**Отчёт**

**Муковнин.Д.Ю**

**Практическая работа №3**

**Цель**: овладеть навыками разработки программ по алгоритмам условной структуры.

Выполнить задания из блока **Практические задания** создавая для каждой задачи новый проект консольного приложения C#

**Практические задания**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | B | И (A && B ) | ИЛИ (A || B ) | НЕ (!A) |
| false | false | false | false | true |
| false | true | false | true | true |
| true | false | false | true | false |
| true | true | true | true | false |

* Вычислить значение логического выражения при всех возможных значениях логических величин А и В:

а) не (не А и не В) и А;

using System;

public class Program

{

public static void Main()

{

Console.WriteLine("Задание 1а: не (не А и не В) и А");

Console.WriteLine("Введите значение A (true/false):");

bool A = bool.Parse(Console.ReadLine());

Console.WriteLine("Введите значение B (true/false):");

bool B = bool.Parse(Console.ReadLine());

bool result = !(!A && !B) && A;

Console.WriteLine($"не (не {A} и не {B}) и {A} = {result}");

}

}

б) не (не А или не В) или А;

using System;

public class Program

{

public static void Main()

{

Console.WriteLine("Задание 1б: не (не А или не В) или А");

Console.WriteLine("Введите значение A (true/false):");

bool A = bool.Parse(Console.ReadLine());

Console.WriteLine("Введите значение B (true/false):");

bool B = bool.Parse(Console.ReadLine());

bool result = !(!A || !B) || A;

Console.WriteLine($"не (не {A} или не {B}) или {A} = {result}");

}

}

в) не (не А или не В) и В.

using System;

public class Program

{

public static void Main()

{

Console.WriteLine("Задание 1в: не (не А или не В) и В");

Console.WriteLine("Введите значение A (true/false):");

bool A = bool.Parse(Console.ReadLine());

Console.WriteLine("Введите значение B (true/false):");

bool B = bool.Parse(Console.ReadLine());

bool result = !(!A || !B) && B;

Console.WriteLine($"не (не {A} или не {B}) и {B} = {result}");

}

}

* Вычислить значение логического выражения при всех возможных значениях логических величин А, В и С:

а) не (А или не В и С);

using System;

public class Program

{

public static void Main()

{

Console.WriteLine("Задание 2а: не (А или не В и С)");

Console.Write("Введите A (true/false): ");

bool A = bool.Parse(Console.ReadLine());

Console.Write("Введите B (true/false): ");

bool B = bool.Parse(Console.ReadLine());

Console.Write("Введите C (true/false): ");

bool C = bool.Parse(Console.ReadLine());

bool result = !(A || !B && C);

Console.WriteLine($"не ({A} или не {B} и {C}) = {result}");

}

}

б) А и не (В и А или не С);

using System;

public class Program

{

public static void Main()

{

Console.WriteLine("Задание 2б: А и не (В и А или не С)");

Console.Write("Введите A (true/false): ");

bool A = bool.Parse(Console.ReadLine());

Console.Write("Введите B (true/false): ");

bool B = bool.Parse(Console.ReadLine());

Console.Write("Введите C (true/false): ");

bool C = bool.Parse(Console.ReadLine());

bool result = A && !(B && A || !C);

Console.WriteLine($"{A} и не ({B} и {A} или не {C}) = {result}");

}

}

в) не (не А или В и С)

using System;

public class Program

{

public static void Main()

{

Console.WriteLine("Задание 2в: не (не А или В и С)");

Console.Write("Введите A (true/false): ");

bool A = bool.Parse(Console.ReadLine());

Console.Write("Введите B (true/false): ");

bool B = bool.Parse(Console.ReadLine());

Console.Write("Введите C (true/false): ");

bool C = bool.Parse(Console.ReadLine());

bool result = !(!A || B && C);

Console.WriteLine($"не (не {A} или {B} и {C}) = {result}");

}

}

г) не (А или не В и С) или С;

using System;

public class Program

{

public static void Main()

{

Console.WriteLine("Задание 2г: не (А или не В и С) или С");

Console.Write("Введите A (true/false): ");

bool A = bool.Parse(Console.ReadLine());

Console.Write("Введите B (true/false): ");

bool B = bool.Parse(Console.ReadLine());

Console.Write("Введите C (true/false): ");

bool C = bool.Parse(Console.ReadLine());

bool result = !(A || !B && C) || C;

Console.WriteLine($"не ({A} или не {B} и {C}) или {C} = {result}");

}

}

д) не (А и не В или С) и В;

using System;

public class Program

{

public static void Main()

{

Console.WriteLine("Задание 2д: не (А и не В или С) и В");

Console.Write("Введите A (true/false): ");

bool A = bool.Parse(Console.ReadLine());

Console.Write("Введите B (true/false): ");

bool B = bool.Parse(Console.ReadLine());

Console.Write("Введите C (true/false): ");

bool C = bool.Parse(Console.ReadLine());

bool result = !(A && !B || C) && B;

Console.WriteLine($"не ({A} и не {B} или {C}) и {B} = {result}");

}

}

е) не (не А или В и С) или А.

using System;

public class Program

{

public static void Main()

{

Console.WriteLine("Задание 2е: не (не А или В и С) или А");

Console.Write("Введите A (true/false): ");

bool A = bool.Parse(Console.ReadLine());

Console.Write("Введите B (true/false): ");

bool B = bool.Parse(Console.ReadLine());

Console.Write("Введите C (true/false): ");

bool C = bool.Parse(Console.ReadLine());

bool result = !(!A || B && C) || A;

Console.WriteLine($"не (не {A} или {B} и {C}) или {A} = {result}");

}

}

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| X | y | ∨ (x || y) | ∧ (x && y) | **→** (!x || y ) | ≡ ((!x || y) && (!y || x))  ≡(x == y) |
| False | false | false | false | true | true |
| False | true | true | false | true | false |
| True | false | true | false | false | false |
| True | true | true | true | true | true |

* Вычислить значение логического выражения при всех возможных значениях логических величин X и Y:

а) (x ∨ y) → (у ≡ x)

using System;

public class Program

{

public static void Main()

{

Console.WriteLine("Задание 3а: (x ∨ y) → (у ≡ x)");

Console.Write("Введите X (true/false): ");

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

Console.Write("Введите Y (true/false): ");

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

bool result = !(x || y) || (y == x);

Console.WriteLine($"({x} ∨ {y}) → ({y} ≡ {x}) = {result}");

}

}

б) (x ∨ y) ≡ (у ∧ x)

using System;

public class Program

{

public static void Main()

{

Console.WriteLine("Задание 3б: (x ∨ y) ≡ (у ∧ x)");

Console.Write("Введите X (true/false): ");

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

Console.Write("Введите Y (true/false): ");

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

bool result = (x || y) == (y && x);

Console.WriteLine($"({x} ∨ {y}) ≡ ({y} ∧ {x}) = {result}");

}

}

в) (x ∧ y) ∨ (у → x)

using System;

public class Program

{

public static void Main()

{

Console.WriteLine("Задание 3в: (x ∧ y) ∨ (у → x)");

Console.Write("Введите X (true/false): ");

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

Console.Write("Введите Y (true/false): ");

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

bool result = (x && y) || (!y || x);

Console.WriteLine($"({x} ∧ {y}) ∨ ({y} → {x}) = {result}");

}

}

г) (x ≡ y) → у

using System;

public class Program

{

public static void Main()

{

Console.WriteLine("Задание 3г: (x ≡ y) → у");

Console.Write("Введите X (true/false): ");

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

Console.Write("Введите Y (true/false): ");

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

bool result = !(x == y) || y;

Console.WriteLine($"({x} ≡ {y}) → {y} = {result}");

}

}

д) (x ∨ y) → (z ≡ x);

using System;

public class Program

{

public static void Main()

{

Console.WriteLine("Задание 3д: (x ∨ y) → (z ≡ x)");

Console.Write("Введите X (true/false): ");

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

Console.Write("Введите Y (true/false): ");

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

Console.Write("Введите Z (true/false): ");

bool z = bool.Parse(Console.ReadLine());

bool result = !(x || y) || (z == x);

Console.WriteLine($"({x} ∨ {y}) → ({z} ≡ {x}) = {result}");

}

}

е) (x ≡ (z ∨ y)) ∨ ((x→ z) ∧ (y → x));

using System;

public class Program

{

public static void Main()

{

Console.WriteLine("Задание 3е: (x ≡ (z ∨ y)) ∨ ((x→ z) ∧ (y → x))");

Console.Write("Введите X (true/false): ");

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

Console.Write("Введите Y (true/false): ");

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

Console.Write("Введите Z (true/false): ");

bool z = bool.Parse(Console.ReadLine());

bool part1 = x == (z || y);

bool part2 = (!x || z) && (!y || x);

bool result = part1 || part2;

Console.WriteLine($"({x} ≡ ({z} ∨ {y})) ∨ (({x}→{z}) ∧ ({y}→{x})) = {result}");

}

}

ж) y → (z ≡ x);

using System;

public class Program

{

public static void Main()

{

Console.WriteLine("Задание 3ж: y → (z ≡ x)");

Console.Write("Введите X (true/false): ");

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

Console.Write("Введите Y (true/false): ");

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

Console.Write("Введите Z (true/false): ");

bool z = bool.Parse(Console.ReadLine());

bool result = !y || (z == x);

Console.WriteLine($"{y} → ({z} ≡ {x}) = {result}");

}

}

з) x ≡ (y → z).

using System;

public class Program

{

public static void Main()

{

Console.WriteLine("Задание 3з: x ≡ (y → z)");

Console.Write("Введите X (true/false): ");

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

Console.Write("Введите Y (true/false): ");

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

Console.Write("Введите Z (true/false): ");

bool z = bool.Parse(Console.ReadLine());

bool result = x == (!y || z);

Console.WriteLine($"{x} ≡ ({y} → {z}) = {result}");

}

}

* Записать логические выражения, которые имеют значение «Истина» только при выполнении указанных условий:

Console.Write("Введите число x: ");

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

Console.Write("Введите число y: ");

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

а) x < 2 и у < 3;

using System;

public class Program

{

public static void Main()

{

Console.WriteLine("Задание 4а: x < 2 и y < 3");

Console.Write("Введите число x: ");

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

Console.Write("Введите число y: ");

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

bool result = x < 2 && y < 3;

Console.WriteLine($"{x} < 2 и {y} < 3 = {result}");

}

}

б) неверно, что x < 2;

using System;

public class Program

{

public static void Main()

{

Console.WriteLine("Задание 4б: неверно, что x < 2");

Console.Write("Введите число x: ");

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

bool result = !(x < 2);

Console.WriteLine($"неверно, что {x} < 2 = {result}");

}

}

в) x < 1 или y < 2;

using System;

public class Program

{

public static void Main()

{

Console.WriteLine("Задание 4в: x < 1 или y < 2");

Console.Write("Введите число x: ");

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

Console.Write("Введите число y: ");

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

bool result = x < 1 || y < 2;

Console.WriteLine($"{x} < 1 или {y} < 2 = {result}");

}

}

г) неверно, что (x < 0 и x < 5);

using System;

public class Program

{

public static void Main()

{

Console.WriteLine("Задание 4г: неверно, что (x < 0 и x < 5)");

Console.Write("Введите число x: ");

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

bool result = !(x < 0 && x < 5);

Console.WriteLine($"неверно, что ({x} < 0 и {x} < 5) = {result}");

}

}

д) x < 0 и у > 5;

using System;

public class Program

{

public static void Main()

{

Console.WriteLine("Задание 4д: x < 0 и y > 5");

Console.Write("Введите число x: ");

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

Console.Write("Введите число y: ");

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

bool result = x < 0 && y > 5;

Console.WriteLine($"{x} < 0 и {y} > 5 = {result}");

}

}

е) 10 < x < 20;

using System;

public class Program

{

public static void Main()

{

Console.WriteLine("Задание 4е: 10 < x < 20");

Console.Write("Введите число x: ");

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

bool result = x > 10 && x < 20;

Console.WriteLine($"10 < {x} < 20 = {result}");

}

}

ж) x > 3 или x < 1;

using System;

public class Program

{

public static void Main()

{

Console.WriteLine("Задание 4ж: x > 3 или x < 1");

Console.Write("Введите число x: ");

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

bool result = x > 3 || x < 1;

Console.WriteLine($"{x} > 3 или {x} < 1 = {result}");

}

}

з) 0 < y < 4 и x < 5;

using System;

public class Program

{

public static void Main()

{

Console.WriteLine("Задание 4з: 0 < y < 4 и x < 5");

Console.Write("Введите число x: ");

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

Console.Write("Введите число y: ");

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

bool result = y > 0 && y < 4 && x < 5;

Console.WriteLine($"0 < {y} < 4 и {x} < 5 = {result}");

}

}

и) х > 3  и x < 10.

using System;

public class Program

{

public static void Main()

{

Console.WriteLine("Задание 4и: x > 3 и x < 10");

Console.Write("Введите число x: ");

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

bool result = x > 3 && x < 10;

Console.WriteLine($"{x} > 3 и {x} < 10 = {result}");

}

}

* Записать условие, которое является истинным, когда:

а) каждое из чисел *А* и *В* больше 100;

using System;

public class Program

{

public static void Main()

{

Console.WriteLine("Задание 5а: Каждое из чисел A и B больше 100");

Console.Write("Введите число A: ");

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

Console.Write("Введите число B: ");

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

bool result = A > 100 && B > 100;

Console.WriteLine($"Каждое из чисел {A} и {B} больше 100: {result}");

}

}

б) хотя бы одно из чисел *А* и *В* положительное;

using System;

public class Program

{

public static void Main()

{

Console.WriteLine("Задание 5б: Хотя бы одно из чисел A и B положительное");

Console.Write("Введите число A: ");

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

Console.Write("Введите число B: ");

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

bool result = A > 0 || B > 0;

Console.WriteLine($"Хотя бы одно из чисел {A} и {B} положительное: {result}");

}

}

в) каждое из чисел *А*, *В*, *С* кратно (делится без остатка) трем;

using System;

public class Program

{

public static void Main()

{

Console.WriteLine("Задание 5в: Каждое из чисел A, B, C кратно трём");

Console.Write("Введите число A: ");

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

Console.Write("Введите число B: ");

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

Console.Write("Введите число C: ");

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

bool result = A % 3 == 0 && B % 3 == 0 && C % 3 == 0;

Console.WriteLine($"Каждое из чисел {A}, {B}, {C} кратно трём: {result}");

}

}

г) только одно из чисел *А*, *В* и *С* меньше 50;

using System;

public class Program

{

public static void Main()

{

Console.WriteLine("Задание 5г: Только одно из чисел A, B и C меньше 50");

Console.Write("Введите число A: ");

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

Console.Write("Введите число B: ");

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

Console.Write("Введите число C: ");

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

int count = 0;

if (A < 50) count++;

if (B < 50) count++;

if (C < 50) count++;

bool result = count == 1;

Console.WriteLine($"Только одно из чисел {A}, {B}, {C} меньше 50: {result}");

}

}

д) хотя бы одно из чисел *А*, *В*, *С* отрицательно.

using System;

public class Program

{

public static void Main()

{

Console.WriteLine("Задание 5д: Хотя бы одно из чисел A, B, C отрицательно");

Console.Write("Введите число A: ");

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

Console.Write("Введите число B: ");

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

Console.Write("Введите число C: ");

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

bool result = A < 0 || B < 0 || C < 0;

Console.WriteLine($"Хотя бы одно из чисел {A}, {B}, {C} отрицательно: {result}");

}

}

е) каждое из чисел *X* и *Y* нечетное;

using System;

public class Program

{

public static void Main()

{

Console.WriteLine("Задание 5е: Каждое из чисел X и Y нечетное");

Console.Write("Введите число X (целое): ");

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

Console.Write("Введите число Y (целое): ");

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

bool result = X % 2 != 0 && Y % 2 != 0;

Console.WriteLine($"Каждое из чисел {X} и {Y} нечетное: {result}");

}

}

ж) только одно из чисел *X* и *Y* меньше 20;

using System;

public class Program

{

public static void Main()

{

Console.WriteLine("Задание 5ж: Только одно из чисел X и Y меньше 20");

Console.Write("Введите число X: ");

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

Console.Write("Введите число Y: ");

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

bool result = (X < 20 && Y >= 20) || (X >= 20 && Y < 20);

Console.WriteLine($"Только одно из чисел {X} и {Y} меньше 20: {result}");

}

}

з) хотя бы одно из чисел *X* и *Y* равно нулю;

using System;

public class Program

{

public static void Main()

{

Console.WriteLine("Задание 5з: Хотя бы одно из чисел X и Y равно нулю");

Console.Write("Введите число X: ");

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

Console.Write("Введите число Y: ");

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

bool result = X == 0 || Y == 0;

Console.WriteLine($"Хотя бы одно из чисел {X} и {Y} равно нулю: {result}");

}

}

и) каждое из чисел *X*, *Y*, *Z* отрицательное;

using System;

public class Program

{

public static void Main()

{

Console.WriteLine("Задание 5и: Каждое из чисел X, Y, Z отрицательное");

Console.Write("Введите число X: ");

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

Console.Write("Введите число Y: ");

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

Console.Write("Введите число Z: ");

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

bool result = X < 0 && Y < 0 && Z < 0;

Console.WriteLine($"Каждое из чисел {X}, {Y}, {Z} отрицательное: {result}");

}

}

к) только одно из чисел *X*, *Y* и *Z* кратно пяти;

using System;

public class Program

{

public static void Main()

{

Console.WriteLine("Задание 5к: Только одно из чисел X, Y и Z кратно пяти");

Console.Write("Введите число X: ");

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

Console.Write("Введите число Y: ");

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

Console.Write("Введите число Z: ");

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

int count = 0;

if (X % 5 == 0) count++;

if (Y % 5 == 0) count++;

if (Z % 5 == 0) count++;

bool result = count == 1;

Console.WriteLine($"Только одно из чисел {X}, {Y}, {Z} кратно пяти: {result}");

}

}

л) хотя бы одно из чисел *X*, *Y*, *Z*  больше 100.

using System;

public class Program

{

public static void Main()

{

Console.WriteLine("Задание 5л: Хотя бы одно из чисел X, Y, Z больше 100");

Console.Write("Введите число X: ");

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

Console.Write("Введите число Y: ");

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

Console.Write("Введите число Z: ");

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

bool result = X > 100 || Y > 100 || Z > 100;

Console.WriteLine($"Хотя бы одно из чисел {X}, {Y}, {Z} больше 100: {result}");

}

}

* Записать условие, которое является истинным, когда:

а) целое А кратно двум или трем;

using System;

public class Program

{

public static void Main()

{

Console.WriteLine("Задание 6а: Кратно ли число A двум или трём?");

Console.Write("Введите целое число A: ");

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

bool result = (A % 2 == 0) || (A % 3 == 0);

Console.WriteLine($"Число {A} кратно 2 или 3: {result}");

}

}

б) целое А не кратно трем и оканчивается нулем.

using System;

public class Program

{

public static void Main()

{

Console.WriteLine("Задание 6б: Не кратно ли A трём и оканчивается на 0?");

Console.Write("Введите целое число A: ");

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

bool result = (A % 3 != 0) && (A % 10 == 0);

Console.WriteLine($"Число {A} не кратно 3 и оканчивается на 0: {result}");

}

}

в) целое N кратно пяти или семи;

using System;

public class Program

{

public static void Main()

{

Console.WriteLine("Задание 6в: Кратно ли число N пяти или семи?");

Console.Write("Введите целое число N: ");

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

bool result = (N % 5 == 0) || (N % 7 == 0);

Console.WriteLine($"Число {N} кратно 5 или 7: {result}");

}

}

г) целое *N* кратно четырем и не оканчивается нулем.

using System;

public class Program

{

public static void Main()

{

Console.WriteLine("Задание 6г: Кратно ли N четырём и не оканчивается на 0?");

Console.Write("Введите целое число N: ");

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

bool result = (N % 4 == 0) && (N % 10 != 0);

Console.WriteLine($"Число {N} кратно 4 и не оканчивается на 0: {result}");

}

}

* Рассчитать значение у при заданном значении х:  
  y=sin(x) при x>0 или y=cos(x) в противном случае.

using System;

public class Program

{

public static void Main()

{

Console.WriteLine("Задание 7: Вычисление значения функции y = sin(x) при x > 0, иначе y = cos(x)");

Console.Write("Введите значение x (в радианах): ");

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

double y;

if (x > 0)

{

y = Math.Sin(x);

Console.WriteLine($"При x = {x} > 0, y = sin({x}) = {y:F4}");

}

else

{

y = Math.Cos(x);

Console.WriteLine($"При x = {x} <= 0, y = cos({x}) = {y:F4}");

}

}

}

* Даны два различных вещественных числа. Определить:

а) какое из них больше;

using System;

public class Program

{

public static void Main()

{

Console.WriteLine("Задание 8а: Какое из двух чисел больше?");

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

double num1 = double.Parse(Console.ReadLine());

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

double num2 = double.Parse(Console.ReadLine());

if (num1 > num2)

Console.WriteLine($"Число {num1} больше {num2}");

else

Console.WriteLine($"Число {num2} больше {num1}");

}

}

б) какое из них меньше.

using System;

public class Program

{

public static void Main()

{

Console.WriteLine("Задание 8б: Какое из двух чисел меньше?");

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

double num1 = double.Parse(Console.ReadLine());

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

double num2 = double.Parse(Console.ReadLine());

if (num1 < num2)

Console.WriteLine($"Число {num1} меньше {num2}");

else

Console.WriteLine($"Число {num2} меньше {num1}");

}

}

* Определить максимальное и минимальное значения из двух различных вещественных чисел.

using System;

public class Program

{

public static void Main()

{

Console.WriteLine("Определение максимального и минимального из двух чисел");

// Ввод данных с проверкой на различия

double num1, num2;

do {

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

num1 = double.Parse(Console.ReadLine());

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

num2 = double.Parse(Console.ReadLine());

if(num1 == num2) {

Console.WriteLine("Числа должны быть разными! Повторите ввод.");

}

} while(num1 == num2);

// Определение максимального и минимального

double max = Math.Max(num1, num2);

double min = Math.Min(num1, num2);

// Альтернативный вариант без Math:

// double max = num1 > num2 ? num1 : num2;

// double min = num1 < num2 ? num1 : num2;

// Вывод результата

Console.WriteLine($"\nРезультат:");

Console.WriteLine($"Максимальное число: {max}");

Console.WriteLine($"Минимальное число: {min}");

}

}

* Даны три различных вещественных числа. Не используя полный условный оператор, определить:

а) наибольшее из них;

using System;

public class Program

{

public static void Main()

{

Console.WriteLine("Нахождение наибольшего из трех чисел");

// Ввод и проверка чисел

double a, b, c;

do {

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

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

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

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

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

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

if(a == b || a == c || b == c)

Console.WriteLine("Числа должны быть разными!");

} while(a == b || a == c || b == c);

// Находим максимум последовательными сравнениями

double max = a;

if(b > max) max = b;

if(c > max) max = c;

Console.WriteLine($"Наибольшее число: {max}");

}

}

б) наименьшее из них

using System;

public class Program

{

public static void Main()

{

Console.WriteLine("Нахождение наименьшего из трех чисел");

// Ввод и проверка чисел

double a, b, c;

do {

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

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

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

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

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

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

if(a == b || a == c || b == c)

Console.WriteLine("Числа должны быть разными!");

} while(a == b || a == c || b == c);

// Находим минимум последовательными сравнениями

double min = a;

if(b < min) min = b;

if(c < min) min = c;

Console.WriteLine($"Наименьшее число: {min}");

}

}

* Даны четыре различных вещественных числа. Не используя полный условный оператор, определить:

а) наибольшее из них;

using System;

public class Program

{

public static void Main()

{

Console.WriteLine("Нахождение наибольшего из четырех чисел");

// Ввод и проверка чисел

double a, b, c, d;

do {

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

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

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

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

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

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

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

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

if(a == b || a == c || a == d || b == c || b == d || c == d)

Console.WriteLine("Все числа должны быть разными!");

} while(a == b || a == c || a == d || b == c || b == d || c == d);

// Находим максимум последовательными сравнениями

double max = a;

if(b > max) max = b;

if(c > max) max = c;

if(d > max) max = d;

Console.WriteLine($"Наибольшее число: {max:F4}");

}

}

б) наименьшее из них.

using System;

public class Program

{

public static void Main()

{

Console.WriteLine("Нахождение наименьшего из четырех чисел");

// Ввод и проверка чисел

double a, b, c, d;

do {

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

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

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

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

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

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

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

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

if(a == b || a == c || a == d || b == c || b == d || c == d)

Console.WriteLine("Все числа должны быть разными!");

} while(a == b || a == c || a == d || b == c || b == d || c == d);

// Находим минимум последовательными сравнениями

double min = a;

if(b < min) min = b;

if(c < min) min = c;

if(d < min) min = d;

Console.WriteLine($"Наименьшее число: {min:F4}");

}

}

Известны два расстояния: одно в километрах, другое — в футах (1 фут 0,305 м). Какое из расстояний меньше?

using System;

public class Program

{

public static void Main()

{

Console.WriteLine("Сравнение расстояний: километры vs футы");

// Ввод данных

Console.Write("Введите расстояние в километрах: ");

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

Console.Write("Введите расстояние в футах: ");

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

// Конвертация футов в километры (1 фут = 0.0003048 км)

double feetToKm = feet \* 0.0003048;

// Сравнение с выводом результата

Console.WriteLine("\nРезультат сравнения:");

if(km < feetToKm)

{

Console.WriteLine($"{km} км < {feet} футов");

Console.WriteLine($"({feet} футов = {feetToKm:F6} км)");

}

else if(km > feetToKm)

{

Console.WriteLine($"{km} км > {feet} футов");

Console.WriteLine($"({feet} футов = {feetToKm:F6} км)");

}

else

{

Console.WriteLine("Расстояния равны!");

Console.WriteLine($"({feet} футов = {feetToKm:F6} км)");

}

}

}

* Известны две скорости: одна в километрах в час, другая — в метрах в секунду. Какая из скоростей больше?

using System;

public class Program

{

public static void Main()

{

Console.WriteLine("Сравнение скоростей: км/ч и м/с");

// Ввод данных с проверкой

double kmh, ms;

while(true)

{

try

{

Console.Write("Введите скорость в км/ч: ");

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

Console.Write("Введите скорость в м/с: ");

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

break;

}

catch

{

Console.WriteLine("Ошибка ввода! Введите числовые значения.");

}

}

// Конвертация км/ч в м/с (1 км/ч = 1000/3600 м/с ≈ 0.277778 м/с)

double kmhToMs = kmh \* 1000 / 3600;

// Сравнение с выводом результата

Console.WriteLine("\nРезультат сравнения:");

Console.WriteLine($"{kmh} км/ч = {kmhToMs:F6} м/с");

Console.WriteLine($"{ms} м/с = {ms \* 3.6:F6} км/ч"); // Обратная конвертация

if(Math.Abs(kmhToMs - ms) < 0.000001) // Учет погрешности при сравнении

{

Console.WriteLine("Скорости равны!");

}

else if(kmhToMs > ms)

{

Console.WriteLine($"{kmh} км/ч > {ms} м/с");

}

else

{

Console.WriteLine($"{kmh} км/ч < {ms} м/с");

}

}

}

* Дано натуральное число.

а) Верно ли, что оно заканчивается четной цифрой?

using System;

public class Program

{

public static void Main()

{

Console.WriteLine("Проверка четности последней цифры числа");

int number;

while(true)

{

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

if(int.TryParse(Console.ReadLine(), out number) && number > 0)

break;

Console.WriteLine("Ошибка! Введите целое число больше 0.");

}

// Получаем последнюю цифру

int lastDigit = number % 10;

bool isEven = lastDigit % 2 == 0;

Console.WriteLine($"\nЧисло: {number}");

Console.WriteLine($"Последняя цифра: {lastDigit}");

Console.WriteLine($"Заканчивается на четную цифру: {isEven}");

}

}

б) Верно ли, что оно заканчивается нечетной цифрой?

using System;

public class Program

{

public static void Main()

{

Console.WriteLine("Проверка нечетности последней цифры числа");

int number;

while(true)

{

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

if(int.TryParse(Console.ReadLine(), out number) && number > 0)

break;

Console.WriteLine("Ошибка! Введите целое число больше 0.");

}

// Получаем последнюю цифру

int lastDigit = number % 10;

bool isOdd = lastDigit % 2 != 0;

Console.WriteLine($"\nЧисло: {number}");

Console.WriteLine($"Последняя цифра: {lastDigit}");

Console.WriteLine($"Заканчивается на нечетную цифру: {isOdd}");

}

}

* Даны радиус круга и сторона квадрата. У какой фигуры площадь больше?

Площадь круг S = π\*R2; Площадь квадрата S = a2;

using System;

public class Program

{

public static void Main()

{

Console.WriteLine("Сравнение площади круга и квадрата");

// Ввод данных с проверкой

double radius, side;

while(true)

{

try

{

Console.Write("Введите радиус круга (R > 0): ");

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

Console.Write("Введите сторону квадрата (a > 0): ");

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

if(radius > 0 && side > 0) break;

Console.WriteLine("Значения должны быть положительными!");

}

catch

{

Console.WriteLine("Ошибка ввода! Введите числовые значения.");

}

}

// Вычисление площадей

double circleArea = Math.PI \* Math.Pow(radius, 2);

double squareArea = Math.Pow(side, 2);

// Сравнение и вывод результата

Console.WriteLine("\nРезультаты расчетов:");

Console.WriteLine($"Площадь круга: {circleArea:F4}");

Console.WriteLine($"Площадь квадрата: {squareArea:F4}");

if(Math.Abs(circleArea - squareArea) < 0.0001)

{

Console.WriteLine("Площади равны!");

}

else if(circleArea > squareArea)

{

Console.WriteLine("Площадь круга больше площади квадрата");

}

else

{

Console.WriteLine("Площадь квадрата больше площади круга");

}

}

}

Определить, является ли число, а делителем числа b? А наоборот? (Получить два ответа.)

using System;

public class Program

{

public static void Main()

{

Console.WriteLine("Проверка взаимной делимости двух чисел");

// Ввод данных с проверкой

int a, b;

while(true)

{

try

{

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

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

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

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

break;

}

catch

{

Console.WriteLine("Ошибка! Введите целые числа.");

}

}

// Проверка делимости

bool aDividesB = (b % a == 0) && (a != 0);

bool bDividesA = (a % b == 0) && (b != 0);

// Вывод результатов

Console.WriteLine("\nРезультаты проверки:");

Console.WriteLine($"{a} является делителем {b}: {aDividesB}");

Console.WriteLine($"{b} является делителем {a}: {bDividesA}");

// Дополнительная информация

if(a == 0 && b == 0)

{

Console.WriteLine("\nПримечание: 0 не может быть делителем!");

}

else if(a == 0)

{

Console.WriteLine("\nПримечание: 0 не может быть делителем (первый случай)");

}

else if(b == 0)

{

Console.WriteLine("\nПримечание: 0 не может быть делителем (второй случай)");

}

}

}

* Даны два прямоугольника, стороны которых параллельны или перпендикулярны осям координат. Известны координаты левого нижнего угла каждого из них и длины их сторон. Один из прямоугольников назовем первым, другой — вторым.

а) Определить, принадлежат ли все точки первого прямоугольника второму.

using System;

public class Program

{

public struct Rectangle

{

public double X, Y; // Левый нижний угол

public double Width, Height;

public double Right => X + Width;

public double Top => Y + Height;

}

public static void Main()

{

Console.WriteLine("17а: Принадлежность первого прямоугольника второму");

// Ввод параметров прямоугольников

Rectangle rect1 = InputRectangle("первого");

Rectangle rect2 = InputRectangle("второго");

// Проверка условия

bool isInside = rect1.X >= rect2.X &&

rect1.Y >= rect2.Y &&

rect1.Right <= rect2.Right &&

rect1.Top <= rect2.Top;

Console.WriteLine($"\nРезультат: {isInside}");

}

static Rectangle InputRectangle(string name)

{

Console.WriteLine($"\nВвод {name} прямоугольника:");

return new Rectangle {

X = ReadDouble("X левого нижнего угла"),

Y = ReadDouble("Y левого нижнего угла"),

Width = ReadPositive("ширина"),

Height = ReadPositive("высота")

};

}

static double ReadDouble(string prompt)

{

while(true)

{

Console.Write($"{prompt}: ");

if(double.TryParse(Console.ReadLine(), out double value))

return value;

Console.WriteLine("Ошибка! Введите число.");

}

}

static double ReadPositive(string prompt)

{

while(true)

{

double value = ReadDouble(prompt);

if(value > 0) return value;

Console.WriteLine("Значение должно быть положительным!");

}

}

}

б) Определить, принадлежат ли все точки одного из прямоугольников другому.

using System;

public class Program

{

public struct Rectangle { /\* как в 17а \*/ }

public static void Main()

{

Console.WriteLine("17б: Проверка вложенности прямоугольников");

Rectangle rect1 = InputRectangle("первого");

Rectangle rect2 = InputRectangle("второго");

bool isFirstInSecond = IsContained(rect1, rect2);

bool isSecondInFirst = IsContained(rect2, rect1);

Console.WriteLine($"\nРезультат: {isFirstInSecond || isSecondInFirst}");

}

static bool IsContained(Rectangle inner, Rectangle outer)

{

return inner.X >= outer.X &&

inner.Y >= outer.Y &&

inner.Right <= outer.Right &&

inner.Top <= outer.Top;

}

// Методы InputRectangle, ReadDouble, ReadPositive аналогичны 17а

}

в) Определить, пересекаются ли эти прямоугольники.

using System;

public class Program

{

public struct Rectangle { /\* как в 17а \*/ }

public static void Main()

{

Console.WriteLine("17в: Проверка пересечения прямоугольников");

Rectangle rect1 = InputRectangle("первого");

Rectangle rect2 = InputRectangle("второго");

bool intersects = rect1.X < rect2.Right &&

rect1.Right > rect2.X &&

rect1.Y < rect2.Top &&

rect1.Top > rect2.Y;

Console.WriteLine($"\nРезультат: {intersects}");

}

// Методы InputRectangle, ReadDouble, ReadPositive аналогичны 17а

}

* Год является високосным, если его номер кратен 4, однако из кратных 100 високосными являются лишь кратные 400, например, 1700, 1800 и 1900 — не-  
  високосные года, 2000 — високосный. Дано натуральное число n. Определить, является ли високосным год с таким номером.

using System;

public class Program

{

public static void Main()

{

Console.WriteLine("Определение високосного года");

int year;

while(true)

{

Console.Write("Введите год (натуральное число): ");

if(int.TryParse(Console.ReadLine(), out year) && year > 0)

break;

Console.WriteLine("Ошибка! Введите целое число больше 0.");

}

bool isLeap = (year % 400 == 0) || (year % 100 != 0 && year % 4 == 0);

Console.WriteLine($"\nГод {year} {(isLeap ? "високосный" : "не високосный")}");

Console.WriteLine($"Правила:\n- Кратен 400: {year % 400 == 0}" +

$"\n- Кратен 100, но не 400: {year % 100 == 0 && year % 400 != 0}" +

$"\n- Кратен 4, но не 100: {year % 4 == 0 && year % 100 != 0}");

}

}

Имеются стол прямоугольной формы с размерами a x b (a и b — целые числа, a > b) и кости домино с размерами c x d x e (c, d и e — целые числа, c > d > e). Найти вариант размещения на столе наибольшего количества костей. Все размещаемые кости должны лежать на одной и той же грани в один ярус без свешивания со стола. Все ребра костей домино должны быть параллельны или перпендикулярны каждой стороне стола.

using System;

public class Program

{

public static void Main()

{

Console.WriteLine("Оптимальное размещение костей домино на столе");

// Ввод параметров стола

Console.WriteLine("\nВведите размеры стола (a > b):");

int a = ReadPositiveInt("Длина стола (a)");

int b = ReadPositiveInt("Ширина стола (b)");

if (a <= b)

{

Console.WriteLine("Ошибка: a должно быть больше b!");

return;

}

// Ввод параметров кости домино

Console.WriteLine("\nВведите размеры кости домино (c > d > e):");

int c = ReadPositiveInt("Наибольший размер (c)");

int d = ReadPositiveInt("Средний размер (d)");

int e = ReadPositiveInt("Наименьший размер (e)");

if (c <= d || d <= e)

{

Console.WriteLine("Ошибка: должны соблюдаться условия c > d > e!");

return;

}

// Рассчет максимального количества для всех возможных ориентаций

int maxCount = 0;

string bestOrientation = "";

// Вариант 1: Ориентация c x d (плоская)

int count1 = (a / c) \* (b / d);

if (count1 > maxCount)

{

maxCount = count1;

bestOrientation = $"c x d ({c}×{d}) - плоская";

}

// Вариант 2: Ориентация d x c (плоская, повернутая)

int count2 = (a / d) \* (b / c);

if (count2 > maxCount)

{

maxCount = count2;

bestOrientation = $"d x c ({d}×{c}) - плоская повернутая";

}

// Вариант 3: Ориентация c x e (реберная)

int count3 = (a / c) \* (b / e);

if (count3 > maxCount)

{

maxCount = count3;

bestOrientation = $"c x e ({c}×{e}) - реберная";

}

// Вариант 4: Ориентация e x c (реберная, повернутая)

int count4 = (a / e) \* (b / c);

if (count4 > maxCount)

{

maxCount = count4;

bestOrientation = $"e x c ({e}×{c}) - реберная повернутая";

}

// Вариант 5: Ориентация d x e (торцевая)

int count5 = (a / d) \* (b / e);

if (count5 > maxCount)

{

maxCount = count5;

bestOrientation = $"d x e ({d}×{e}) - торцевая";

}

// Вариант 6: Ориентация e x d (торцевая, повернутая)

int count6 = (a / e) \* (b / d);

if (count6 > maxCount)

{

maxCount = count6;

bestOrientation = $"e x d ({e}×{d}) - торцевая повернутая";

}

// Вывод результатов

Console.WriteLine($"\nМаксимальное количество костей: {maxCount}");

Console.WriteLine($"Оптимальная ориентация: {bestOrientation}");

Console.WriteLine($"Размещение: {a / GetFirstDim(bestOrientation)} × {b / GetSecondDim(bestOrientation)}");

}

static int ReadPositiveInt(string prompt)

{

while (true)

{

Console.Write($"{prompt}: ");

if (int.TryParse(Console.ReadLine(), out int value) && value > 0)

return value;

Console.WriteLine("Ошибка! Введите целое число больше 0.");

}

}

static int GetFirstDim(string orientation)

{

return int.Parse(orientation.Split('×')[0].Split(' ').Last());

}

static int GetSecondDim(string orientation)

{

return int.Parse(orientation.Split('×')[1].Split(')')[0]);

}

}

* Дано целое число k (1 <= k <= 365). Определить, каким будет k-й день года: выходным (суббота и воскресенье) или рабочим, если 1 января — понедельник.

Даны два различных вещественных числа. Определить наибольшее и наименьшее из них, не используя полный условный оператор, а применив:

а) два неполных условных оператора;

using System;

public class Program

{

public static void Main()

{

Console.WriteLine("Определение выходного дня в году (1 января - понедельник)");

int k;

do

{

Console.Write("Введите день года (1-365): ");

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

} while (k < 1 || k > 365);

// Определяем день недели (0-пн, 1-вт, ..., 5-сб, 6-вс)

int dayOfWeek = (k - 1) % 7;

bool isWeekend = dayOfWeek >= 5; // 5 или 6 - выходные

string dayName = GetDayName(dayOfWeek);

Console.WriteLine($"\n{k}-й день года - {dayName}");

Console.WriteLine($"Это {(isWeekend ? "выходной" : "рабочий")} день");

}

static string GetDayName(int dayIndex)

{

return dayIndex switch

{

0 => "понедельник",

1 => "вторник",

2 => "среда",

3 => "четверг",

4 => "пятница",

5 => "суббота",

6 => "воскресенье",

\_ => throw new ArgumentException()

};

}

}

б) один неполный условный оператор.

using System;

public class Program

{

public static void Main()

{

Console.WriteLine("Нахождение max и min (один тернарный оператор)");

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

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

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

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

double max = a > b ? a : b;

double min = a < b ? a : b;

Console.WriteLine($"\nМаксимальное: {max}");

Console.WriteLine($"Минимальное: {min}");

}

}

* Работа светофора для пешеходов запрограммирована следующим образом: в начале каждого часа в течение трех минут горит зеленый сигнал, затем в течение двух минут — красный, в течение трех минут — опять зеленый и т. д. Дано вещественное число t, означающее время в минутах, прошедшее с начала очередного часа. Определить, сигнал какого цвета горит для пешеходов в этот момент.

using System;

public class Program

{

public static void Main()

{

Console.WriteLine("Определение сигнала светофора для пешеходов");

double t;

while (true)

{

Console.Write("Введите время в минутах (0 ≤ t < 60): ");

if (double.TryParse(Console.ReadLine(), out t) && t >= 0 && t < 60)

break;

Console.WriteLine("Ошибка! Введите число от 0 до 60 (не включая 60).");

}

// Определяем текущий цикл (5-минутные циклы: 3 зеленый + 2 красный)

double cyclePosition = t % 5; // Позиция в текущем 5-минутном цикле

string color = cyclePosition < 3 ? "зеленый" : "красный";

Console.WriteLine($"\nВ момент времени {t} минут горит: {color}");

Console.WriteLine($"Текущая позиция в цикле: {cyclePosition:F2} минут");

// Дополнительная информация о времени до смены

if (color == "зеленый")

{

double timeLeft = 3 - cyclePosition;

Console.WriteLine($"До смены на красный: {timeLeft:F2} минут");

}

else

{

double timeLeft = 5 - cyclePosition;

Console.WriteLine($"До смены на зеленый: {timeLeft:F2} минут");

}

}

}

Имеется стол прямоугольной формы с размерами a x b (a и b — целые числа, a > b). В каком случае на столе можно разместить большее количество картонных прямоугольников с размерами c x d (c и d — целые числа, c > d): при размещении их длинной стороной вдоль длинной стороны стола или вдоль короткой. Прямоугольники не должны лежать один на другом и не должны свисать со стола.

using System;

public class Program

{

public static void Main()

{

Console.WriteLine("Оптимальное размещение прямоугольников на столе");

// Ввод параметров стола

Console.WriteLine("\nВведите размеры стола (a > b):");

int a = ReadPositiveInt("Длина стола (a)");

int b = ReadPositiveInt("Ширина стола (b)");

if (a <= b)

{

Console.WriteLine("Ошибка: a должно быть больше b!");

return;

}

// Ввод параметров прямоугольников

Console.WriteLine("\nВведите размеры прямоугольника (c > d):");

int c = ReadPositiveInt("Длина прямоугольника (c)");

int d = ReadPositiveInt("Ширина прямоугольника (d)");

if (c <= d)

{

Console.WriteLine("Ошибка: c должно быть больше d!");

return;

}

// Расчет количества для двух вариантов размещения

int count1 = (a / c) \* (b / d); // Длинной стороной вдоль длинной стороны стола

int count2 = (a / d) \* (b / c); // Длинной стороной вдоль короткой стороны стола

// Определение оптимального варианта

string result;

if (count1 > count2)

{

result = $"лучше размещать длинной стороной вдоль длинной стороны стола ({count1} > {count2})";

}

else if (count2 > count1)

{

result = $"лучше размещать длинной стороной вдоль короткой стороны стола ({count2} > {count1})";

}

else

{

result = $"количество одинаково в обоих вариантах ({count1})";

}

// Вывод результатов

Console.WriteLine($"\nРезультаты:");

Console.WriteLine($"1. Длинной вдоль длинной: {count1} прямоугольников");

Console.WriteLine($"2. Длинной вдоль короткой: {count2} прямоугольников");

Console.WriteLine($"\nОптимально: {result}");

}

static int ReadPositiveInt(string prompt)

{

while (true)

{

Console.Write($"{prompt}: ");

if (int.TryParse(Console.ReadLine(), out int value) && value > 0)

return value;

Console.WriteLine("Ошибка! Введите целое число больше 0.");

}

}

}

Игральным картам условно присвоены следующие порядковые номера в зависимости от их достоинства: «валету» — 11, «даме» — 12, «королю» — 13, «тузу» — 14. Порядковые номера остальных карт соответствуют их названиям («шестерка», «девятка» и т. п.). По заданному номеру карты k (6 <=k <= 14) определить достоинство соответствующей карты.

using System;

public class Program

{

public static void Main()

{

Console.WriteLine("Определение достоинства карты");

int k;

do

{

Console.Write("Введите номер карты (6-14): ");

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

} while (k < 6 || k > 14);

string cardName = k switch

{

11 => "валет",

12 => "дама",

13 => "король",

14 => "туз",

\_ => k.ToString() // Для номеров 6-10

};

Console.WriteLine($"\nКарта с номером {k} - это {cardName}");

}

}

* Мастям игральных карт условно присвоены следующие порядковые номера: масти «пики» — 1, масти «трефы» — 2, масти «бубны» — 3, масти «червы» — 4, а достоинству карт: «валету» — 11, «даме» — 12, «королю» — 13, «тузу» — 14 (порядковые номера карт остальных достоинств соответствуют их названиям: «шестерка», «девятка» и т. п.). По заданным номеру масти m (1 <= m <= 4) и номеру достоинства карты k (6 <= k <= 14) определить полное название (масть и достоинство) соответствующей карты в виде «Дама пик», Шестерка бубен» и т. п.

using System;

public class Program

{

public static void Main()

{

Console.WriteLine("Определение полного названия карты");

// Ввод номера масти с проверкой

int m;

do

{

Console.Write("Введите номер масти (1-4):\n" +

"1 - пики\n2 - трефы\n3 - бубны\n4 - червы\n> ");

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

} while (m < 1 || m > 4);

// Ввод номера достоинства с проверкой

int k;

do

{

Console.Write("\nВведите номер достоинства (6-14):\n" +

"6-10 - соответствующие числа\n11 - валет\n12 - дама\n" +

"13 - король\n14 - туз\n> ");

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

} while (k < 6 || k > 14);

// Определение масти

string suit = m switch

{

1 => "пик",

2 => "треф",

3 => "бубен",

4 => "червей",

\_ => throw new Exception("Недопустимая масть")

};

// Определение достоинства

string rank = k switch

{

11 => "валет",

12 => "дама",

13 => "король",

14 => "туз",

\_ => k.ToString() + "ка" // Для 6-10: "шестерка", "семерка" и т.д.

};

// Склонение для числительных (6-10)

if (k >= 6 && k <= 10)

{

rank = k switch

{

6 => "шестерка",

7 => "семерка",

8 => "восьмерка",

9 => "девятка",

10 => "десятка",

\_ => rank

};

}

// Формирование результата

string fullName = $"{rank.First().ToString().ToUpper()}{rank.Substring(1)} {suit}";

Console.WriteLine($"\nПолное название карты: {fullName}");

}

}

* Дано целое число k (1 <= k <= 365). Определить, каким днем недели (понедельником, вторником, …, субботой или воскресеньем) является k-й день не високосного года, в котором 1 января:

а) понедельник;

using System;

public class Program

{

public static void Main()

{

Console.WriteLine("Определение дня недели (1 января - понедельник)");

int k;

do

{

Console.Write("Введите день года (1-365): ");

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

} while (k < 1 || k > 365);

int dayOfWeek = (k - 1) % 7; // 0-пн, 1-вт, ..., 6-вс

string dayName = dayOfWeek switch

{

0 => "понедельник",

1 => "вторник",

2 => "среда",

3 => "четверг",

4 => "пятница",

5 => "суббота",

6 => "воскресенье",

\_ => "ошибка"

};

Console.WriteLine($"\n{k}-й день года - это {dayName}");

}

}

б) d-й день недели (если 1 января — понедельник, то d = 1,

если вторник — d = 2, …, если воскресенье — d = 7).

using System;

public class Program

{

public static void Main()

{

Console.WriteLine("Определение дня недели (1 января - заданный день)");

int k, d;

do

{

Console.Write("Введите день года (1-365): ");

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

Console.Write("Введите номер дня недели 1 января (1-7): ");

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

} while (k < 1 || k > 365 || d < 1 || d > 7);

int dayOfWeek = (k - 1 + d - 1) % 7; // 0-пн, 1-вт, ..., 6-вс

string dayName = dayOfWeek switch

{

0 => "понедельник",

1 => "вторник",

2 => "среда",

3 => "четверг",

4 => "пятница",

5 => "суббота",

6 => "воскресенье",

\_ => "ошибка"

};

Console.WriteLine($"\n{k}-й день года - это {dayName}");

}

}

* С начала 1990 года по некоторый день прошло n месяцев и 2 дня. Определить название месяца (январь, февраль и т. п.) этого дня.

using System;

public class Program

{

public static void Main()

{

Console.WriteLine("Определение месяца по количеству месяцев и дней с 1990 года");

int n;

do

{

Console.Write("Введите количество полных месяцев (n ≥ 0): ");

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

} while (n < 0);

// Начальная дата - 1 января 1990 года (месяц 0)

int totalMonths = n + 1; // +1 потому что прошло n месяцев и 2 дня (уже следующий месяц)

// Определяем год и месяц

int year = 1990 + totalMonths / 12;

int month = totalMonths % 12;

if (month == 0)

{

month = 12;

year--;

}

// Получаем название месяца

string monthName = new DateTime(year, month, 1).ToString("MMMM", new System.Globalization.CultureInfo("ru-RU"));

Console.WriteLine($"\nПрошло {n} месяцев и 2 дня с января 1990 года");

Console.WriteLine($"Это {monthName.ToLower()} {year} года");

}

}

* Дата некоторого дня характеризуется двумя натуральными числами: m (порядковый номер месяца) и n (число). По заданным n и m определить:

а) дату предыдущего дня (принять, что n и m не характеризуют 1 января);

using System;

public class Program

{

public static void Main()

{

Console.WriteLine("Определение предыдущего дня (не високосный год)");

// Ввод данных с проверкой

int m, n;

do

{

Console.Write("Введите номер месяца (1-12): ");

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

Console.Write("Введите число (1-31): ");

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

// Проверка, что это не 1 января

if (m == 1 && n == 1)

Console.WriteLine("Ошибка: это 1 января! Введите другую дату.");

} while (m < 1 || m > 12 || n < 1 || n > 31 || (m == 1 && n == 1));

// Количество дней в каждом месяце (не високосный год)

int[] daysInMonth = { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };

// Определение предыдущего дня

int prevDay = n - 1;

int prevMonth = m;

int prevYear = 2023; // пример года

if (prevDay < 1)

{

prevMonth--;

if (prevMonth < 1)

{

prevMonth = 12;

prevYear--;

}

prevDay = daysInMonth[prevMonth - 1];

}

// Получение названия месяца

string monthName = new DateTime(prevYear, prevMonth, 1).ToString("MMMM", new System.Globalization.CultureInfo("ru-RU"));

Console.WriteLine($"\nПредыдущий день: {prevDay} {monthName.ToLower()}");

}

}

б) дату следующего дня (принять, что n и m не характеризуют 31 декабря).

using System;

public class Program

{

public static void Main()

{

Console.WriteLine("Определение следующего дня (не високосный год)");

// Ввод данных с проверкой

int m, n;

do

{

Console.Write("Введите номер месяца (1-12): ");

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

Console.Write("Введите число (1-31): ");

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

// Проверка, что это не 31 декабря

if (m == 12 && n == 31)

Console.WriteLine("Ошибка: это 31 декабря! Введите другую дату.");

} while (m < 1 || m > 12 || n < 1 || n > 31 || (m == 12 && n == 31));

// Количество дней в каждом месяце (не високосный год)

int[] daysInMonth = { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };

// Определение следующего дня

int nextDay = n + 1;

int nextMonth = m;

int nextYear = 2023; // пример года

if (nextDay > daysInMonth[m - 1])

{

nextDay = 1;

nextMonth++;

if (nextMonth > 12)

{

nextMonth = 1;

nextYear++;

}

}

// Получение названия месяца

string monthName = new DateTime(nextYear, nextMonth, 1).ToString("MMMM", new System.Globalization.CultureInfo("ru-RU"));

Console.WriteLine($"\nСледующий день: {nextDay} {monthName.ToLower()}");

}

}