данных на экран. Однако существует несколько способов применения данного метода (таблица 4):

Таблица 4. Способы вывода данных

<code>Console.WriteLine(x);</code>	На экран выводится значение идентификатора <code>x</code>
<code>Console.WriteLine("x="+x+"y="+y);</code>	На экран выводится строка, образованная последовательным слиянием строки <code>"x="</code> и значения <code>x</code> , <code>"y="</code> и значения <code>y</code> ,
<code>Console.WriteLine("x={0} y={1}", x, y);</code>	На экран выводится строка, формат которой задан первым аргументом метода, при этом вместо параметра <code>{0}</code> выводится значение <code>x</code> , а <code>{1}</code> вместо – значение <code>y</code> .

Если использовать при выводе вместо метода `WriteLine` метод `Write`, вывод будет выполняться без перевода строки.

2.5 Использование управляющих последовательностей

Управляющей последовательностью называют определенный символ, предваряемый обратной косой чертой. Данная совокупность символов интерпретируется как одиночный символ и используется для представления кодов символов, не имеющих графического обозначения (например, символа перевода курсора на новую строку) или символов, имеющих специальное обозначение в символьных и строковых константах (например, апо-

строф). Рассмотрим управляющие символы (таблица 5):

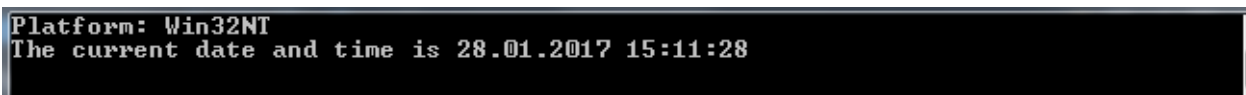
Таблица 5. Управляющие символы

Вид	Наименование	Вид	Наименование
\a	Звуковой сигнал	\t	Горизонтальная табуляция
\b	Возврат на шаг назад	\v	Вертикальная табуляция
\f	Перевод страницы	\\	Обратная косая черта
\n	Перевод строки	\'	Апостроф
\r	Возврат каретки	\"	Кавычки

Пример 3. Вывести сообщение о версии текущей ОС, текущую дату и время.

```
using System;
using System.IO;
using System.Text;

namespace Lab_2
{
    class Program
    {
        static void Main(string[] args)
        {
            //вывести версию ОС
            OperatingSystem os = Environment.OSVersion;
            Console.WriteLine("Platform: {0}", os.Platform);
            Console.WriteLine("The current date and time is " +
DateTime.Now);
            // дата и время
            Console.ReadKey();
        }
    }
}
```



```
Platform: Win32NT
The current date and time is 28.01.2017 15:11:28
```

Пример 4. Использование консольного ввода для вычисления значения функции

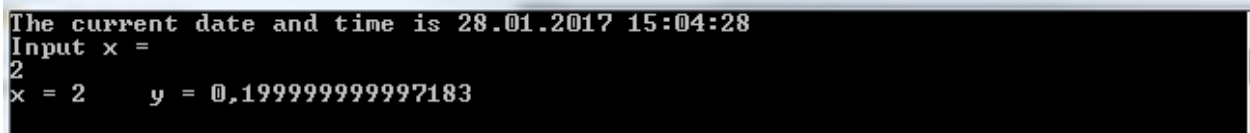
$$Y = \frac{\cos \pi x}{1 + x^2}$$

```
using System;
using System.IO;
using System.Text;
namespace Lab_2
```

```

{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("The current date and time is " +
DateTime.Now);
            double pi = 3.14159;
            Console.WriteLine("Input x =\r");
            double x = Convert.ToDouble(Console.ReadLine());
            double y = Math.Cos(pi * x) / (1 + x * x);
            Console.WriteLine("x = {0} \t y = {1}", x, y);
            Console.ReadKey();
        }
    }
}

```



```

The current date and time is 28.01.2017 15:04:28
Input x =
2
x = 2    y = 0,199999999997183

```

3. Задания

1. Напишите программу для расчета функции, согласно номеру варианта:

$$1) y = \frac{a^2 \cdot \ln(2/a)}{\sin(x + \sqrt{x + b^2})} + e^{x/z}$$

при $a = 1,678$; $b = 19,792$; $x = 3,67$; $z = 2$.

$$2) d = (a^2 + \cos b^2) \cdot \sin^2 x + y/\sqrt{a}$$

при $a = 5,234$; $b = 2,94$; $y = -18,32$; $x = 65^\circ$.

$$3) S = (x/a + b) \cdot e^{-x/a+1} \cdot \frac{\sin(x+y)}{\cos(a+1)}$$

при $a = 2,961$; $b = 1,96$; $x = 11,271$; $y = 6,718$.

$$4) y = \sqrt{a\sqrt{\pi} + e^{bx}} \cdot m \ln\left(a + \frac{x}{\sin x}\right)$$

при $a = 1,297$; $b = 0,5054$; $x = 2,1212$; $m = 3$.

$$5) S = \frac{\cos^2 z - ax^2 \sqrt{b}}{e^{b+ax} + \sqrt{2\pi y}}$$

при $a = 1,1111$; $b = 2,2222$; $x = 3,45$; $y = 1,234$; $z = 14,8$.

$$6) Z = \frac{\sqrt{a + \cos^2 x}}{b + y \sin x} \cdot \ln q$$

при $a = 1$; $b = 2$; $q = 3$; $x = 137^\circ$; $y = 1,158$.

$$7) y = \frac{ax - e^{-b/x}}{z \cdot \sqrt{|\sin(z/t + t)|}}$$

при $a = 1,957$; $b = 9$; $t = 6$; $x = 8,33$; $z = 5,777$.

$$8) y = e^{x/\sqrt{b}} \cos\left(\sqrt{x}/b + \pi/2\right) \cdot \sqrt{2\pi x + x/a}$$

при $a = 1,79$; $b = 2$; $x = 27$.

$$9) y = \frac{\sqrt{2\pi x} \cdot x^{x+1} \cdot e^{-x}}{\cos x/p + a/b}$$

при $a = 1,234$; $b = 0,4321$; $p = 2$; $x = 0,378$.

$$10) Z = \frac{x+b}{y+b^2} + \frac{a \ln x + y}{e^x + y}$$

при $a = 19,78$; $b = 1$; $x = 3,413$; $y = 1,789$.

$$11) y = \frac{1}{\sin a \sqrt[3]{n+3} p^2} - \frac{5bc+d}{\cos x}$$

при $a = 173,5^\circ$; $b = 0,8$; $c = 5,0839$; $d = -3,39$; $m = 4$; $p = -2$; $x = 1$.

$$12) S = \frac{t+py^2}{t^2+p^2} \cdot \sin x \cdot e^{pt^2/\sqrt{2}}$$

при $x = 119^\circ$; $y = 2,345$; $t = 3,788$; $p = 0,198$.

$$13) y = \frac{\sqrt{3ab}}{m-i} + \frac{m}{5} \cdot \cos x - \ln z$$

при $a = 1,645$; $b = 0,069$; $I = 2$; $m = 4$; $x = 18^\circ$; $z = 3$.

$$14) y = \ln\left(\frac{m+n}{7}\right) + \sin ax - 1,24 \sqrt[4]{\frac{2,5+ac}{|\cos b|}}$$

при $a = 4$; $b = 34^\circ$; $c = 1,097$; $m = 11$; $n = -3$; $x = 1$.

$$15) y = \frac{i^3-1}{k-4} \sqrt{2ab + \sin x + \cos z}$$

при $a = 1,835$; $b = 3$; $I = 3$; $k = 7$; $x = 0,2$; $z = 20^\circ$.

$$16) Z = \frac{a^2 + b^2 x + y^2 \sqrt{x+y}}{a+b}$$

при $a = -1,791$; $b = 2,796$; $x = 0,798$; $y = 1,678$.

$$17) y = \sin \sqrt[4]{m+2/x+k/n} + \ln \sqrt[3]{k/2+x^2}$$

при $k = 17$; $m = 6$; $n = 3$; $x = 1,726$.

$$18) U = a \sqrt{x^2+y^2} + b \ln x/y \cdot e^{\sqrt{x+y}}$$

при $a = -6,918$; $b = 3,961$; $x = 1,892$; $y = 0,3671$.

$$19) y = \frac{\sqrt{a\sqrt{\pi} + e^{bx}}}{bx^2 + az - \cos^{bx}}$$

при $a = 1,297$; $b = 0,5054$; $x = 2,1212$; $z = 0,5$.

$$20) Z = \frac{\cos^2 b + e^{-x/a} \sqrt{x^2+y^2}}{\sqrt{a} + \sqrt{b} + \sqrt{x} + \sqrt{y}}$$

при $a = 5,65$; $b = 1,472$; $x = 1,87$; $y = 4,17$.

2. Написать программу, реализующую функцию согласно варианту задания. Исходные данные вводятся с клавиатуры.

Вар.	Функция	x	y
1	$A = \sqrt{\ln(\frac{4}{3} + x) + \frac{9}{7}} - e^{-\sin(1,3x-0,7)}$	0,31 2,5	-0,0049
2	$B = (x + \frac{7}{6})^{\frac{4}{3}} + \sin e^x + \arcsin(\cos px)$	-0,75 1,2	-0,018
3	$C = 3,7\sqrt{5-x}\cos(3,5-x) - \sqrt[3]{(5-x)^3}$	2,23 3,2	-0,018
4	$D = -e^{-\cos\sqrt{x+\frac{5}{3}}} - 1,7\arctg(\frac{x}{5} - \frac{3}{4})\sin 1,7x$	-0,35 1,5	-1,318
5	$E = 6,3\sin(1,3x - \frac{p}{3}) - x + \sqrt{x + \frac{9}{4}} + (x + \frac{7}{3})^{\frac{2}{3}}$	0,40 1,5	0,016
6	$F = \cos 1,5x - e^{\sin(x+\frac{4}{3})} + \sqrt{x + \frac{7}{6}}$	2,26 1,2	0,235
7	$G = \frac{5}{3} - \arctg\sqrt{2 - \cos 2x} - e^{-\frac{x}{5}}$	2,09 1,7	0,920
8	$H = \sin \ln(x+2) - \cos(\pi \ln(x + \frac{5}{3})) + \frac{x}{5}$	-0,26 0,25	-0,0049
9	$I = 4\sin(15e^{\frac{x}{8}} + 10,2) - 9\cos e^{-x} + \sqrt{x + \frac{5}{3}}$	-0,61 0,5	-0,012
10	$J = e^{\frac{4x}{5}} + 2\sin(7\ln(x + \frac{5}{3})) - p$	0,97 -0,5	-0,0024

11	$K = 1,3e^{-\frac{x}{2}} + \left \cos\left(\frac{2\pi x}{3} - 1,4\right) \right - \frac{6}{11}$	2,81 1,25	0,253
12	$L = p + \ln \left \frac{4}{7} - \frac{\sin \operatorname{arctg} x}{2} \right $	2,03 1,7	1,043
13	$M = e^{-\frac{x}{p}} + \frac{4}{3} \arcsin \cos x$	1,97 0,7	0,0017
14	$F = \cos 1,5x - e^{\sin(x + \frac{5}{3})} + \sqrt{x + \frac{7}{6}}$	0,96 1,23	-0,528
15	$O = \arccos \sin(3x + 1,3) - xe^{\operatorname{arctg} x} + 0,7$	1,32 -0,5	0,307
16	$P = 1,3x - 2,5 \sin\left(5\sqrt{\frac{4}{3} + \operatorname{arctg} x} - 0,7\right)$	-0,71 0,7	0,0252
17	$Q = e^{-\frac{x}{2}} \cos(2x - 0,3) + \frac{x^2}{2,7 + x}$	-0,73 1,53	-4,197
18	$R = \sqrt{e^{-\frac{x}{2}} - 0,1} - x \cos(3x - 1,5)$	2,15 1,2	-1,485
19	$S = 5 \sin(x - 0,3) - \sqrt{2 + e^{-x} - 0,1x^2}$	0,62 1,1	-0,0082
20	$T = 20,7 + \left(\sin^2(1,2x) - \arccos \frac{x}{8}\right) \cdot e^{1,5x}$	2,07 1,35	-0,1699

4 Контрольные вопросы

1. Как записываются операторы начала и конца программы?
2. Какой метод позволяет организовывать вывод данных на экран? Какие способы существуют?
3. Какие типы данных существуют в C#?
4. Что такое переменная?
5. Для чего необходим класс `Math` пространства имен `System`? Перечислите некоторые математические функции(минимум 4).
6. Каким преобразованием необходимо воспользоваться , для того чтобы получить числовое значение?
7. Что называется управляющей последовательностью? Какие существуют управляющие последовательности?