**“HELLO, WORLD!” APP**

// Импорт сторонних пространств имён

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

/// <summary>

/// Пространство имён

/// Создание собственной библиотеки

/// </summary>

namespace HelloWorldConsoleApp

{

/// <summary>

/// Главная составляющая единица приложения - класс

/// Подробности в Теме 7. Структуры и введение в ООП

/// </summary>

class Program

{

/// <summary>

/// Код, который выполняется в первую очередь

/// </summary>

/// <param name="args">Параметры запуска</param>

static void Main(string[] args)

{

Console.WriteLine("Привет, мир!"); // Вывод текста

// в консоль

Console.ReadKey(); // Ожидание от пользователя

// нажатия какой-либокнопки

/\*

\* Умение писать программы тренирует ваш разум.

\* Это поможет вам ясно мыслить во всех областях.

\* © Билл Гейтс, основатель Microsoft

\*/

}

}

}

**2.1. VARIABLE**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Example\_001\_Variable

{

class Program

{

static void Main(string[] args)

{

// Переменная - это "контейнер" для хранения данных

// ────────────────┐

// ┌──────────┐ │

// │ 6174 │ Значение │ Переменная

// └──────────┘ │

// box Идентификатор |

// ────────────────┘

// Переменные используются для хранения данных в памяти во

// время выполнения программы

// Переменная объявляется следующим образом:

// Ключевое слово var за которым следует идетификатор переменной,

// далее оператор присваивания и значение переменной,

// инструкция заканчивается точкой с запятой

//

// Примеры

var a = 28;

var b = 84.5;

var c = "Первая программа написанная на языке Visual C#";

// Замечание

// Имя (идентификатор переменной) может содержать

// латинские строчные и прописные буквы

// символ нижнего подчеркивание - "\_"

// цифры,

// ! но цифра не может быть на первом месте

// Пример того, как не нужно писать код

var g = 1;

var ab = 2;

var qwerty = "Я программист!";

var asdf1234 = 20.19;

var zx\_cv = 28;

var \_\_\_gg777\_oy19\_\_\_ = "Сегодня отличный день";

//var 799abcde = "упс..."; //- ошибка

// ! Важно

// Привыкать к осмысленному именованию переменных

var age = 28;

var weight = 84.5;

var title = "Первая программа написанная на языке Visual C#";

var month = "March";

var schoolName = "SkillBox";

var csharpCreater = "Anders Hejlsberg";

Console.ReadKey();

}

}

}

**2.2. DATA TYPES**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Example\_002\_DataTypes

{

class Program

{

static void Main(string[] args)

{

// В зависимости от того, какие данные нужно хранить, есть несколько видов "контейнеров"

// или типов данных.

// В C# есть

/// Числовые типы

/// Целые числа со знаком: -352, -17, 0, 4

/// Целые числа без знака: 0, 1, 2, 2000, 7985

/// Вещественные числа: -9.03, 3.2019, 3.1415

//

/// Строковый

/// Любая строка записывается в двойных ковычках - "пример"

///

//

/// Символьный

/// Переменная символьного типа записывается в одинарных ковычках - ''

//

/// Логический

/// Переменная логического типа принимает два возможных значения

/// истина - true

/// ложь - false

}

}

}

**2.3. INTEGER TYPES**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Example\_003\_IntegerTypes

{

class Program

{

static void Main(string[] args)

{

// Существует четыре типа целых чисел со знаком,

// которые можно использовать в зависимости от того,

// насколько большое(или маленькое) число нужно хранить

// Целые числа со знаком

sbyte int8 = 2; // от -128 дo +127

short int16 = 1; // от -32768 дo +32767

int int32 = 0; // от -2^31 дo +2^31-1 >>> от -2 147 483 648 до +2 147 483 647

long int64 = -1; // от -2^63 дo +2^63-1 >>> от -9 223 372 036 854 775 808 до +9 223 372 036 854 775 807

// Типы без знака можно использовать, если нужно хранить только положительные значения

// Целые числа без знака

byte uInt8 = 0; // от 0 дo 255

ushort uInt16 = 1; // от 0 дo 65535

uint uInt32 = 2; // от 0 дo 2^32-1 >>> от 0 до 4 294 967 296

ulong uInt64 = 3; // от 0 дo 2^64-1 >>> от 0 до 18 446 744 073 709 551 616

// Замечание 1 Работает с C# 7.0

// В дополнение к стандартной десятичной записи, целые числа также могут быть

// записаны с использованием шестнадцатеричная и двоичная формы записи.

// Для шестнадцатеричной нужно использовать префикс 0x

// Для двоичной нужно использовать префикс 0b

// Пример

int hexNumber = 0x7C6; // 0x7C6 - это 1990 в десятичной форме записи

Console.WriteLine(hexNumber); // 1990

int binNumber = 0b11100; // 0b11100 - это 28 в десятичной форме записи

Console.WriteLine(binNumber); // 28

// Замечание 2 Работает с C# 7.0

// Добавлено использование разделителей цифр (\_) для улучшения читабельности длинных чисел.

// Этот разделитель цифр может появляться в любом месте числа

int number = 1000000000;

int \_number = 1\_000\_000\_\_\_\_\_\_\_000;

}

}

}

**2.4. FLOAT POINT TYPES**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Example\_004\_FloatPointTypes

{

class Program

{

static void Main(string[] args)

{

// Типы с плавающей точкой могут хранить действительные числа с разной точностью.

// Постоянные числа с плавающей точкой в C# всегда сохраняются как переменные типа double,

// Для использования типа float требуется в конце добавить символ F или f

// Аналогично для типа decimal требуется в конце добавить символ M или m

// Аналогично для типа double добавление в конце символа D или d необязательно

// Пример

float floatVariable = 3.14F;

double doubleVariable = 3.14;

decimal decimalVariable = 3.14M;

// Тип float принимает значения От ±1,5 x 10^(−45) до ±3,4 x 10^38 точность - 7 цифр

floatVariable = 1.12345678910f;

Console.WriteLine(floatVariable); // 1.1234567

// Тип double принимает значения От от ±5,0 × 10^(−324) до ±1,7 × 10^308 точность - 15 цифр doubleVariable = 1.1234567898765431d;

doubleVariable = 1.1234567898765431d;

Console.WriteLine(doubleVariable); // 1,12345678987654

// Тип decimal принимает значения от ±1,0 x 10^(-28) до ±7,9228 x 10^28 точность - 28 цифр

decimalVariable = 1.12345678912345678901234567891234m;

Console.WriteLine(decimalVariable); // 1,1234567891234567890123456789

}

}

}

**2.5. CHAR TYPES**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Example\_005\_CharTypes

{

class Program

{

static void Main(string[] args)

{

// Тип char может содержать один Unicode-символ

char VarChar1 = 'a'; Console.WriteLine(VarChar1); // a

char VarChar2 = 'ю'; Console.WriteLine(VarChar2); // ю

char VarChar3 = '!'; Console.WriteLine(VarChar3); // !

char VarChar4 = '&'; Console.WriteLine(VarChar4); // &

char VarChar5 = '9'; Console.WriteLine(VarChar5); // 9

char VarChar6 = '§'; Console.WriteLine(VarChar6); // 󠅒󠅒

}

}

}

**2.6. BOOL TYPES**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Example\_006\_BoolTypes

{

class Program

{

static void Main(string[] args)

{

// Тип bool может хранить логическое значение, которое принимает значение

// правда или ложь.

// Эти значения указываются с ключевыми словами true и false

bool flag = true; // переменной flag присвоить истину

flag = false; // переменной flag присвоить ложь

// Замечание

// Подробное использование этого типа в уроке

// "Логические операции"

}

}

}

**2.7. STRING TYPES**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Example\_007\_StringTypes

{

class Program

{

static void Main(string[] args)

{

// Тип данных string используется для хранения строковых констант.

// Они обрамляются двойными кавычками.

string word = "Привет!";

word = "Привет, мир!";

// Замечание

// Продвинутые возможности использования этого типа в уроке

// "Файлы"

}

}

}

**2.8. FORMATTED OUTPUT**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Example\_001\_FormattedOutput

{

class Program

{

static void Main(string[] args)

{

// Составное форматирование

string firstName = "Александр";

string lastName = "Пушкин";

string specialization = "Писатель";

byte age = 37;

ulong yearOfBirth = 1799;

// АлександрПушкинПисатель371799

Console.WriteLine(firstName + lastName + specialization + age + yearOfBirth);

Console.ReadKey();

// Александр Пушкин Писатель 37 1799

Console.WriteLine(firstName + " " + lastName + " " + specialization + " " + age + " " + yearOfBirth);

Console.ReadKey();

// Имя: Александр Фамилия: Пушкин Специализация: Писатель Возраст: 37 Год рождения:1799

Console.WriteLine("Имя: " + firstName + " Фамилия: " + lastName + " Специализация: " + specialization + " Возраст: " + age + " Год рождения: " + yearOfBirth);

Console.ReadKey();

// Имя: Александр Фамилия: Пушкин Специализация: Писатель Возраст: 37 Год рождения:1799

Console.WriteLine("Имя: {0} Фамилия: {1} Специализация: {2} Возраст: {3} Год рождения: {4}",

firstName,

lastName,

specialization,

age,

yearOfBirth);

Console.ReadKey();

// Имя: Александр Фамилия: Пушкин Специализация: Писатель Возраст: 37 Год рождения:1799

string pattern = "Имя: {0} Фамилия: {1} Специализация: {2} Возраст: {3} Год рождения: {4}";

Console.WriteLine(pattern,

firstName,

lastName,

specialization,

age,

yearOfBirth);

#region переиспользование pattern

string fName = "Ада";

string lName = "Лавлейс";

string profession = "Программист";

byte years = 37;

ulong birth = 1815;

Console.WriteLine(pattern,

fName,

lName,

profession,

years,

birth);

#endregion

}

}

}

**2.9. ESCAPE SEQUNCES**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Example\_002\_EscapeSequences

{

class Program

{

static void Main(string[] args)

{

string firstName = "Александр";

string lastName = "Пушкин";

string specialization = "Писатель";

byte age = 37;

ulong yearOfBirth = 1799;

string pattern = "Имя: {0} Фамилия: {1} Специализация: {2} Возраст: {3} Год рождения: {4}";

#region newPattern

string newPattern = "Имя: {0} \nФамилия: {1} \nСпециализация: {2} \nВозраст: {3} \nГод рождения: {4}";

#endregion

Console.WriteLine(newPattern,

firstName,

lastName,

specialization,

age,

yearOfBirth);

Console.ReadKey();

#region Вывод данных Escape-последовательности

// \n Новая строка

// \r Возврат каретки

// \t Горизонтальная табуляция

// \' Одиночная кавычка

// \" Двойная кавычка

// \\ Обратная косая черта

// \? Литерал вопросительного знака

#endregion

}

}

}

**2.10. STRING INTERPOLATION**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Example\_003\_StringInterpolation

{

class Program

{

static void Main(string[] args)

{

string firstName = "Александр";

string lastName = "Пушкин";

string specialization = "Писатель";

string pattern = "Имя: {0} Фамилия: {1} Специализация: {2}";

Console.WriteLine(pattern,

firstName,

lastName,

specialization );

// Интерполяция строк

Console.WriteLine($"Имя: {firstName} Фамилия: {lastName} Специализация: {specialization}");

#region Калькулятор

Console.ReadKey(); Console.Clear();

int a = 100;

int b = 11;

int c = 111;

Console.WriteLine($"{a} + {b} = {c}");

#endregion

#region База данных v 1.0

Console.ReadKey(); Console.Clear();

string name1 = "Агата", lastName1 = "Лебедева";

string name2 = "Аделаида", lastName2 = "Виноградова";

string name3 = "Алима", lastName3 = "Попова";

string name4 = "Катя", lastName4 = "Калинина";

string name5 = "Рената", lastName5 = "Иванова";

Console.WriteLine($"{name1} {lastName1}");

Console.WriteLine($"{name2} {lastName2}");

Console.WriteLine($"{name3} {lastName3}");

Console.WriteLine($"{name4} {lastName4}");

Console.WriteLine($"{name5} {lastName5}");

#endregion

#region База данных v 2.0

Console.ReadKey(); Console.Clear();

Console.WriteLine($"{name1,10} {lastName1,11}");

Console.WriteLine($"{name2,10} {lastName2,11}");

Console.WriteLine($"{name3,10} {lastName3,11}");

Console.WriteLine($"{name4,10} {lastName4,11}");

Console.WriteLine($"{name5,10} {lastName5,11}");

#endregion

}

}

}

**2.11. NUMBER AFTER POINT**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Example\_004\_NumbersAfterPoint

{

class Program

{

static void Main(string[] args)

{

//double d = 123456.654321;

//Console.WriteLine(d); // 123456.654321

//string dFormated = d.ToString("#.###");

//Console.WriteLine(dFormated); // 123456,654

//Console.WriteLine("{0:0.000}", d); // 123456,654

//Console.WriteLine(d.ToString("#.###")); // 123456,654

//dFormated = d.ToString("#.## ## ## ## ##");

//Console.WriteLine(dFormated); // 123456,65 43 21

//dFormated = d.ToString("### ###.## ## ## ## ##");

//Console.WriteLine(dFormated); // 123 456,65 43 21

//Console.WriteLine("{0:000 000 000.000 000 000}", d); // 000 123 456,654 321 000

#region Замечание

// В зависимости от настроек языкового стандарта программы

// выводимые символы могут отличаться.

Console.WriteLine();

var date = new DateTime(2025, 09, 28, 01, 30, 59);

Console.WriteLine(date); //

//

//

Console.ReadKey();

Console.WriteLine($"{date:HH:mm}");

Console.WriteLine($"{date:yyyy-MM-dd}");

Console.WriteLine($"{date:yy.MM.dd}");

Console.WriteLine($"{date:dd.MM.yyy}");

#endregion

}

}

}

**2.12. DATA TYPE CONVERSIONS**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Example\_001\_DataTypeConversion

{

class Program

{

static void Main(string[] args)

{

#region Тиды данных

// byte от 0 до 255

// ushort от 0 до 65535

// uint от 0 до 2^32-1 >>> от 0 до 4 294 967 296

// ulong от 0 до 2^64-1 >>> от 0 до 18 446 744 073 709 551 616

// sbyte от -128 дo +127

// short от -32768 дo +32767

// int от -2^31 дo +2^31-1 >>> от -2 147 483 648 до +2 147 483 647

// long от -2^63 дo +2^63-1 >>> от -9 223 372 036 854 775 808 до +9 223 372 036 854 775 807

// Тип float принимает значения От от ±5,0 × 10^(−324) до ±1,7 × 10^308 точность - 15 цифр

// Тип double принимает значения От ±1,5 x 10^(−45) до ±3,4 x 10^38 точность - 7 цифр

// Тип decimal принимает значения от ±1,0 x 10^(-28) до ±7,9228 x 10^28 точность - 28 цифр

#endregion

byte b = 100; //

int a = b; // Всё хорошо

//int i = 10; //

//byte j = i; // Ошибка компилятора

double d = a; // Всё хорошо

//int g = d; // Ошибка компилятора

}

}

}

**2.13. Explicit Type Conversion**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Example\_003\_ExplicitTypeConversion

{

class Program

{

static void Main(string[] args)

{

#region Тиды данных

// byte от 0 до 255

// ushort от 0 до 65535

// uint от 0 до 2^32-1 >>> от 0 до 4 294 967 296

// ulong от 0 до 2^64-1 >>> от 0 до 18 446 744 073 709 551 616

// sbyte от -128 дo +127

// short от -32768 дo +32767

// int от -2^31 дo +2^31-1 >>> от -2 147 483 648 до +2 147 483 647

// long от -2^63 дo +2^63-1 >>> от -9 223 372 036 854 775 808 до +9 223 372 036 854 775 807

// Тип float принимает значения От от ±5,0 × 10^(−324) до ±1,7 × 10^308 точность - 15 цифр

// Тип double принимает значения От ±1,5 x 10^(−45) до ±3,4 x 10^38 точность - 7 цифр

// Тип decimal принимает значения от ±1,0 x 10^(-28) до ±7,9228 x 10^28 точность - 28 цифр

#endregion

//int i = 330; //

//byte j = i; // Ошибка компилятора

int i = 330; //

byte j = (byte)i; // Ошибки компилятора не будет

Console.WriteLine($"j = {j}");

//double d = 1.2; //

//byte byteValue = d; // Ошибка компилятора

double d = 1.2; //

byte byteValue = (byte)d; // Ошибки компилятора не будет

int h = (int)d;

int q = (int)((byte)d);

}

}

}

**2.14. Implicit Type Сonversion**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Example\_002\_ImplicitTypeСonversion

{

class Program

{

static void Main(string[] args)

{

#region Тиды данных

// byte от 0 до 255

// ushort от 0 до 65535

// uint от 0 до 2^32-1 >>> от 0 до 4 294 967 296

// ulong от 0 до 2^64-1 >>> от 0 до 18 446 744 073 709 551 616

// sbyte от -128 дo +127

// short от -32768 дo +32767

// int от -2^31 дo +2^31-1 >>> от -2 147 483 648 до +2 147 483 647

// long от -2^63 дo +2^63-1 >>> от -9 223 372 036 854 775 808 до +9 223 372 036 854 775 807

// Тип float принимает значения От от ±5,0 × 10^(−324) до ±1,7 × 10^308 точность - 15 цифр

// Тип double принимает значения От ±1,5 x 10^(−45) до ±3,4 x 10^38 точность - 7 цифр

// Тип decimal принимает значения от ±1,0 x 10^(-28) до ±7,9228 x 10^28 точность - 28 цифр

#endregion

byte byteValue = 1; // Задание значения переменной типа byte

short shortValue = byteValue; // Неявное приведение byte к short

int intValue = shortValue; // Неявное приведение short к int

long longValue = intValue; // Неявное приведение int к long

double doubleValue = longValue; // Неявное приведение long к double

//char c = 'я';

//int d = c;

Console.WriteLine(d);

}

}

}

**2.15. Parse Conversion**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Example\_004\_ParseConversion

{

class Program

{

static void Main(string[] args)

{

//string inputString = "2029";

//int intValue = (int)inputString; // Ошибка CS0029

// // Не удается неявно преобразовать тип "string" в "int".

string inputString = "2029";

int intValue = int.Parse(inputString);

Console.WriteLine(intValue); // 2029

double doubleValue = double.Parse(inputString);

Console.WriteLine(doubleValue); // 2029

ulong ulongValue = ulong.Parse(inputString);

Console.WriteLine(ulongValue); // 2029

byte byteValue = byte.Parse(inputString);

Console.WriteLine(byteValue); // Необработанное исключение:

// System.OverflowException: Значение было недопустимо

// малым или недопустимо большим для беззнакового байта.

}

}

}

**2.16. CONVERT TO**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Example\_005\_ConvertTo

{

class Program

{

static void Main(string[] args)

{

string inputString = "2029";

//int intValue = int.Parse(inputString);

int intValue = Convert.ToInt32(inputString);

Console.WriteLine(intValue); // 2029

long l = Convert.ToInt32(intValue);

//double doubleValue = double.Parse(inputString);

double doubleValue = Convert.ToDouble(inputString);

Console.WriteLine(doubleValue); // 2029

//ulong ulongValue = ulong.Parse(inputString);

ulong ulongValue = Convert.ToUInt64(inputString);

Console.WriteLine(ulongValue); // 2029

//byte byteValue = byte.Parse(inputString);

//byte byteValue = Convert.ToByte(inputString);

//Console.WriteLine(byteValue); // Необработанное исключение:

// // System.OverflowException: Значение было недопустимо

// // малым или недопустимо большим для беззнакового байта.

#region Типы данных и их синонимы

//Тип данных Тип.NET

//sbyte System.SByte

//short System.Int16

//int System.Int32

//long System.Int64

//byte System.SByte

//ushort System.UInt16

//uint System.UInt32

//ulong System.UInt64

//float System.Single

//double System.Double

//bool System.Boolean

#endregion

#region Замечание

//checked

//{

// byte byteValue = (byte)intValue;

// Console.WriteLine(byteValue);

//}

//unchecked

//{

// byte byteValue = (byte)intValue;

// Console.WriteLine(byteValue);

//}

#endregion

}

}

}

**2.17. Arithmetic Operations**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Example\_001\_ArithmeticOperations

{

class Program

{

static void Main(string[] args)

{

// Арифметические операции -,\*,/,+

#region Сумма +

//int valueA = 2\_140\_000\_003; int valueB = 2\_140\_000\_003;

//double valueC = (double)valueA + (double)valueB;

//Console.WriteLine(valueC);

//double valueD = 20.22; double valueE = 20.25;

//double valueI = valueD + valueE;

//sbyte valueF = 123; short valueG = 22;

//var valueH = valueF + valueG;

//long valueK = 32; byte valueL = 11;

//var valueM = valueK + valueL;

#endregion

#region Разность -

//int valueA = 3; int valueB = 2019;

//int valueC = valueA - valueB;

//double valueD = 20.22; double valueE = 20.25;

//double valueI = valueD - valueE;

//sbyte valueF = 123; short valueG = 22;

//var valueH = valueF - valueG;

//long valueK = 32; byte valueL = 11;

//var valueM = valueK - valueL;

#endregion

#region Произведение \*

//int valueA = 3; int valueB = 2019;

//int valueC = valueA \* valueB;

//double valueD = 20.22; double valueE = 20.25;

//double valueI = valueD \* valueE;

//sbyte valueF = 123; short valueG = 22;

//var valueH = valueF \* valueG;

//long valueK = 32; byte valueL = 11;

//double valueM = valueK \* valueL;

#endregion

#region Частное /

//int valueA = 10; int valueB = 2;

//int valueC = valueA / valueB;

//Console.WriteLine(valueC); // 5

//double valueA = 5; double valueB = 2;

//double valueC = valueA / valueB;

//Console.WriteLine(valueC); // Ожидание: 2.5

//// // Реальность: 2

//////// Деление в целых чисах [ byte, int, ulong и т.д.]

////////

//////// 56 / 7 = 8

//////// Делимое / Делитель = Частное

//////// 56 = 7 \* 8

////////

//////// 59 / 8 = 7 + 3

////////

//////// a = b \* q + r

//////// Делимое / Делитель = Неполное частное + Остаток [ 0 ⩽ Остаток < |Делитель| ]

//////// 59 = 7 \* 8 + [остаток] 3

//////// 21 = 4 \* 5 + [остаток] 1

//Console.WriteLine("Деление в целых числах");

//Console.WriteLine($"5/2 = {5 / 2}");

//Console.WriteLine($"59/8 = {59 / 8}");

//Console.WriteLine($"21/4 = {21 / 4}");

//Console.WriteLine("Деление в вещественных");

//Console.WriteLine($"5.0/2.0 = {5.0 / 2.0}");

//Console.WriteLine($"59.0/8.0 = {59.0 / 8.0}");

//Console.WriteLine($"21.0/4.0 = {21.0 / 4.0}");

#endregion

#region Остаток от деления %

//////// Деление в целых чисах [ byte, int, ulong и т.д.]

////////

//////// 56 / 7 = 8

//////// Делимое / Делитель = Частное

//////// 56 = 7 \* 8

////////

//////// 59 / 8 = 7 + 3

////////

//////// a = b \* q + r

//////// Делимое / Делитель = Неполное частное + Остаток [ 0 ⩽ Остаток < |Делитель| ]

//////// 59 = 7 \* 8 + [остаток] 3

//////// 21 = 4 \* 5 + [остаток] 1

//Console.WriteLine("Деление в целых числах. Остаток");

//Console.WriteLine($"5 / 2 = {5 / 2}"); // 2

//Console.WriteLine($"5 % 2 = {5 % 2}"); // 1

//Console.WriteLine($"59 % 8 = {59 % 8}");

//Console.WriteLine($"21 % 4 = {21 % 4}");

#endregion

}

}

}

**2.18. Increment And Decrement**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Example\_002\_IncrementAndDecrement

{

class Program

{

static void Main(string[] args)

{

#region +=

//Console.ReadKey(); Console.WriteLine("\n+=");

//int a = 2\_000\_000\_000;

//int b = 2\_000\_000\_000;

//int c = a + b;

//Console.WriteLine(c); // 25

//a = a + b;

//a += b;

//Console.WriteLine(a); // 25

#endregion

#region -=

Console.ReadKey(); Console.WriteLine("\n-=");

int a = 12;

int b = 13;

int c = a - b;

Console.WriteLine(c); // -1

//a = a - b;

a -= b;

Console.WriteLine(a); // -1

#endregion

#region \*=

Console.ReadKey(); Console.WriteLine("\n\*=");

a = 4;

b = 5;

c = a \* b;

Console.WriteLine(c); // 20

//a = a \* b;

a \*= b;

Console.WriteLine(a); // 20

#endregion

#region /=

Console.ReadKey(); Console.WriteLine("\n/=");

a = 33;

b = 11;

c = a / b;

Console.WriteLine(c); // 3

//a = a / b;

a /= b;

Console.WriteLine(a); // 3

a = 30;

b = 11;

c = a / b;

Console.WriteLine(c); //2

a /= b;

Console.WriteLine(a); //2

double d = 10;

double e = 4;

d /= e;

Console.WriteLine(d); //2.5

#endregion

#region %=

Console.ReadKey(); Console.WriteLine("\n%=");

a = 14;

b = 5;

c = a % b;

Console.WriteLine(c); // 4

//a = a % b;

a %= b;

Console.WriteLine(a); // 4

#endregion

#region ++

Console.ReadKey(); Console.WriteLine("\n++");

a = 10;

a += 20;

Console.WriteLine(a); //30

a += 1;

Console.WriteLine(a); //31

a++; // постфиксный инкремент

Console.WriteLine(a); //32

++a;// префиксный инкремент

Console.WriteLine(a); //33

#endregion

#region --

Console.ReadKey(); Console.WriteLine("\n--");

a = 50;

a -= 20;

Console.WriteLine(a); //30

a -= 1;

Console.WriteLine(a); //29

a--; // постфиксный инкремент

Console.WriteLine(a); //28

--a;// префиксный инкремент

Console.WriteLine(a); //27

#endregion

}

}

}

**2.19. Operations Priority**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Example\_003\_OperationsPriority

{

class Program

{

static void Main(string[] args)

{

//

// Унарные операции

// Префиксный инкремент

// %, /, \*

// +, -

// Постфиксный инкремент

// () - скобки меняют приоритет операций

//

Console.WriteLine((2+2)\*2);

}

}

}

**2.20. LOGICAL OPERATIONS**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Example\_004\_LogicalOperations

{

class Program

{

static void Main(string[] args)

{

// Высказывание (в математической логике) - Высказывание — это грамматически правильное

// повествовательное предложени языка (естественного или искусственного),

// которое выражает некоторый смысл и является либо истинным,

// либо ложным, но не тем и другим сразу.

// Пример 1. Na — металл - истина

// Пример 2: 2 + 2 = 5 - ложь

// Пример 3: Проксима Центавра - ближайшая к Земле звезде - Ложь

bool variable1 = true; // 1

bool variable2 = false; // 0

#region Инверсия / Отрицание / Логическое "Не" / !

Console.WriteLine("Инверсия: ");

// Определение. Отрицанием высказывания, будет являться Высказывание

// «противоположное» исходному

Console.WriteLine($"variable1 = {variable1} !variable1 = {!variable1}");

Console.WriteLine($"variable2 = {variable2} !variable2 = {!variable2}");

#endregion

#region Конъюнкция / Логическое умножение / Логическое "И" / &&

Console.WriteLine("\nКонъюнкция: ");

// Определение 2. Конъюнкцией двух высказываний, будет является высказывание

// истинное тогда и только тогда, когда оба исходных суждения истинны

// Пример 4: Nа — металл И Проксима Центавра - ближайшая к солнечной системе звезде

// Пример 5: Компьютер работает с нулями и единицами И все люди пьют воду

//variable1 = true;

//variable2 = false;

//bool result = variable1 && variable2;

//Console.WriteLine($" {true} && {true} = {true && true}");

//Console.WriteLine($" {true} && {false} = {true && false}");

//Console.WriteLine($"{false} && {true} = {false && true}");

//Console.WriteLine($"{false} && {false} = {false && false}");

//Console.ReadKey();

#endregion

#region Дизъюнкция / Логическое сложение / Логическое "ИЛИ" / ||

//Console.WriteLine("\nДизъюнкция: ");

// Определение 3. Дизъюнкцией двух высказываний, будет является высказывание

// ложное тогда и только тогда, когда оба исходных высказывания ложны

// Пример 6: Nа — металл ИЛИ 2 + 2 = 5

// Пример 7: Проксима Центавра - ближайшая к Земле звезде ИЛИ все люди пьют воду

//variable1 = true;

//variable2 = false;

//bool result = variable1 || variable2;

//Console.WriteLine($" {true} || {true} = {true || true}");

//Console.WriteLine($" {true} || {false} = {true || false}");

//Console.WriteLine($"{false} || {true} = {false || true}");

//Console.WriteLine($"{false} || {false} = {false || false}");

//Console.ReadKey();

#endregion

#region Разделительная дизъюнкция / Строгая дизъюнкция / Исключающее «ИЛИ» / ^

Console.WriteLine("\nРазделительная дизъюнкция: ");

// Определение 4. Разделительной дизъюнкцией двух высказываний, будет является высказывание

// истинное тогда и только тогда, когда только одно из исходных суждения истинно

//variable1 = true;

//variable2 = false;

//bool result = variable1 ^ variable2;

//Console.WriteLine($" {true} ^ {true} = {true ^ true}");

//Console.WriteLine($" {true} ^ {false} = {true ^ false}");

//Console.WriteLine($"{false} ^ {true} = {false ^ true}");

//Console.WriteLine($"{false} ^ {false} = {false | false}");

#endregion

#region Другие операции

bool flag1 = 4 > 5; Console.WriteLine(flag1);

bool flag2 = 1 < 3; Console.WriteLine(flag2);

char c = 'f';

bool flag3 = c == 'f'; Console.WriteLine(flag3);

bool flag4 = 28 == 28; Console.WriteLine(flag4);

bool flag5 = 28 >= 24; Console.WriteLine(flag5);

bool flag6 = 9 <= 9; Console.WriteLine(flag6);

bool flag7 = 28 != 90; Console.WriteLine(flag7);

string s = "C#";

bool flag8 = 28 != 90 && s == "C#"; Console.WriteLine(flag8);

#endregion

}

}

}

**2.21. MATH LOGICAL OPERATIONS**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Example\_005\_MathLogicalOperations

{

class Program

{

static void Main(string[] args)

{

int x1 = 30, x2 = 150, // Координаты вершин прямоугольника

y1 = 20, y2 = 90;

int pA = 50, pB = 110; // Координаты точки

// Проверка условия принадлежности точки ограниченной области

bool check = ((pA >= x1) && (pA <= x2)) && ((pB >= y1) && (pB <= y2));

Console.WriteLine(check);

Console.ReadKey();

#region \_

Console.ReadKey();

pA = 50; pB = 110;

check = ((pA >= x1) && (pA <= x2)) && ((pB >= y1) && (pB <= y2));

Console.WriteLine(check);

#endregion

}

}

}

**3.1. DATA READ**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

#region Мы узнаем о

// 1. Считывании данных с консоли

// 2. Генераторе псевдослучайных чисел

// 3. Практическом применение логического типа

// 4. Условном операторе (сокращённая форма)

// 5. Условном операторе (полная форма)

// 6. Условном операторе (каскады условий)

#endregion

namespace Example\_000\_DataRead

{

class Program

{

static void Main(string[] args)

{

// Вывод данных

string name = "Билл Гейтс";

byte age = 63;

ushort costs = 118;

#region \_

//Ввод данных

Console.WriteLine("Введите имя: ");

name = Console.ReadLine();

Console.WriteLine("Введите возраст: ");

age = byte.Parse(Console.ReadLine());

Console.WriteLine("Введите сумму ежемесячных трат: ");

costs = Convert.ToUInt16(Console.ReadLine());

#endregion

Console.WriteLine("Имя: " + name + " Возраст: " + age + " Расходы: $" + costs); // конкатенация строк

Console.WriteLine("Имя: {0} Возраст: {1} Расходы: ${2}", name, age, costs); // форматированный вывод

Console.WriteLine($"Имя: {name} Возраст: {age} Расходы: ${costs}"); // интерполяция строк

Console.ReadKey();

}

}

}

**3.2. RANDOM**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Example\_001\_Random

{

class Program

{

static void Main(string[] args)

{

Random randomize = new Random(); // Создание переменной randomize для получения псевослучайных чисел

Console.WriteLine($"int.MaxValue = {int.MaxValue}");

Console.WriteLine($"int.MinValue = {int.MinValue}\n");

#region Замечание

//Console.WriteLine($"byte.MaxValue = {byte.MaxValue}");

//Console.WriteLine($"byte.MinValue = {byte.MinValue}\n");

//Console.WriteLine($"uint.MaxValue = {uint.MaxValue}");

//Console.WriteLine($"uint.MinValue = {uint.MinValue}\n");

//Console.WriteLine($"long.MaxValue = {long.MaxValue}");

//Console.WriteLine($"long.MinValue = {long.MinValue}\n");

#endregion

int randomIntResult = randomize.Next(); // Возвращает неотрицательное случайное целое число

// 32-разрядное целое число со знаком,

// которое больше или равно 0 и меньше чем Int32.MaxValue.

Console.WriteLine($"Пример 1. randomIntResult = {randomIntResult}");

randomIntResult = randomize.Next(100); // Возвращает неотрицательное случайное целое число,

// которое меньше максимально допустимого значения

// от 0 до 100, не включая 100

Console.WriteLine($"Пример 2. randomIntResult = {randomIntResult}");

randomIntResult = randomize.Next(10, 100); // Возвращает неотрицательное случайное целое число,

// которое больше минимально допустимого значения

// и меньше максимально допустимого значения

// от 10 до 100, не включая 100

Console.WriteLine($"Пример 3. randomIntResult = {randomIntResult}");

double randomDoubleResult = randomize.NextDouble() + randomize.Next(100); // Возвращает случайное число с плавающей запятой,

// которое больше или равно 0.0 и меньше 1.0

// не включая 1.0

Console.WriteLine($"Пример 4. randomDoubleResult = {randomDoubleResult}");

}

}

}

**3.3. IF ELSE**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Example\_002\_If

{

class Program

{

static void Main(string[] args)

{

// Условные операторы используются для выполнения различных блоков кода на основе разных условий

// Оператор if будет выполняться только в том случае, если условие в скобках будет "истина".

// Условие может включать в себя любое сравнение и логические операторы

// т.е. bool результат

Random rand = new Random();

int x = rand.Next(3); // Выдаёт 0, 1 или 2

Console.WriteLine($"Получено значение x = {x}");

if (x < 2)

Console.WriteLine("x равен 0 или 1");

// Часто используемая форма записи

if (x < 2)

{

Console.WriteLine("x равен 0 или 1");

}

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Example\_003\_If\_Else

{

class Program

{

static void Main(string[] args)

{

// Чтобы проверить другие условия, оператор if может быть расширен любым количество if'ов.

// Каждое дополнительное условие будет проверено, только если все предыдущие условия являются ложными.

// Пример

Random rand = new Random();

int x = rand.Next(3); // Выдаёт 0, 1 или 2

Console.WriteLine($"Получено значение x = {x}");

//if (x == 1) Console.WriteLine("x == 1");

//else Console.WriteLine("x != 1");

//if (x==1)

//{

// Console.WriteLine("x == 1");

//}

//else

//{

// Console.WriteLine("x != 1");

//}

#region if-else-if

if (x < 1) Console.WriteLine("x < 1");

else if (x > 1) Console.WriteLine("x > 1");

else Console.WriteLine("x = 1");

#endregion

#region Пример записи 1

Console.ReadKey();

if (x < 1)

{

Console.WriteLine("x < 1");

}

else if (x > 1)

{

Console.WriteLine("x > 1");

}

else

{

Console.WriteLine("x = 1");

}

#endregion

#region Пример записи 2

Console.ReadKey();

if (true)

{

}

else

{

if (true)

{

}

else

{

if (true)

{

}

else

{

}

}

}

if (x < 1)

{

Console.WriteLine("x < 1");

}

else

{

if (x > 1)

{

Console.WriteLine("x > 1");

}

else

{

Console.WriteLine("x = 1");

}

}

#endregion

}

}

}

EXAMPLE 1

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Example\_004

{

class Program

{

static void Main(string[] args)

{

// Заказчик просит написать программу, которая определяет текущую температуру

// и даёт рекомендацию по выбору одежды для прогулки

Random rand = new Random();

int temperature = rand.Next(30); // Выдаёт от 0 до 30

Console.WriteLine($”Текущая температура = {temperature}°с”);

if (temperature < 18)

{

Console.WriteLine(“На улице прохладно, рекомендую взять куртку потеплее”);

}

else

{

Console.WriteLine(«На улице тепло, теплая куртка не понадобится»);

}

}

}

}

EXAMPLE 2

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Example\_005

{

class Program

{

static void Main(string[] args)

{

// Заказчик просит написать программу, которая должна генерировать два случайных числа

// из диапазона от -10 до 10 после чего на экране должно быть выведено:

// 1.1. Максимальное из чисел

// 1.2. Минимальное из чисел

// 2. Добавить в программу возможность определения равенства чисел

// и вывода соответствующего сообщения на экран

// 3. Добавить возможность ввода данных с клавиатуры

Random rand = new Random();

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

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

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

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

#region Пункт 1 вариант 1

Console.WriteLine("Пункт 1 вариант 1");

if (firstNumber > secondNumber)

{

Console.WriteLine($"Максимальное число = {firstNumber}");

}

else

{

Console.WriteLine($"Максимальное число = {secondNumber}");

}

if (firstNumber < secondNumber)

{

Console.WriteLine($"Минимальное число = {firstNumber}");

}

else

{

Console.WriteLine($"Минимальное число = {secondNumber}");

}

#endregion

#region Пункт 1 вариант 2

Console.WriteLine("\nПункт 1 вариант 2");

if (firstNumber > secondNumber)

{

Console.WriteLine($"Максимальное число = {firstNumber}");

Console.WriteLine($"Минимальное число = {secondNumber}");

}

else

{

Console.WriteLine($"Максимальное число = {secondNumber}");

Console.WriteLine($"Минимальное число = {firstNumber}");

}

#endregion

#region Пункт 1 вариант 3

Console.WriteLine("\nПункт 1 вариант 3");

bool flag = firstNumber > secondNumber;

if (flag)

{

Console.WriteLine($"Максимальное число = {firstNumber}");

Console.WriteLine($"Минимальное число = {secondNumber}");

}

else

{

Console.WriteLine($"Максимальное число = {secondNumber}");

Console.WriteLine($"Минимальное число = {firstNumber}");

}

#endregion

#region Пункт 1 вариант 4

Console.WriteLine("\nПункт 1 вариант 4");

int maxValue, minValue;

if (firstNumber > secondNumber)

{

maxValue = firstNumber;

minValue = secondNumber;

}

else

{

maxValue = secondNumber;

minValue = firstNumber;

}

Console.WriteLine($"Максимальное число = {maxValue}");

Console.WriteLine($"Минимальное число = {minValue}");

#endregion

#region Пункт 2

Console.WriteLine("\nПункт 2");

if (firstNumber > secondNumber)

{

maxValue = firstNumber;

minValue = secondNumber;

}

else if(firstNumber < secondNumber)

{

maxValue = secondNumber;

minValue = firstNumber;

}

else

{

Console.WriteLine("Числа равны");

}

Console.WriteLine($"Максимальное число = {maxValue}");

Console.WriteLine($"Минимальное число = {minValue}");

#endregion

#region Пункт 3

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

//int firstNumber = int.Parse(Console.ReadLine());

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

//int secondNumber = int.Parse(Console.ReadLine());

#endregion

}

}

}

EXAMPLE 3

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Example\_006

{

class Program

{

static void Main(string[] args)

{

// Заказчик просит написать приложение, которое по введённому номеру дня недели

// выведет на экран название этого дня или сообщит, что введённое число не корректно

Console.Write("Введите номер дня: ");

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

#region Вариант 1

if (dayIndex == 1)

{

Console.WriteLine("Понедельник");

}

else

{

if (dayIndex == 2)

{

Console.WriteLine("Вторник");

}

else

{

if (dayIndex == 3)

{

Console.WriteLine("Среда");

}

else

{

if (dayIndex == 4)

{

Console.WriteLine("Четверг");

}

else

{

if (dayIndex == 5)

{

Console.WriteLine("Пятница");

}

else

{

if (dayIndex == 6)

{

Console.WriteLine("Суббота");

}

else

{

if (dayIndex == 7)

{

Console.WriteLine("Воскресенье");

}

else

{

Console.WriteLine("На планете Земля в неделе семь дней");

}

}

}

}

}

}

}

#endregion

#region Вариант 2

if (dayIndex == 1) Console.WriteLine("Понедельник");

else if (dayIndex == 2) Console.WriteLine("Вторник");

else if (dayIndex == 3) Console.WriteLine("Среда");

else if (dayIndex == 4) Console.WriteLine("Четверг");

else if (dayIndex == 5) Console.WriteLine("Пятница");

else if (dayIndex == 6) Console.WriteLine("Суббота");

else if (dayIndex == 7) Console.WriteLine("Воскресенье");

else Console.WriteLine("На планете Земля в неделе семь дней");

#endregion

}

}

}

#region

// Мы узнали как

// 1. Считывать данные с консоли

// 2. Генерировать псевдослучайные числа

// 3. Применять логический тип при использовании условного оператора

// 4. Применять условный оператор (сокращённая форма)

// 5. Применять условный оператор (полная форма)

// 6. Применять условный оператор (каскады условий)

#endregion

**3.4. TERNARY OPERATOR**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Example\_001\_TernaryOperator

{

class Program

{

static void Main(string[] args)

{

// Тернарный оператор

// В дополнение к оператору if есть, так называемый, тернарный оператор (?:).

// Этот оператор может заменить одно предложение if-else, которое

// присваивает значение конкретной переменной.

// Оператор принимает три выражения.

// Если первый оценивается как true, будет возвращено второе выражение,

// и если оно false, то возвращается третий

Random r = new Random();

double x = r.NextDouble(); // Получение числа от 0.0 до 1.0, не включая 1.0

int result = (x < 0.5) ? 0 : 1; // Тернарный оператор (?:)

Console.WriteLine($"result = {result}");

if (x < 0.5)

{

result = 0;

}

else

{

result = 1;

}

Console.WriteLine($"result = {result}");

#region Замечание

Console.ReadKey();

Console.WriteLine(x < 0.5 ? 0 : 1);

if (x < 0.5)

{

Console.WriteLine(0);

}

else

{

Console.WriteLine(1);

}

#endregion

}

}

}

EXAMPLE 1

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Example\_002

{

class Program

{

static void Main(string[] args)

{

// Заказчик просит написать программу, которая должна генерировать три случайных числа

// из диапазона от -21 до 22 после чего на экране должно быть выведено максимальное из чисел.

// Если максимальных чисил несколько ( т.е. каких-то два или все три числа равны)

// на экране показать одно число без дополнительных сообщений

#region Решение if-else

Random rand = new Random();

int firstNumber = rand.Next(-21, 23); // Выдаёт от -21 до 23

int secondNumber = rand.Next(-21, 23); // Выдаёт от -21 до 23

int thirdNumber = rand.Next(-21, 23); // Выдаёт от -21 до 23

int maxValue;

// Если первое число больше двух других,

if((firstNumber > secondNumber) && (firstNumber > thirdNumber))

{

maxValue = firstNumber; // то оно максимальное

}

else // Если первое число меньше второго или меньше третьего, то максимальное среди двух оставшихся

{

if (secondNumber> thirdNumber) // Сравниваем второе и третье число, если второе больше третьего

{

maxValue = secondNumber; // то максимальное - второе

}

else// иначе

{

maxValue = thirdNumber;// максимальное - третье

}

}

Console.WriteLine($"Максимальное значение = {maxValue}"); // Выводим результат на экран

#endregion

#region Решение ?:

Console.ReadKey();

int a = rand.Next(-21, 23); // Выдаёт от -21 до 23

int b = rand.Next(-21, 23); // Выдаёт от -21 до 23

int c = rand.Next(-21, 23); // Выдаёт от -21 до 23

maxValue = (a > b) && (a > c) ? a : (b > c) ? b : c;

maxValue = (firstNumber > b) && (firstNumber > thirdNumber) ? firstNumber : (secondNumber > thirdNumber) ? secondNumber : thirdNumber;

Console.WriteLine($"Максимальное значение = {maxValue}"); // Выводим результат на экран

#endregion

}

}

}

EXAMPLE 2

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Example\_003

{

class Program

{

static void Main(string[] args)

{

Random rand = new Random();

int a = rand.Next(-21, 23); // Выдаёт от -21 до 23

int b = rand.Next(-21, 23); // Выдаёт от -21 до 23

bool f = a > b;

int outputInt = f ? a : b;

Console.WriteLine(outputInt);

string outputString = f ? "да" : "нет";

var result = f ? "да" : "";

Console.WriteLine(outputString);

}

}

}

**3.5. SWITCH**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Example\_001\_Switch

{

class Program

{

static void Main(string[] args)

{

// Целое число может принимть множество вариантов

// Применение if-else для всех значений может быть громоздко

// поэтому в некоторых случаях выгоднее использовать оператор выбора Switch

Console.Write("Введите номер дня: ");

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

if (dayIndex == 1) Console.WriteLine("Понедельник");

else if (dayIndex == 2) Console.WriteLine("Вторник");

else if (dayIndex == 3) Console.WriteLine("Среда");

else if (dayIndex == 4) Console.WriteLine("Четверг");

else if (dayIndex == 5) Console.WriteLine("Пятница");

else if (dayIndex == 6) Console.WriteLine("Суббота");

else if (dayIndex == 7) Console.WriteLine("Воскресенье");

else Console.WriteLine("На планете Земля в неделе семь дней");

switch (dayIndex)

{

case 1:

Console.WriteLine("Понедельник");

break;

case 2:

Console.WriteLine("Вторник");

break;

case 3:

Console.WriteLine("Среда");

break;

case 4:

Console.WriteLine("Четверг");

break;

case 5:

Console.WriteLine("Пятница");

break;

case 6:

Console.WriteLine("Суббота");

break;

case 7:

Console.WriteLine("Воскресенье");

break;

default:

Console.WriteLine("На планете Земля в неделе семь дней");

break;

}

}

}

}

EXAMPLE 1

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Example\_002

{

class Program

{

static void Main(string[] args)

{

// Заказчик просит создать эмулятор игры, в которм у игрока есть возможность выбрать

// уровень сложности, в зависимости от которого будет генерироваться разный игровой мир

Console.WriteLine("Введите уровень сложности. 1 - Простой, 2 - Средний, 3 - Сложный");

int level; // 1 - Простой

// 2 - Средний

// 3 - Сложный

// В зависимости от уровня определить количество генериуемых NPC

// Простой - Два NPC

// Средний - Четыре NPC

// Сложный - Пять NPC

level = Convert.ToInt32(Console.ReadLine());

#region if-else

if (level == 1)

{

Console.WriteLine("NPC сгенерирован");

Console.WriteLine("NPC сгенерирован");

}

else if (level == 2)

{

Console.WriteLine("NPC сгенерирован");

Console.WriteLine("NPC сгенерирован");

Console.WriteLine("NPC сгенерирован");

Console.WriteLine("NPC сгенерирован");

}

else if (level == 3)

{

Console.WriteLine("NPC сгенерирован");

Console.WriteLine("NPC сгенерирован");

Console.WriteLine("NPC сгенерирован");

Console.WriteLine("NPC сгенерирован");

Console.WriteLine("NPC сгенерирован");

}

else

{

Console.WriteLine("Выбран неизвестный уровень сложности");

}

#endregion

#region switch

switch (level)

{

case 1:

Console.WriteLine("NPC сгенерирован");

Console.WriteLine("NPC сгенерирован");

break;

case 2:

Console.WriteLine("NPC сгенерирован");

Console.WriteLine("NPC сгенерирован");

Console.WriteLine("NPC сгенерирован");

Console.WriteLine("NPC сгенерирован");

break;

case 3:

Console.WriteLine("NPC сгенерирован");

Console.WriteLine("NPC сгенерирован");

Console.WriteLine("NPC сгенерирован");

Console.WriteLine("NPC сгенерирован");

Console.WriteLine("NPC сгенерирован");

break;

default:

Console.WriteLine("Выбран неизвестный уровень сложности");

break;

}

#endregion

}

}

}

EXAMPLE 2

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Example\_002

{

class Program

{

static void Main(string[] args)

{

// Заказчик просит создать эмулятор игры, в которм у игрока есть возможность выбрать

// уровень сложности, в зависимости от которого будет генерироваться разный игровой мир

Console.WriteLine("Введите уровень сложности. 1 - Простой, 2 - Средний, 3 - Сложный");

int level; // 1 - Простой

// 2 - Средний

// 3 - Сложный

// В зависимости от уровня определить количество генериуемых NPC

// Простой - Два NPC

// Средний - Четыре NPC

// Сложный - Пять NPC

level = Convert.ToInt32(Console.ReadLine());

#region if-else

if (level == 1)

{

Console.WriteLine("NPC сгенерирован");

Console.WriteLine("NPC сгенерирован");

}

else if (level == 2)

{

Console.WriteLine("NPC сгенерирован");

Console.WriteLine("NPC сгенерирован");

Console.WriteLine("NPC сгенерирован");

Console.WriteLine("NPC сгенерирован");

}

else if (level == 3)

{

Console.WriteLine("NPC сгенерирован");

Console.WriteLine("NPC сгенерирован");

Console.WriteLine("NPC сгенерирован");

Console.WriteLine("NPC сгенерирован");

Console.WriteLine("NPC сгенерирован");

}

else

{

Console.WriteLine("Выбран неизвестный уровень сложности");

}

#endregion

#region switch

switch (level)

{

case 1:

Console.WriteLine("NPC сгенерирован");

Console.WriteLine("NPC сгенерирован");

break;

case 2:

Console.WriteLine("NPC сгенерирован");

Console.WriteLine("NPC сгенерирован");

Console.WriteLine("NPC сгенерирован");

Console.WriteLine("NPC сгенерирован");

break;

case 3:

Console.WriteLine("NPC сгенерирован");

Console.WriteLine("NPC сгенерирован");

Console.WriteLine("NPC сгенерирован");

Console.WriteLine("NPC сгенерирован");

Console.WriteLine("NPC сгенерирован");

break;

default:

Console.WriteLine("Выбран неизвестный уровень сложности");

break;

}

#endregion

}

}

}

**3.6. REITERATION**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Example\_001\_Reiteration

{

class Program

{

static void Main(string[] args)

{

// Пример 1. Сгенерировать 10 случайных чисел от -8 до 9

Random rand = new Random();

Console.WriteLine($"1-e случайное число {rand.Next(-8, 10)}");

Console.WriteLine($"2-e случайное число {rand.Next(-8, 10)}");

Console.WriteLine($"3-e случайное число {rand.Next(-8, 10)}");

Console.WriteLine($"4-e случайное число {rand.Next(-8, 10)}");

Console.WriteLine($"5-e случайное число {rand.Next(-8, 10)}");

Console.WriteLine($"6-e случайное число {rand.Next(-8, 10)}");

Console.WriteLine($"7-e случайное число {rand.Next(-8, 10)}");

Console.WriteLine($"8-e случайное число {rand.Next(-8, 10)}");

Console.WriteLine($"9-e случайное число {rand.Next(-8, 10)}");

Console.WriteLine($"10-e случайное число {rand.Next(-8, 10)}");

Console.WriteLine($"11-e случайное число {rand.Next(-8, 10)}");

Console.WriteLine($"12-e случайное число {rand.Next(-8, 10)}");

Console.WriteLine($"13-e случайное число {rand.Next(-8, 10)}");

Console.WriteLine($"14-e случайное число {rand.Next(-8, 10)}");

Console.WriteLine($"15-e случайное число {rand.Next(-8, 10)}");

Console.WriteLine($"16-e случайное число {rand.Next(-8, 10)}");

Console.WriteLine($"17-e случайное число {rand.Next(-8, 10)}");

Console.WriteLine($"18-e случайное число {rand.Next(-8, 10)}");

Console.WriteLine($"19-e случайное число {rand.Next(-8, 10)}");

Console.WriteLine($"20-e случайное число {rand.Next(-8, 10)}");

}

}

}

EXAMPLE 1

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Example\_002

{

class Program

{

static void Main(string[] args)

{

// Пример 2. Заказчик просит вас написать прогрмму, в которой

// пользователь вводит целое число start.

// Программа должна вывести таблицу квадратов чисел от start до start+10

Console.Write("\nВведите start:");

int start = Convert.ToInt32(Console.ReadLine());

Console.WriteLine($"{start,4}^2 = {start \* start}"); start++;

Console.WriteLine($"{start,4}^2 = {start \* start}"); start++;

Console.WriteLine($"{start,4}^2 = {start \* start}"); start++;

Console.WriteLine($"{start,4}^2 = {start \* start}"); start++;

Console.WriteLine($"{start,4}^2 = {start \* start}"); start++;

Console.WriteLine($"{start,4}^2 = {start \* start}"); start++;

Console.WriteLine($"{start,4}^2 = {start \* start}"); start++;

Console.WriteLine($"{start,4}^2 = {start \* start}"); start++;

Console.WriteLine($"{start,4}^2 = {start \* start}"); start++;

Console.WriteLine($"{start,4}^2 = {start \* start}"); start++;

Console.WriteLine($"{start,4}^2 = {start \* start}");

}

}

}

EXAMPLE 2

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Example\_003

{

class Program

{

static void Main(string[] args)

{

// Пример 3. Заказчик просит вас написать прогрмму, в которой

// пользователь может ввести два целых числа start и end.

// Программа должна вывести таблицу квадратов чисел от start до end

// Разница между start и end не может превышать 20

Console.Write("\nВведите start:");

int start = Convert.ToInt32(Console.ReadLine());

Console.Write("\nВведите end:");

int end = Convert.ToInt32(Console.ReadLine());

if (start <= end) { Console.WriteLine($"{start,4}^2 = {start \* start}"); start++; }

if (start <= end) { Console.WriteLine($"{start,4}^2 = {start \* start}"); start++; }

if (start <= end) { Console.WriteLine($"{start,4}^2 = {start \* start}"); start++; }

if (start <= end) { Console.WriteLine($"{start,4}^2 = {start \* start}"); start++; }

if (start <= end) { Console.WriteLine($"{start,4}^2 = {start \* start}"); start++; }

if (start <= end) { Console.WriteLine($"{start,4}^2 = {start \* start}"); start++; }

if (start <= end) { Console.WriteLine($"{start,4}^2 = {start \* start}"); start++; }

if (start <= end) { Console.WriteLine($"{start,4}^2 = {start \* start}"); start++; }

if (start <= end) { Console.WriteLine($"{start,4}^2 = {start \* start}"); start++; }

if (start <= end) { Console.WriteLine($"{start,4}^2 = {start \* start}"); start++; }

if (start <= end) { Console.WriteLine($"{start,4}^2 = {start \* start}"); start++; }

if (start <= end) { Console.WriteLine($"{start,4}^2 = {start \* start}"); start++; }

if (start <= end) { Console.WriteLine($"{start,4}^2 = {start \* start}"); start++; }

if (start <= end) { Console.WriteLine($"{start,4}^2 = {start \* start}"); start++; }

if (start <= end) { Console.WriteLine($"{start,4}^2 = {start \* start}"); start++; }

if (start <= end) { Console.WriteLine($"{start,4}^2 = {start \* start}"); start++; }

if (start <= end) { Console.WriteLine($"{start,4}^2 = {start \* start}"); start++; }

if (start <= end) { Console.WriteLine($"{start,4}^2 = {start \* start}"); start++; }

if (start <= end) { Console.WriteLine($"{start,4}^2 = {start \* start}"); start++; }

if (start <= end) { Console.WriteLine($"{start,4}^2 = {start \* start}"); start++; }

}

}

}

EXAMPLE 3

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Example\_004

{

class Program

{

static void Main(string[] args)

{

// Пример 4. Заказчик прочит вас написать программу по подсчёту

// среднего балла полученных студентами на экзамене

// Пользователь с клавиатуры вводит 10 чисел

// Требуется найти среднее арифметическое этих чисел

Console.Write("Введите 1-й результат вступительного испытания: ");

double value1 = double.Parse(Console.ReadLine());

Console.Write("Введите 2-й результат вступительного испытания: ");

double value2 = double.Parse(Console.ReadLine());

Console.Write("Введите 3-й результат вступительного испытания: ");

double value3 = double.Parse(Console.ReadLine());

Console.Write("Введите 4-й результат вступительного испытания: ");

double value4 = double.Parse(Console.ReadLine());

Console.Write("Введите 5-й результат вступительного испытания: ");

double value5 = double.Parse(Console.ReadLine());

Console.Write("Введите 6-й результат вступительного испытания: ");

double value6 = double.Parse(Console.ReadLine());

Console.Write("Введите 7-й результат вступительного испытания: ");

double value7 = double.Parse(Console.ReadLine());

Console.Write("Введите 8-й результат вступительного испытания: ");

double value8 = double.Parse(Console.ReadLine());

Console.Write("Введите 9-й результат вступительного испытания: ");

double value9 = double.Parse(Console.ReadLine());

Console.Write("Введите 10-й результат вступительного испытания: ");

double value10 = double.Parse(Console.ReadLine());

double average = (value1 + value2 + value3 + value4 + value5 + value6 + value7 + value8 + value9 + value10) / 10;

Console.WriteLine($"Среднее = {average}");

}

}

}

EXAMPLE 4

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Example\_005

{

class Program

{

static void Main(string[] args)

{

// Пример 5. Пользователь вводит два целых числа numerator и denominator,

// которые являются числителем и знаменателем соответственно

// Нужно выяснить, можно ли сократить эту дробь

// и если можно, то на какое число

// 50 25 5 1

// ----- = ---- = ---- = ---

// 100 50 10 2

// https://ru.wikipedia.org/wiki/Наибольший\_общий\_делитель

// https://ru.wikipedia.org/wiki/Алгоритм\_Евклида

// Математика 5 класс. Тема "Сокращение дробей"

Console.Write("\nВведите numerator:");

double numerator = Convert.ToDouble(Console.ReadLine());

Console.Write("\nВведите denominator:");

double denominator = Convert.ToDouble(Console.ReadLine());

Console.WriteLine($"\n Исходная дробь {numerator}/{denominator}");

double a = numerator, b = denominator;

if (a != b) if (a > b) a = a - b; else b = b - a;

if (a != b) if (a > b) a = a - b; else b = b - a;

if (a != b) if (a > b) a = a - b; else b = b - a;

if (a != b) if (a > b) a = a - b; else b = b - a;

Console.WriteLine($"Можно сократить на {a}");

Console.WriteLine($"Сокращённая дробь {numerator / a}/{denominator / a}");

}

}

}

EXAMPLE 5

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Example\_006

{

class Program

{

static void Main(string[] args)

{

// Пример 6.

// Заказчик прочит вас написать игру "Угадай-ка", в которой

// игрок должен угадать число из диапазона от 1 до 100

// за минимальное количество шагов.

// При этом, после очередной попытки, игрок должен видеть сообщение

// "Введенное число больше загаданного. Попробуйте ещё раз"

// "Введенное число меньше загаданного. Попробуйте ещё раз"

// или "Поздравляю, число угадано! Число попыток: {число}.

}

}

}

3.7. LOOPS

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Example\_000\_Loops

{

class Program

{

static void Main(string[] args)

{

// В C# есть три+одна циклическая конструкция (оператор).

// Они используются для выполнения определенной

// последовательности действий несколько раз.

// Цикл — разновидность управляющей конструкции,

// предназначенная для организации многократного исполнения набора инструкций.

// Блок исполняемых инструкций называется телом цикла

// Одно исполнение набора инструкций называют итерацией цикла

}

}

}

**3.8. FOR LOOP**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Example\_001\_ForLoop

{

class Program

{

static void Main(string[] args)

{

// Цикл for используется для прохождения через блок указанное число раз.

// Он использует три параметра.

// Первый параметр инициализирует счетчик и всегда выполняется один раз, перед циклом.

// Второй параметр - условие для цикла, оно проверяется перед каждой итерацией.

// Третий параметр содержит приращение счетчика и выполняется в конце каждой итерации.

// Первый ; Второй; Третий

for (int i = 0; i < 10; i++)

{

// Тело цикла

}

Console.WriteLine("Вывод чисел от 0 до 9");

for (int i = 0; i < 10; i++)

{

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

}

Console.WriteLine("\n\nВывод чисел от -5 до 5");

for (int i = -5; i <= 5; i++)

{

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

}

Console.WriteLine("\n\nВывод чисел от 10 до 1. Вариант 1");

for (int j = 0; j < 10; j++)

{

Console.Write($"{10 - j} ");

}

Console.WriteLine("\n\nВывод чисел от 10 до 1. Вариант 2");

for (int k = 10; k >= 1; k--)

{

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

}

Console.WriteLine("\n\nВывод чисел от 1 до 22 c шагом 3");

for (int counter = 1; counter <= 22; counter += 3)

{

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

}

Console.WriteLine("\n\nВывод чисел от -1 до 1 c шагом 0.25");

for (double step = -1; step <= 1; step += 0.25)

{

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

}

Console.WriteLine("\n\nВывод чисел от 1 до -1 c шагом 0.5");

for (double interval = 1; interval >= -1; interval -= 0.5)

{

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

}

Console.ReadKey();

}

}

}

EXAMPLE 1

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Example\_002

{

class Program

{

static void Main(string[] args)

{

// Пример 1. Сгенерировать 10 случайных чисел от -8 до 9

Random r = new Random();

for (int i = 0; i < 200; i++)

{

Console.Write($"{r.Next(-8, 10)} ");

}

Console.ReadKey();

}

}

}

EXAMPLE 2

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Example\_003

{

class Program

{

static void Main(string[] args)

{

// Пример 2. Заказчик просит вас написать прогрмму, в которой

// пользователь вводит целое число start.

// Программа должна вывести таблицу квадратов чисел от start до start+10

Console.Write("\nВведите start:");

int start = Convert.ToInt32(Console.ReadLine());

for (int i = start; i <= start + 10; i++)

{

Console.WriteLine($"{i,4}^2 = {i \* i}");

}

}

}

}

EXAMPLE 3

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Example\_004

{

class Program

{

static void Main(string[] args)

{

// Пример 3. Заказчик просит вас написать прогрмму, в которой

// пользователь может ввести два целых числа start и end.

// Программа должна вывести таблицу квадратов чисел от start до end

// (!) Разница между start и end не может превышать 20

Console.Write("\nВведите start:");

int start = Convert.ToInt32(Console.ReadLine());

Console.Write("\nВведите end:");

int end = Convert.ToInt32(Console.ReadLine());

for (int i = start; i <= end; i++)

{

Console.WriteLine($"{i,4}^2 = {i \* i}");

}

}

}

}

EXAMPLE 4

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Example\_005

{

class Program

{

static void Main(string[] args)

{

// Пример 4. Заказчик прочит вас написать программу по подсчёту

// среднего балла полученных студентами на экзамене

// (i) Пользователь с клавиатуры вводит 10 чисел

// Требуется найти среднее арифметическое этих чисел

double value;

double average = 0;

for (int i = 1; i <= 10; i++)

{

Console.Write($"Введите {i}-й результат вступительного испытания: ");

value = double.Parse(Console.ReadLine());

average += value;

}

Console.WriteLine($"Среднее = {average / 10}");

}

}

}

EXAMPLE 5

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Example\_006

{

class Program

{

static void Main(string[] args)

{

// Пример 5. Пользователь вводит два целых числа numerator и denominator,

// которые являются числителем и знаменателем соответственно

// Нужно выяснить, можно ли сократить эту дробь

// и если можно, то на какое число

// 50 25 5 1

// ----- = ---- = ---- = ---

// 100 50 10 2

// https://ru.wikipedia.org/wiki/Наибольший\_общий\_делитель

// https://ru.wikipedia.org/wiki/Алгоритм\_Евклида

// Математика 5 класс. Тема "Сокращение дробей"

Console.Write("\nВведите numerator:");

double numerator = Convert.ToDouble(Console.ReadLine());

Console.Write("\nВведите denominator:");

double denominator = Convert.ToDouble(Console.ReadLine());

Console.WriteLine($"Исходная дробь {numerator}/{denominator}");

double a = numerator, b = denominator;

int count = 0;

for (int i = 0; i < 1000; i++)

{

if (a != b)

{

if (a > b) a = a - b; else b = b - a;

Console.WriteLine($"{a} {b}");

count++;

break;

}

Console.WriteLine($"i = {i} ");

}

Console.WriteLine($"count = {count}");

#region \_

//for (; a != b;)

//{

// if (a > b) a = a - b; else b = b - a;

//}

#endregion

Console.WriteLine($"Можно сократить на {a}");

Console.WriteLine($"Сокращённая дробь {numerator / a}/{denominator / a}");

}

}

}

EXAMPLE 6

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Example\_007

{

class Program

{

static void Main(string[] args)

{

// Пример 6.

// Заказчик прочит вас написать игру "Угадай-ка", в которой

// игрок должен угадать число из диапазона от 1 до 100

// за минимальное количество шагов.

// При этом, после очередной попытки, игрок должен видеть сообщение

// "Введенное число больше загаданного. Попробуйте ещё раз"

// "Введенное число меньше загаданного. Попробуйте ещё раз"

// или "Поздравляю, число угадано! Число попыток: {число}.

Console.WriteLine("Игра \"Угадай-ка\"");

Random rand = new Random();

int iiNumber = rand.Next(1, 101);

int count = 0;

int userNumber;

for (; ; )

{

Console.Write("\nВведите число: ");

count++;

userNumber = Convert.ToInt32(Console.ReadLine());

if (userNumber < iiNumber)

{

Console.WriteLine("Введенное число меньше загаданного. Попробуйте ещё раз");

}

else if (userNumber > iiNumber)

{

Console.WriteLine("Введенное число больше загаданного. Попробуйте ещё раз");

}

else

{

Console.WriteLine($"Поздравляю, число угадано! Число попыток: {count}.");

Console.ReadKey();

break;

}

}

}

}

}

**3.9. WHILE LOOP**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Example\_001\_WhileLoop

{

class Program

{

static void Main(string[] args)

{

// Цикл while выполняет блок кода, только если его условие истинно и

// будет продолжаться до тех пор, пока условие остается верным.

// Условие проверяется только в начале каждой итерации.

bool condition = false;

while (condition)

{

// Тело цикла

}

Console.WriteLine("Вывод чисел от 0 до 9");

int i = 0;

while ( i < 10)

{

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

i++;

}

Console.WriteLine("\n\nВывод чисел от -5 до 5");

i = -5;

while(i <= 5)

{

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

i++;

}

Console.WriteLine("\n\nВывод чисел от 10 до 1. Вариант 1");

int j = 0;

while (j < 10)

{

Console.Write($"{10 - j} ");

j++;

}

Console.WriteLine("\n\nВывод чисел от 10 до 1. Вариант 2");

int k = 10;

while ( k >= 1)

{

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

k--;

}

Console.WriteLine("\n\nВывод чисел от 1 до 22 c шагом 3");

int counter = 1;

while ( counter <= 22)

{

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

counter += 3;

}

Console.WriteLine("\n\nВывод чисел от -1 до 1 c шагом 0.25");

double step = -1;

while ( step <= 1)

{

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

step += 0.25;

}

Console.WriteLine("\n\nВывод чисел от 1 до -1 c шагом 0.5");

double interval = 1;

while ( interval >= -1)

{

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

interval -= 0.5;

}

Console.ReadKey();

}

}

}

EXAMPLE 1

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Example\_002

{

class Program

{

static void Main(string[] args)

{

// Пример 1. Сгенерировать 10 случайных чисел от -8 до 9

Random r = new Random();

int i = 0;

while ( i < 10)

{

Console.Write($"{r.Next(-8, 10)} ");

i++;

}

Console.ReadKey();

}

}

}

EXAMPLE 2

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Example\_003

{

class Program

{

static void Main(string[] args)

{

// Пример 2. Заказчик просит вас написать прогрмму, в которой

// пользователь вводит целое число start.

// Программа должна вывести таблицу квадратов чисел от start до start+10

Console.Write("\nВведите start:");

int start = Convert.ToInt32(Console.ReadLine());

int i = start;

while (i <= start + 10)

{

Console.WriteLine($"{i,4}^2 = {i \* i}");

i++;

}

}

}

}

EXAMPLE 3

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Example\_004

{

class Program

{

static void Main(string[] args)

{

// Пример 3. Заказчик просит вас написать прогрмму, в которой

// пользователь может ввести два целых числа start и end.

// Программа должна вывести таблицу квадратов чисел от start до end

// (!) Разница между start и end не может превышать 20

Console.Write("\nВведите start:");

int start = Convert.ToInt32(Console.ReadLine());

Console.Write("\nВведите end:");

int end = Convert.ToInt32(Console.ReadLine());

while (start <= end)

{

Console.WriteLine($"{start,4}^2 = {start \* start }");

start++;

}

}

}

}

EXAMPLE 4

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Example\_005

{

class Program

{

static void Main(string[] args)

{

// Пример 4. Заказчик прочит вас написать программу по подсчёту

// среднего балла полученных студентами на экзамене

// (i) Пользователь с клавиатуры вводит 10 чисел

// Требуется найти среднее арифметическое этих чисел

double value;

double average = 0;

int i = 1;

while (i <= 10)

{

Console.Write($"Введите {i}-й результат вступительного испытания: ");

value = double.Parse(Console.ReadLine());

average += value;

i++;

}

Console.WriteLine($"Среднее = {average / 10}");

}

}

}

EXAMPLE 5

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Example\_006

{

class Program

{

static void Main(string[] args)

{

// Пример 5. Пользователь вводит два целых числа numerator и denominator,

// которые являются числителем и знаменателем соответственно

// Нужно выяснить, можно ли сократить эту дробь

// и если можно, то на какое число

// 50 25 5 1

// ----- = ---- = ---- = ---

// 100 50 10 2

// https://ru.wikipedia.org/wiki/Наибольший\_общий\_делитель

// https://ru.wikipedia.org/wiki/Алгоритм\_Евклида

// Математика 5 класс. Тема "Сокращение дробей"

Console.Write("\nВведите numerator:");

double numerator = Convert.ToDouble(Console.ReadLine());

Console.Write("\nВведите denominator:");

double denominator = Convert.ToDouble(Console.ReadLine());

Console.WriteLine($"Исходная дробь {numerator}/{denominator}");

double a = numerator, b = denominator;

#region for

//for (int i = 0; i < 100000; i++)

//{

// if (a != b) if (a > b) a = a - b; else b = b - a;

//}

#endregion

while (a != b)

{

if (a > b) a = a - b; else b = b - a;

}

Console.WriteLine($"Можно сократить на {a}");

Console.WriteLine($"Сокращённая дробь {numerator / a}/{denominator / a}");

}

}

}

EXAMPLE 6

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Example\_007

{

class Program

{

static void Main(string[] args)

{

// Пример 6.

// Заказчик прочит вас написать игру "Угадай-ка", в которой

// игрок должен угадать число из диапазона от 1 до 100

// за минимальное количество шагов.

// При этом, после очередной попытки, игрок должен видеть сообщение

// "Введенное число больше загаданного. Попробуйте ещё раз"

// "Введенное число меньше загаданного. Попробуйте ещё раз"

// или "Поздравляю, число угадано! Число попыток: {число}.

Console.WriteLine("Игра \"Угадай-ка\"");

Random rand = new Random();

int iiNumber = rand.Next(1, 101);

int count = 0;

int userNumber;

//for (; ; )

while(true)

{

Console.Write("\nВведите число: ");

count++;

userNumber = Convert.ToInt32(Console.ReadLine());

if (userNumber < iiNumber)

{

Console.WriteLine("Введенное число меньше загаданного. Попробуйте ещё раз");

}

else if (userNumber > iiNumber)

{

Console.WriteLine("Введенное число больше загаданного. Попробуйте ещё раз");

}

else

{

Console.WriteLine($"Поздравляю, число угадано! Число попыток: {count}.");

Console.ReadKey();

break;

}

}

}

}

}

**3.10. DO WHILE LOOP**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Example\_001\_DoWhileLoop

{

class Program

{

static void Main(string[] args)

{

// Цикл do-while продолжает выполнение блока кода, только если условие продолжение истинно.

// Условие проверяется только в конце каждой итерации.

// В следствии чего, при использовании do-while, тело цикла быдет выполнено хотя бы один раз

bool condition = false;

do

{

// Тело цикла

} while (condition);

Console.WriteLine("Вывод чисел от 0 до 9");

int i = 0;

do

{

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

i++;

} while (i < 10);

Console.WriteLine("\n\nВывод чисел от -5 до 5");

i = -5;

do

{

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

i++;

} while (i <= 5);

Console.WriteLine("\n\nВывод чисел от 10 до 1. Вариант 1");

int j = 0;

do

{

Console.Write($"{10 - j} ");

j++;

} while (j < 10);

Console.WriteLine("\n\nВывод чисел от 10 до 1. Вариант 2");

int k = 10;

do

{

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

k--;

} while (k >= 1);

Console.WriteLine("\n\nВывод чисел от 1 до 22 c шагом 3");

int counter = 1;

do

{

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

counter += 3;

} while (counter <= 22);

Console.WriteLine("\n\nВывод чисел от -1 до 1 c шагом 0.25");

double step = -1;

do

{

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

step += 0.25;

} while (step <= 1);

Console.WriteLine("\n\nВывод чисел от 1 до -1 c шагом 0.5");

double interval = 1;

do

{

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

interval -= 0.5;

} while (interval >= -1);

#region Замечание 1

Console.ReadKey();

Console.Clear();

Console.WriteLine("\n\nЗамечание 1");

Console.WriteLine("\n\nВывод чисел от 1 до 22 c шагом 3");

counter = 28;

do

{

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

counter += 3;

} while (counter <= 22);

Console.WriteLine("\n\nВывод чисел от -1 до 1 c шагом 0.25");

step = 9;

do

{

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

step += 0.25;

} while (step <= 1);

Console.WriteLine("\n\nВывод чисел от 1 до -1 c шагом 0.5");

interval = -90;

do

{

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

interval -= 0.5;

} while (interval >= -1);

#endregion

#region Замечание 2

Console.ReadKey();

Console.Clear();

Console.WriteLine("\n\nЗамечание 2");

Console.WriteLine("\n\nВывод чисел от 1 до 22 c шагом 3");

counter = 28;

if (counter < 22)

{

do

{

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

counter += 3;

} while (counter <= 22);

}

Console.WriteLine("\n\nВывод чисел от -1 до 1 c шагом 0.25");

step = 9;

if (step == 1)

{

do

{

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

step += 0.25;

} while (step <= 1);

}

Console.WriteLine("\n\nВывод чисел от 1 до -1 c шагом 0.5");

interval = -90;

if (interval == 1)

{

do

{

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

interval -= 0.5;

} while (interval >= -1);

}

#endregion

Console.ReadKey();

}

}

}

EXAMPLE 1

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Example\_002

{

class Program

{

static void Main(string[] args)

{

// Пример 1. Сгенерировать 10 случайных чисел от -8 до 9

Random r = new Random();

int i = 0;

do

{

Console.Write($"{r.Next(-8, 10)} ");

i++;

} while (i < 10);

Console.ReadKey();

}

}

}

EXAMPLE 2

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Example\_003

{

class Program

{

static void Main(string[] args)

{

// Пример 2. Заказчик просит вас написать прогрмму, в которой

// пользователь вводит целое число start.

// Программа должна вывести таблицу квадратов чисел от start до start+10

Console.Write("\nВведите start:");

int start = Convert.ToInt32(Console.ReadLine());

int i = start;

do

{

Console.WriteLine($"{i,4}^2 = {i \* i}");

i++;

} while (i <= start + 10) ;

}

}

}

EXAMPLE 3

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Example\_004

{

class Program

{

static void Main(string[] args)

{

// Пример 3. Заказчик просит вас написать прогрмму, в которой

// пользователь может ввести два целых числа start и end.

// Программа должна вывести таблицу квадратов чисел от start до end

// (!) Разница между start и end не может превышать 20

Console.Write("\nВведите start:");

int start = Convert.ToInt32(Console.ReadLine());

Console.Write("\nВведите end:");

int end = Convert.ToInt32(Console.ReadLine());

if(start <= end)

do

{

Console.WriteLine($"{start,4}^2 = {start \* start }");

start++;

} while (start <= end);

}

}

}

EXAMPLE 4

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Example\_006

{

class Program

{

static void Main(string[] args)

{

// Пример 5. Пользователь вводит два целых числа numerator и denominator,

// которые являются числителем и знаменателем соответственно

// Нужно выяснить, можно ли сократить эту дробь

// и если можно, то на какое число

// 50 25 5 1

// ----- = ---- = ---- = ---

// 100 50 10 2

// https://ru.wikipedia.org/wiki/Наибольший\_общий\_делитель

// https://ru.wikipedia.org/wiki/Алгоритм\_Евклида

// Математика 5 класс. Тема "Сокращение дробей"

Console.Write("\nВведите numerator:");

double numerator = Convert.ToDouble(Console.ReadLine());

Console.Write("\nВведите denominator:");

double denominator = Convert.ToDouble(Console.ReadLine());

Console.WriteLine($"Исходная дробь {numerator}/{denominator}");

double a = numerator, b = denominator;

#region for

//for (int i = 0; i < 100000; i++)

//{

// if (a != b) if (a > b) a = a - b; else b = b - a;

//}

#endregion

if (a != b)

{

do

{

if (a > b) a = a - b; else b = b - a;

Console.WriteLine($">>> {a} {b}");

} while (a != b);

}

Console.WriteLine($"Можно сократить на {a}");

Console.WriteLine($"Сокращённая дробь {numerator / a}/{denominator / a}");

}

}

}

EXAMPLE 5

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Example\_007

{

class Program

{

static void Main(string[] args)

{

// Пример 6.

// Заказчик прочит вас написать игру "Угадай-ка", в которой

// игрок должен угадать число из диапазона от 1 до 100

// за минимальное количество шагов.

// При этом, после очередной попытки, игрок должен видеть сообщение

// "Введенное число больше загаданного. Попробуйте ещё раз"

// "Введенное число меньше загаданного. Попробуйте ещё раз"

// или "Поздравляю, число угадано! Число попыток: {число}.

Console.WriteLine("Игра \"Угадай-ка\"");

Random rand = new Random();

int iiNumber = rand.Next(1, 101);

int count = 0;

int userNumber;

//for (; ; )

//while (true)

do

{

Console.Write("\nВведите число: ");

count++;

userNumber = Convert.ToInt32(Console.ReadLine());

if (userNumber < iiNumber)

{

Console.WriteLine("Введенное число меньше загаданного. Попробуйте ещё раз");

}

else if (userNumber > iiNumber)

{

Console.WriteLine("Введенное число больше загаданного. Попробуйте ещё раз");

}

else

{

Console.WriteLine($"Поздравляю, число угадано! Число попыток: {count}.");

Console.ReadKey();

break;

}

}

while (true);

}

}

}

EXAMPLE 6

using System;

using System.Collections.Generic;

using System.Diagnostics;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Example\_008

{

class Program

{

static void Main(string[] args)

{

for (int i = 0; i < 100; i++)

{

if (i % 2 != 0) continue;

Debug.Write(">>> ");

Debug.Write($"{i} ");

}

Console.ReadKey();

}

}

}

**4.1. ARRAY**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Example\_001\_Array

{

class Program

{

static void Main(string[] args)

{

// 1. Одномерные массивы: инициализация, определения, использование, задание

#region Введение

//

// Массив - структура данных, хранящая набор значений (элементов массива),

// идентифицируемых по индексу или набору индексов,

//

// Массив данных имеет одинаковый тип данных.

// Чтобы объявить массив, типу данных добавляется набор квадратных скобок.

// Это будет означать, что выпланируете создать массив

// После этого указывается идентификатор (имя) массива.

// Массив может быть объявлен любым типом данных, и все его элементы будут иметь этот тип.

// Задание массива целых чисел

int[] x;

// Массив объявляется с ключевым словом new, за которым идёт информация о

// типе данных и количестве элементов массива.

// Это фиксированное количество, которые может содержать массив.

// Когда массив создан, элементы будут автоматически назначены данными по умолчанию для

// этого типа. (числа - нули, bool - false и т.д. )

x = new int[3]; // int-массив из 3 элементов

double[] array = new double[5]; // double-массив из 5 элементов

// Для заполнения элементов массива нужно указать его номер в квадратных скобках.

// Обратите внимание, что индекс первого элемента начинается с нуля.

x[0] = 1;

x[1] = 12;

x[2] = 223;

array[0] = 3;

array[1] = 5;

array[2] = 8;

array[3] = 13;

array[4] = 21;

//

int[] birthdayCollection = new int[8] { 2, 8, 0, 9, 1, 9, 9, 0 };

int[] newYearCollection = new int[] { 31, 12, 2319 };

string[] dateCollection = { "01", "01", "1771" };

////

//// ┌───────────────────────────────────────────┐

//// │ ┌────────┐ ┌────────┐ ┌────────┐ │

//// │ │ 31 │ │ 12 │ │ 1771 │ │

//// │ └────────┘ └────────┘ └────────┘ │

//// │ 0 1 2 │

//// └───────────────────────────────────────────┘

//// newYearCollection

////

//// birthdayCollection[0]

//// birthdayCollection[1]

//// birthdayCollection[2]

//// Индексы меняются от 0 до 2

///

#endregion

}

}

}

**EXAMPLES**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Example\_001\_ArrayExs

{

class Program

{

static void Main(string[] args)

{

#region Задача 1

//Console.WriteLine("Сформировать массив из N случайных чисел от -5 до 5");

//// Решение

//Console.Write("Введите n: "); // Приглашение ко вводу числа

//int n = int.Parse(Console.ReadLine()); // Ввод числа с клавиатуры

//Random random = new Random(); // Инициализация генератора псевдослучайных чисел

//int[] numbs = new int[n]; // Объявление и инициализация массива

//for (int i = 0; i < numbs.Length; i++) //

//{ //

// numbs[i] = random.Next(-5, 6); // Заполнение массива

// Console.Write($"{numbs[i]} "); // Вывод массива на экран

//} //

Console.ReadKey();

#endregion

#region Задача 2

//Console.Clear();

//Console.WriteLine("Сформировать массив N случайных чисел. Подсчитать количество четных чисел");

Random random = new Random();

//// Решение

//Console.Write("Введите n: "); // Приглашение ко вводу числа

//int n = int.Parse(Console.ReadLine()); // Ввод числа с клавиатуры

//var numbs = new int[n]; // Объявление и инициализация массива

//int evenElement = 0; // Количество четных элементов

//for (int i = 0; i < numbs.Length; i++) //

//{ //

// numbs[i] = random.Next(10); // Заполнение массива

// Console.Write($"{numbs[i]} "); // Вывод массива на экран

// if (numbs[i] % 2 == 0) evenElement++; // Подсчёт чётных элементов

//} //

// //

//Console.WriteLine($"\nЧётных: {evenElement}");

//Console.ReadKey();

#endregion

#region Задача 3

//Console.Clear();

//Console.WriteLine("В массиве из 10 чисел вычислить, сколько раз встречается введенное число");

//// Решение

//int[] numbs = new int[10]; // Объявление и инициализация массива

//for (int i = 0; i < numbs.Length; i++) //

//{ //

// numbs[i] = random.Next(10); // Заполнение массива

// Console.Write($"{numbs[i]} "); // Вывод массива на экран

//} //

//Console.Write("\nКакое число ищём: "); // Приглашение ко вводу числа

//int n = int.Parse(Console.ReadLine()); // Ввод искомого числа с клавиатуры

//int find = 0; // Результат поиска

//for (int i = 0; i < numbs.Length; i++) //

//{ //

// if (numbs[i] == n) find++; // Поиск искомого элемента

//}

//Console.WriteLine($"\nЭлемент '{n}' найден: {find} р. ");

//Console.ReadKey();

#endregion

#region Задача 4

Console.Clear();

Console.WriteLine("Массив из N элементов состоит из нулей и единиц.\nПоставить нули в начало, а единицы – в конец.");

// Решение

Console.Write("\nКоличество элементов: "); // Приглашение ко вводу числа

int n = int.Parse(Console.ReadLine()); // Ввод искомого числа с клавиатуры

int[] numbs = new int[n]; // Объявление и инициализация массива

Console.Write("\nИсходный массив: "); //

for (int i = 0; i < numbs.Length; i++) //

{ //

numbs[i] = random.Next(0, 2); // Заполнение массива

Console.Write($"{numbs[i]} "); // Вывод массива на экран

} //

// 0 0 0 0 0 0 0 1 1 1

// ^ ^

//

int head = 0; // Указатель головного элемента

int tail = numbs.Length - 1; // Указатель хвостового элемента

bool flag = true; // Флаг, показывающий увеличивать позицию головного - flag = true

// или уменьшать позицию хвостового - flag = false

while (head < tail) // Пока "голова" меньше "хвоста"

{

if(flag) // flag = true

{ //

if (numbs[head] == 1) flag = false; // Работаем с головным элементом

else head++; //

}

else // иначе flag = false

{ //

if (numbs[tail] == 0) flag = true; // Работаем с хвостовым элементом

else tail--; //

}

if(numbs[head]==1 && numbs[tail] == 0) // Если головной элемент = 1 и хвостовой = 0

{

int temp = numbs[head]; //

numbs[head] = numbs[tail]; // Меняем их местами

numbs[tail] = temp; //

head++; // Сдвигаем голову

tail--; // и хвост

}

}

Console.Write("\nПолучившийся массив: ");

for (int i = 0; i < numbs.Length; i++) //

{ //

Console.Write($"{numbs[i]} "); // Вывод массива на экран

}

Console.ReadKey();

#endregion

}

}

}

**4.2. ARRAY METHODS**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Example\_002\_ArrayMethods

{

class Program

{

static void Main(string[] args)

{

// 2. Основные методы работы с одномерными массивами

#region Задача 4

Console.Clear();

Console.WriteLine("Массив из N элементов состоит из нулей и единиц.\nПоставить нули в начало, а единицы – в конец.");

// Решение

Random random = new Random(); // Инициализация генератора псевдослучайных чисел

Console.Write("\nКоличество элементов: "); // Приглашение ко вводу числа

int n = int.Parse(Console.ReadLine()); // Ввод искомого числа с клавиатуры

int [] numbs = new int[n]; // Объявление и инициализация массива

Console.Write("\nИсходный массив: "); //

for (int i = 0; i < numbs.Length; i++) //

{ //

numbs[i] = random.Next(0, 2); // Заполнение массива

Console.Write($"{numbs[i]} "); // Вывод массива на экран

} //

// 0 1 0 0 0 0 1 0 1 0

// ^ ^

//

#region Old

//int head = 0; // Указатель головного элемента

//int tail = numbs.Length - 1; // Указатель хвостового элемента

//bool flag = true; // Флаг, показывающий увеличивать позицию головного - flag = true

// // или уменьшать позицию хвостового - flag = false

//while (head < tail) // Пока "голова" меньше "хвоста"

//{

// if (flag) // flag = true

// { //

// if (numbs[head] == 1) flag = false; // Работаем с головным элементом

// else head++; //

// }

// else // иначе flag = false

// { //

// if (numbs[tail] == 0) flag = true; // Работаем с хвостовым элементом

// else tail--; //

// }

// if (numbs[head] == 1 && numbs[tail] == 0) // Если головной элемент = 1 и хвостовой = 0

// {

// int temp = numbs[head]; //

// numbs[head] = numbs[tail]; // Меняем их местами

// numbs[tail] = temp; //

// head++; // Сдвигаем голову

// tail--; // и хвост

// }

//}

#endregion

#region ToDo

Array.Sort(numbs);

#endregion

Console.Write("\nПолучившийся массив: ");

for (int i = 0; i < numbs.Length; i++) //

{ //

Console.Write($"{numbs[i]} "); // Вывод массива на экран

}

Console.ReadKey();

#endregion

}

}

}

**EXAMPLES**

using System;

using System.Collections.Generic;

using System.Text;

using System.Threading.Tasks;

namespace Example\_002\_ArrayMethodsEx

{

class Program

{

static void Main(string[] args)

{

var data = new int[] { 1, 5, 2, 5, 4, 6, 5, 7, 6, 8, 6, 9, 5, 1 };

Console.WriteLine("Исходный массив: ");

for (int i = 0; i < data.Length; i++)

{

Console.Write($"{data[i]} ");

}

Console.WriteLine();

Console.WriteLine($"data.Length = {data.Length}"); // Получает общее число элементов во всех измерениях массива

int result = Array.IndexOf(data, 5); // Выполняет поиск указанного объекта внутри

// всего одномерного массива и возвращает индекс его первого вхождения.

Console.WriteLine($"Array.IndexOf(data, 5) = {result}");

result = Array.LastIndexOf(data, 5);

Console.WriteLine($"Array.LastIndexOf(data, 5) = {result}");

int rank = data.Rank; // Получает ранг (число измерений) массива.

Console.WriteLine($"data.Rank = {rank}");

Array.Sort(data); // Сортирует элементы во всем массиве

Console.WriteLine("\nМассив после применения Array.Sort(data): ");

for (int i = 0; i < data.Length; i++)

{

Console.Write($"{data[i]} ");

}

Console.WriteLine();

Array.Reverse(data); // Изменяет порядок элементов во всем одномерном массиве на обратный.

Console.WriteLine("Массив после применения Array.Reverse(data): ");

for (int i = 0; i < data.Length; i++)

{

Console.Write($"{data[i]} ");

}

Console.WriteLine();

Array.Clear(data, 7, 3); // Задает для диапазона элементов в массиве значение, предусмотренное по умолчанию

Console.WriteLine("Массив после применения Array.Clear(data, 7, 3): ");

for (int i = 0; i < data.Length; i++)

{

Console.Write($"{data[i]} ");

}

Console.WriteLine();

int[] copyData = new int[20];

Array.ConstrainedCopy(data, 2, copyData, 10, 5);

// Копирует диапазон элементов из массива (data), начиная с заданного индекса (2)

// источника, и вставляет его в другой массив (copyData) , начиная с заданного

// индекса (10) назначения. Гарантирует, что в случае невозможности успешно скопировать

// весь диапазон, все изменения будут отменены.

// 5 - количество копируемых элементов

//Array.Copy()

Console.WriteLine("Массив copyData: ");

for (int i = 0; i < copyData.Length; i++)

{

Console.Write($"{copyData[i]} ");

}

Console.WriteLine();

Console.WriteLine("Массив copyData после применения Array.ConstrainedCopy(data, 4, copyData, 10, 5): ");

for (int i = 0; i < copyData.Length; i++)

{

Console.Write($"{copyData[i]} ");

}

Console.WriteLine("\n");

}

}

}

**4.3. LOOP**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Example\_003\_Loop

{

class Program

{

static void Main(string[] args)

{

// 3. Вложенные циклы, операторы break и continue

#region Распечатать таблицу умножения

// 2 x 1 = 2 3 x 1 = 3 4 x 1 = 4

// 2 x 2 = 4 3 x 2 = 6 4 x 2 = 8

// 2 x 3 = 6 3 x 3 = 9 4 x 3 = 12

// 2 x 4 = 8 3 x 4 = 12 4 x 4 = 16

// 2 x 5 = 10 3 x 5 = 15 4 x 5 = 20

// 2 x 6 = 12 3 x 6 = 18 4 x 6 = 24

// 2 x 7 = 14 3 x 7 = 21 4 x 7 = 28

// 2 x 8 = 16 3 x 8 = 24 4 x 8 = 32

// 2 x 9 = 18 3 x 9 = 27 4 x 9 = 36

// 2 x 10 = 20 3 x 10 = 30 4 x 10 = 40

//for (int i = 1; i <= 10; i++)

//{

// Console.WriteLine($"2 x {i} = {2 \* i}");

//}

//Console.WriteLine();

//for (int i = 1; i <= 10; i++)

//{

// Console.WriteLine($"3 x {i} = {3 \* i}");

//}

//Console.WriteLine();

//for (int i = 1; i <= 10; i++)

//{

// Console.WriteLine($"4 x {i} = {4 \* i}");

//}

Console.WriteLine();

#region ToDo

//for (int i = 2; i < 10; i++)

//{

// for (int j = 1; j <= 10; j++)

// {

// Console.WriteLine($"{i} x {j} = {i \* j} ");

// }

// Console.WriteLine();

//}

#endregion

#endregion

#region Показать всевозможные трехбуквенные слова из алфавита S K I L

//4 \* 4 \* 4 = 4^3

#region ToDo

//char[] symbols = { 'S', 'K', 'I', 'L' };

//int count = 1;

//for (int i = 0; i < symbols.Length; i++)

//{

// for (int j = 0; j < symbols.Length; j++)

// {

// for (int k = 0; k < symbols.Length; k++)

// {

// Console.WriteLine($"{count++,3}: {symbols[i]}{symbols[j]}{symbols[k]}");

// }

// }

//}

#endregion

#endregion

#region Найти хотя бы одно решение уравнения в целых числах

// ┌ ┐

// │ ┌ ┐ 3 ┌ ┐ 2 │

// │ 2 × │ a │ + 3 × │ b │ │ = 8337

// │ └ ┘ └ ┘ │

// └ ┘

// 2 \* 1^3 + 3 \* 1^2

// 2 \* 1^3 + 3 \* 2^2

// 2 \* 1^3 + 3 \* 3^2

// 2 \* 1^3 + 3 \* 4^2

// 2 \* 1^3 + 3 \* 5^2

// ...

// 2 \* 2^3 + 3 \* 1^2

// 2 \* 2^3 + 3 \* 2^2

// 2 \* 2^3 + 3 \* 3^2

// 2 \* 2^3 + 3 \* 4^2

// 2 \* 2^3 + 3 \* 5^2

// ...

// 2 \* 3^3 + 3 \* 1^2

// 2 \* 3^3 + 3 \* 2^2

// 2 \* 3^3 + 3 \* 3^2

// 2 \* 3^3 + 3 \* 4^2

// 2 \* 3^3 + 3 \* 5^2

#region ToDo

//for (double a = -500; a < 500; a++)

//{

// for (double b = -500; b < 500; b++)

// {

// if ((2 \* a \* a \* a + 3 \* b \* b) == 8337) Console.WriteLine($"a = {a} b = {b}");

// }

//}

#endregion

#endregion

#region break и continue

int count = 0;

for (int i = 0; i < 1\_000; i++)

{

//if (i % 2 == 0) continue;

for (int j = 0; j < 1\_000; j++)

{

break;

//Console.Write($"({i} {j}) ");

}

count++;

}

Console.WriteLine($"count = {count}");

Console.WriteLine();

#endregion

}

}

}

**4.4. ARRAY 2D**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Example\_004\_Array2D

{

class Program

{

static void Main(string[] args)

{

// 4. Двумерные массивы: использование, задание

// int[] collection = new int[3];

//// ┌───────────────────────────────────────────┐

//// │ ┌────────┐ ┌────────┐ ┌────────┐ │

//// │ │ 31 │ │ 12 │ │ 1771 │ │

//// │ └────────┘ └────────┘ └────────┘ │

//// │ 0 1 2 │

//// └───────────────────────────────────────────┘

//// 0 1 2 3

//// ┌─────────────────────────┐

//// │ ┌───┐ ┌───┐ ┌───┐ ┌───┐ │

//// 0 │ │ 1 │ │ 3 │ │ 0 │ │ 1 │ │

//// │ └───┘ └───┘ └───┘ └───┘ │

//// │ ┌───┐ ┌───┐ ┌───┐ ┌───┐ │

//// 1 │ │ 1 │ │ 0 │ │ 3 │ │ 1 │ │

//// │ └───┘ └───┘ └───┘ └───┘ │

//// │ ┌───┐ ┌───┐ ┌───┐ ┌───┐ │

//// 2 │ │ 2 │ │ 0 │ │ 1 │ │ 9 │ │

//// │ └───┘ └───┘ └───┘ └───┘ │

//// └─────────────────────────┘

//

int[,] matrix = new int[3, 4]; // Задание двумерного массива, в котором 3 строки и 4 столбца

matrix[1, 2] = 3;

Random r = new Random();

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < 4; j++)

{

matrix[i, j] = r.Next(10);

Console.Write($"{matrix[i, j]} ");

}

Console.WriteLine();

}

}

}

}

**EXAMPLE 1**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Example\_004\_Array2D\_Demo

{

class Program

{

static void Main(string[] args)

{

Console.Write("Введите количество строк: ");

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

Console.Write("Введите количество столбцов: ");

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

int[,] arr2d = new int[row, col];

Random r = new Random();

for (int i = 0; i < row; i++)

{

for (int j = 0; j < col; j++)

{

arr2d[i, j] = r.Next(10);

Console.Write($"{arr2d[i, j]} ");

}

Console.WriteLine();

}

Console.WriteLine();

string[,] table = new string[,] {

{ "00", "01", "02", "03" },

{ "10", "11", "12", "13" },

{ "20", "21", "22", "23" }

};

Console.WriteLine();

Console.WriteLine($"table.Rank = {arr2d.Rank}");

Console.WriteLine($"table.GetLength(0) = {arr2d.GetLength(0)}");

Console.WriteLine($"table.GetLength(1) = {arr2d.GetLength(1)}");

for (int i = 0; i < table.GetLength(0); i++)

{

for (int j = 0; j < table.GetLength(1); j++)

{

Console.Write($"{table[i, j],3} ");

}

Console.WriteLine();

}

}

}

}

**EXAMPLE 2**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Example\_004\_Array2D\_Exs

{

class Program

{

static void Main(string[] args)

{

#region Элементы одномерного массива А из 9 чисел построчно расположить в матрице B[3×3]

//int[] a = { 1, 2, 3, 4, 5, 6, 7, 8, 9 };

//int[,] b = new int[3, 3];

//int counterA = 0;

//for (int i = 0; i < b.GetLength(0); i++)

//{

// for (int j = 0; j < b.GetLength(1); j++)

// {

// b[i, j] = a[counterA++];

// Console.Write($"{b[i, j]} ");

// }

// Console.WriteLine();

//}

#endregion

#region В матрице Z[3×3] каждый элемент разделить на диагональный, стоящий в том же столбце.

//

// 1 2 3

// 4 5 6

// 7 8 9

//Random r = new Random();

//var Z = new double[3, 3];

//Console.WriteLine("Исходная матрица Z[3x3]: ");

//for (int i = 0; i < Z.GetLength(0); i++)

//{

// for (int j = 0; j < Z.GetLength(1); j++)

// {

// Z[i, j] = r.Next(1, 10);

// Console.Write($"{Z[i, j],6} ");

// }

// Console.WriteLine();

//}

//Console.WriteLine("Матрица Z[3x3] после преобразования: ");

//for (int i = 0; i < Z.GetLength(0); i++)

//{

// double div = Z[i, i];

// for (int j = 0; j < Z.GetLength(1); j++)

// {

// Z[i, j] /= div;

// Console.Write($"{Z[i, j],6} ");

// }

// Console.WriteLine();

//}

#endregion

#region Для каждой строки матрицы A[4×4] найти сумму элементов.

Random r = new Random();

var A = new int[4, 4];

Console.WriteLine("Исходная матрица A[4x4]: ");

for (int i = 0; i < A.GetLength(0); i++)

{

int sum = 0;

for (int j = 0; j < A.GetLength(1); j++)

{

A[i, j] = r.Next(1, 3);

sum += A[i, j];

Console.Write($"{A[i, j],2} ");

}

Console.WriteLine($" : {sum}");

}

#endregion

}

}

}

**4.5. MULTI ARRAY**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Example\_005\_MultiArray

{

class Program

{

static void Main(string[] args)

{

// 5. Многомерные массивы: использование, задание

int[,,] matrix3D = new int[3, 4, 5];

Console.WriteLine($"matrix3D.Rank = {matrix3D.Rank}");

Console.WriteLine($"matrix3D.Length = {matrix3D.Length}");

Console.WriteLine($"matrix3D.GetLength(0) = {matrix3D.GetLength(0)}");

Console.WriteLine($"matrix3D.GetLength(1) = {matrix3D.GetLength(1)}");

Console.WriteLine($"matrix3D.GetLength(2) = {matrix3D.GetLength(2)}");

Random r = new Random();

for (int i = 0; i < matrix3D.GetLength(0); i++)

{

for (int j = 0; j < matrix3D.GetLength(1); j++)

{

for (int k = 0; k < matrix3D.GetLength(2); k++)

{

matrix3D[i, j, k] = r.Next(10);

Console.Write($"{matrix3D[i, j, k]} ");

}

}

}

Console.WriteLine("\n\n");

int[,,,] matrix4D = new int[3, 4, 5, 6];

Console.WriteLine($"matrix4D.Rank = {matrix4D.Rank}");

Console.WriteLine($"matrix4D.Length = {matrix4D.Length}");

Console.WriteLine($"matrix4D.GetLength(0) = {matrix4D.GetLength(0)}");

Console.WriteLine($"matrix4D.GetLength(1) = {matrix4D.GetLength(1)}");

Console.WriteLine($"matrix4D.GetLength(2) = {matrix4D.GetLength(2)}");

Console.WriteLine($"matrix4D.GetLength(3) = {matrix4D.GetLength(3)}");

for (int i = 0; i < matrix4D.GetLength(0); i++)

{

for (int j = 0; j < matrix4D.GetLength(1); j++)

{

for (int k = 0; k < matrix4D.GetLength(2); k++)

{

for (int l = 0; l < matrix4D.GetLength(3); l++)

{

matrix4D[i, j, k, l] = r.Next(10);

Console.Write($"{matrix4D[i, j, k, l]} ");

}

}

}

}

Console.WriteLine("\n\n");

int[,,,,] matrix5D = new int[3, 4, 5, 6, 10];

Console.WriteLine($"matrix5D.Rank = {matrix5D.Rank}");

Console.WriteLine($"matrix5D.Length = {matrix5D.Length}");

Console.WriteLine($"matrix5D.GetLength(0) = {matrix5D.GetLength(0)}");

Console.WriteLine($"matrix5D.GetLength(1) = {matrix5D.GetLength(1)}");

Console.WriteLine($"matrix5D.GetLength(2) = {matrix5D.GetLength(2)}");

Console.WriteLine($"matrix5D.GetLength(3) = {matrix5D.GetLength(3)}");

Console.WriteLine($"matrix5D.GetLength(4) = {matrix5D.GetLength(4)}");

Console.WriteLine("\n\n");

Console.ReadKey();

double[,,,,,] matrix6D = new double[30, 43, 55, 61, 10, 7];

Console.WriteLine($"matrix6D.Rank = {matrix6D.Rank}");

Console.WriteLine($"matrix6D.Length = {matrix6D.Length}");

Console.WriteLine($"memory: {(matrix6D.Length \* sizeof(double)) }");

Console.WriteLine($"matrix6D.GetLength(0) = {matrix6D.GetLength(0)}");

Console.WriteLine($"matrix6D.GetLength(1) = {matrix6D.GetLength(1)}");

Console.WriteLine($"matrix6D.GetLength(2) = {matrix6D.GetLength(2)}");

Console.WriteLine($"matrix6D.GetLength(3) = {matrix6D.GetLength(3)}");

Console.WriteLine($"matrix6D.GetLength(4) = {matrix6D.GetLength(4)}");

Console.WriteLine($"matrix6D.GetLength(4) = {matrix6D.GetLength(5)}");

}

}

}

**4.6. JAGGED ARRAY**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Example\_006\_JaggedArray

{

class Program

{

static void Main(string[] args)

{

// 6. Массив массивов

//// 0 1 2 3

//// ┌─────────────────────────┐

//// │ ┌───┐ ┌───┐ ┌───┐ ┌───┐ │

//// 0 │ │ 1 │ │ 3 │ │ 0 │ │ 1 │ │

//// │ └───┘ └───┘ └───┘ └───┘ │

//// │ ┌───┐ ┌───┐ ┌───┐ ┌───┐ │

//// 1 │ │ 1 │ │ 0 │ │ 3 │ │ 1 │ │

//// │ └───┘ └───┘ └───┘ └───┘ │

//// │ ┌───┐ ┌───┐ ┌───┐ ┌───┐ │

//// 2 │ │ 2 │ │ 0 │ │ 1 │ │ 9 │ │

//// │ └───┘ └───┘ └───┘ └───┘ │

//// └─────────────────────────┘

//// 0 1 2 3

//// ┌─────────────┐

//// │ ┌───┐ ┌───┐ │

//// 0 │ │ 1 │ │ 3 │ │

//// │ └───┘ └───┘ └───────────┐

//// │ ┌───┐ ┌───┐ ┌───┐ ┌───┐ │

//// 1 │ │ 1 │ │ 0 │ │ 3 │ │ 1 │ │

//// │ └───┘ └───┘ └───┘ └───┘ │

//// │ ┌───┐ ┌───┐ ┌───┐ ┌─────┘

//// 2 │ │ 2 │ │ 0 │ │ 1 │ │

//// │ └───┘ └───┘ └───┘ │

//// └───────────────────┘

// Зубчатые массивы являются массивами массивов

// Они могут иметь неправильные размеры.

// Размеры распределяются по одному за раз

//int[,]

int[][] jaggedArray = new int[3][];

jaggedArray[0] = new int[2];

//jaggedArray[0,1]

jaggedArray[0][0] = 1;

jaggedArray[0][1] = 3;

jaggedArray[1] = new int[4];

jaggedArray[1][0] = 1;

jaggedArray[1][1] = 0;

jaggedArray[1][2] = 3;

jaggedArray[1][3] = 1;

jaggedArray[2] = new int[3];

jaggedArray[2][0] = 2;

jaggedArray[2][1] = 0;

jaggedArray[2][2] = 1;

for (int i = 0; i < jaggedArray.Length; i++)

{

for (int j = 0; j < jaggedArray[i].Length; j++)

{

Console.Write($"{jaggedArray[i][j]} ");

}

Console.WriteLine();

}

}

}

}

**EXAMPLES**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Example\_006\_JaggedArrayExs

{

class Program

{

static void Main(string[] args)

{

// Написать программу, моделирующую получение показаний метоостанции

// Собирать и хранить информацию о температуре воздуха

// В период с 9 ч утра до 21 ч вечера каждые 5 минут

// В период с 21 ч вечера до 9 утра каждые 15 минут

#region int[,] dataTime

//Random r = new Random();

//int[,] dataTime = new int[24, 12];

//for (int i = 0; i < dataTime.GetLength(0); i++)

//{

// Console.Write($"{i,3}ч. ");

// for (int j = 0; j < dataTime.GetLength(1); j++)

// {

// if (i >= 9 && i < 21) dataTime[i, j] = r.Next(20);

// else dataTime[i, j] = (j < 4) ? r.Next(20) : 0;

// Console.Write($"{dataTime[i, j],3} ");

// }

// Console.WriteLine();

//}

//Console.ReadKey();

//Console.Clear();

//for (int i = 0; i < dataTime.GetLength(0); i++)

//{

// Console.ForegroundColor = ConsoleColor.White;

// Console.Write($"{i,3}ч. ");

// for (int j = 0; j < dataTime.GetLength(1); j++)

// {

// if (i >= 9 && i < 21) Console.ForegroundColor = ConsoleColor.White;

// else Console.ForegroundColor = (j < 4) ? ConsoleColor.White : ConsoleColor.Red;

// Console.Write($"{dataTime[i, j],3} ");

// }

// Console.WriteLine();

//}

//Console.ForegroundColor = ConsoleColor.White;

#endregion

#region int[][] dataTime

//Random r = new Random();

//int[][] dataTime = new int[24][];

//for (int j = 0; j < dataTime.Length; j++)

//{

// dataTime[j] = (j >= 9 && j < 21) ? new int[12] : new int[4];

//}

//for (int i = 0; i < dataTime.Length; i++)

//{

// Console.Write($"{i} ч. ");

// for (int j = 0; j < dataTime[i].Length; j++)

// {

// dataTime[i][j] = r.Next(20);

// Console.Write($"{dataTime[i][j],3} ");

// }

// Console.WriteLine();

//}

#endregion

}

}

}

**4.7. FOR EACH**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Example\_007\_foreach

{

class Program

{

static void Main(string[] args)

{

// 7. Цикл foreach

#region Задача 1

//Console.WriteLine("Сформировать массив из N случайных чисел от -5 до 5");

////Решение

//Console.Write("Введите n: "); // Приглашение ко вводу числа

//int n = int.Parse(Console.ReadLine()); // Ввод числа с клавиатуры

//Random random = new Random(); // Инициализация генератора псевдослучайных чисел

//int[] numbs = new int[n]; // Объявление и инициализация массива

//for (int i = 0; i < numbs.Length; i++) //

//{ //

// numbs[i] = random.Next(-5, 6); // Заполнение массива

//} //

//for (int i = 0; i < numbs.Length; i++) //

//{ //

// Console.Write($"{numbs[i]} "); // Вывод массива на экран

//} //

//Console.WriteLine();

//foreach (var e in numbs)

//{

// Console.Write($"{e} ");

//}

//Console.ReadKey();

#endregion

#region int[][] dataTime

//int[][] dataTime = new int[24][];

////int[][][] vs = new int[3][][];

////vs[0] = new int[3][];

//for (int j = 0; j < dataTime.Length; j++)

//{

// dataTime[j] = (j >= 9 && j < 21) ? new int[12] : new int[4];

//}

////for (int i = 0; i < dataTime.Length; i++)

////{

//// for (int j = 0; j < dataTime[i].Length; j++)

//// {

//// Console.Write($"{dataTime[i][j],3} ");

//// }

//// Console.WriteLine();

////}

//foreach (var array in dataTime)

//{

// foreach (var e in array)

// {

// Console.Write($"{e,3} ");

// }

// Console.WriteLine();

//}

#endregion

#region String

//string message = "Я С#-разработчик";

//for (int i = 0; i < message.Length; i++)

//{

// Console.Write($"{message[i]} ");

//}

//Console.WriteLine();

//foreach (var c in message)

//{

// Console.Write($"{c} ");

//}

Console.WriteLine();

Console.ReadKey();

#endregion

}

}

}

**4.8. HOMEWORK**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Homework\_Theme\_04

{

class Program

{

static void Main(string[] args)

{

// Задание 1.

// Заказчик просит вас написать приложение по учёту финансов

// и продемонстрировать его работу.

// Суть задачи в следующем:

// Руководство фирмы по 12 месяцам ведет учет расходов и поступлений средств.

// За год получены два массива – расходов и поступлений.

// Определить прибыли по месяцам

// Количество месяцев с положительной прибылью.

// Добавить возможность вывода трех худших показателей по месяцам, с худшей прибылью,

// если есть несколько месяцев, в некоторых худшая прибыль совпала - вывести их все.

// Организовать дружелюбный интерфейс взаимодействия и вывода данных на экран

// Пример

//

// Месяц Доход, тыс. руб. Расход, тыс. руб. Прибыль, тыс. руб.

// 1 100 000 80 000 20 000

// 2 120 000 90 000 30 000

// 3 80 000 70 000 10 000

// 4 70 000 70 000 0

// 5 100 000 80 000 20 000

// 6 200 000 120 000 80 000

// 7 130 000 140 000 -10 000

// 8 150 000 65 000 85 000

// 9 190 000 90 000 100 000

// 10 110 000 70 000 40 000

// 11 150 000 120 000 30 000

// 12 100 000 80 000 20 000

//

// Худшая прибыль в месяцах: 7, 4, 1, 5, 12

// Месяцев с положительной прибылью: 10

// \* Задание 2

// Заказчику требуется приложение строящее первых N строк треугольника паскаля. N < 25

//

// При N = 9. Треугольник выглядит следующим образом:

// 1

// 1 1

// 1 2 1

// 1 3 3 1

// 1 4 6 4 1

// 1 5 10 10 5 1

// 1 6 15 20 15 6 1

// 1 7 21 35 35 21 7 1

//

//

// Простое решение:

// 1

// 1 1

// 1 2 1

// 1 3 3 1

// 1 4 6 4 1

// 1 5 10 10 5 1

// 1 6 15 20 15 6 1

// 1 7 21 35 35 21 7 1

//

// Справка: https://ru.wikipedia.org/wiki/Треугольник\_Паскаля

//

// \* Задание 3.1

// Заказчику требуется приложение позволяющщее умножать математическую матрицу на число

// Справка https://ru.wikipedia.org/wiki/Матрица\_(математика)

// Справка https://ru.wikipedia.org/wiki/Матрица\_(математика)#Умножение\_матрицы\_на\_число

// Добавить возможность ввода количество строк и столцов матрицы и число,

// на которое будет производиться умножение.

// Матрицы заполняются автоматически.

// Если по введённым пользователем данным действие произвести невозможно - сообщить об этом

//

// Пример

//

// | 1 3 5 | | 5 15 25 |

// 5 х | 4 5 7 | = | 20 25 35 |

// | 5 3 1 | | 25 15 5 |

//

//

// \*\* Задание 3.2

// Заказчику требуется приложение позволяющщее складывать и вычитать математические матрицы

// Справка https://ru.wikipedia.org/wiki/Матрица\_(математика)

// Справка https://ru.wikipedia.org/wiki/Матрица\_(математика)#Сложение\_матриц

// Добавить возможность ввода количество строк и столцов матрицы.

// Матрицы заполняются автоматически

// Если по введённым пользователем данным действие произвести невозможно - сообщить об этом

//

// Пример

// | 1 3 5 | | 1 3 4 | | 2 6 9 |

// | 4 5 7 | + | 2 5 6 | = | 6 10 13 |

// | 5 3 1 | | 3 6 7 | | 8 9 8 |

//

//

// | 1 3 5 | | 1 3 4 | | 0 0 1 |

// | 4 5 7 | - | 2 5 6 | = | 2 0 1 |

// | 5 3 1 | | 3 6 7 | | 2 -3 -6 |

//

// \*\*\* Задание 3.3

// Заказчику требуется приложение позволяющщее перемножать математические матрицы

// Справка https://ru.wikipedia.org/wiki/Матрица\_(математика)

// Справка https://ru.wikipedia.org/wiki/Матрица\_(математика)#Умножение\_матриц

// Добавить возможность ввода количество строк и столцов матрицы.

// Матрицы заполняются автоматически

// Если по введённым пользователем данным действие произвести нельзя - сообщить об этом

//

// | 1 3 5 | | 1 3 4 | | 22 48 57 |

// | 4 5 7 | х | 2 5 6 | = | 35 79 95 |

// | 5 3 1 | | 3 6 7 | | 14 36 45 |

//

//

// | 4 |

// | 1 2 3 | х | 5 | = | 32 |

// | 6 |

//

}

}

}