

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

Кафедра программных систем

**Лабораторная работа №3**

**Использование выражений**

Выполнил: Мороз

Илья Олегович

Группа № K3120

Проверила: Казанова П.П.

Санкт-Петербург

2019

**Цель работы:**

Целью работы является знакомство с условными операторами и циклами в языке C#, а также создание программ на основе полученных знаний.

**Ход работы:**

**Упражнение 1.**

1. Был создан новый проект Shapeifelse. Данная программа позволит определять, лежит ли точка внутри фигуры или нет.
2. Был реализован ввод координат точки с клавиатуры.
3. При помощи оператора if-else была реализована проверка местонахождения точки.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Shapeifelse

{

class Program

{

static void Main(string[] args)

{

Console.Write("x = ");

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

Console.Write("y = ");

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

if (x \* x + y \* y < 9 && y > 0)

{

Console.WriteLine("Inside");

}

else if (x \* x + y \* y > 9 || y < 0)

{

Console.WriteLine("Outside");

}

else

{

Console.WriteLine("On the border");

}

}

}

}

Листинг 1 – исходный код программы Shapeifelse.cs

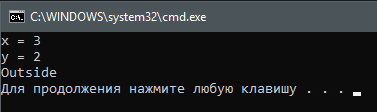


Рисунок 1 – результат работы программы

1. Был создан новый проект Calc\_switch. В данной программе будет использован оператор выбора switch.
2. Был реализован ввод чисел и арифметической операции с клавиатуры.
3. При помощи оператора выбора switch был получен результат вычислений, который после выводится в консоль.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Calc\_switch

{

class Program

{

static void Main(string[] args)

{

Console.Write("A = ");

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

Console.Write("OP = ");

char op = char.Parse(Console.ReadLine());

Console.Write("B = ");

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

bool ok = true;

double res = 0;

switch (op)

{

case '+': res = a + b; break;

case '-': res = a - b; break;

case '\*': res = a \* b; break;

case '/':

case ':': res = a / b; break;

default: ok = false; break;

}

if (ok)

{

Console.WriteLine("{0} {1} {2} = {3}", a, op, b, res);

}

else

{

Console.WriteLine("Операция не определена");

}

}

}

}

Листинг 2 – исходный код программы Calc\_switch.cs

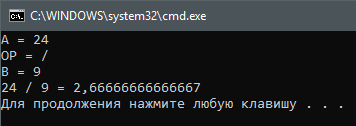


Рисунок 2 – результат работы программы

1. Был создан новый проект VisYear. Данная программа будет проверять, является ли год високосным.
2. Были реализованы ввод года с клавиатуры и его проверка на високосность.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace VisYear

{

class Program

{

static void Main(string[] args)

{

Console.Write("Enter year: ");

int year = Int32.Parse(Console.ReadLine());

if ((year % 4 == 0 && year % 100 != 0) || year % 400 == 0)

{

Console.WriteLine("Yes");

}

else

{

Console.WriteLine("No");

}

}

}

}

Листинг 3 – исходный код программы VisYear.cs

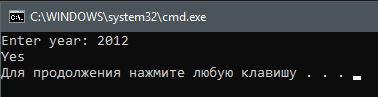


Рисунок 3 – результат работы программы

**Упражнение 2.**

1. Был создан новый проект Loop. В данной программе будут рассмотрены циклы в языке C#.
2. Был реализован ввод кол-ва чисел с клавиатуры.
3. Был реализован вывод нечётных чисел в промежутке от 1 до n на экран при помощи циклов while, do-while и for.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Loop

{

class Program

{

static void Main(string[] args)

{

Console.Write("n = ");

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

Console.Write("\nwhile: \t\t");

int i = 1;

while (i <= n)

{

Console.Write(" " + i);

i += 2;

}

Console.Write("\ndo while: \t");

i = 1;

do

{

Console.Write(" " + i);

i += 2;

} while (i <= n);

Console.Write("\nfor: \t\t");

for (i = 1; i <= n; i += 2)

{

Console.Write(" " + i);

}

}

}

}

Листинг 4 – исходный код программы Loop.cs

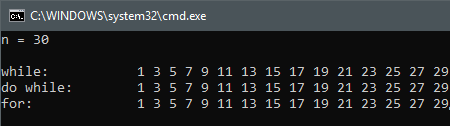


Рисунок 4 – результат работы программы

1. Был создан новый проект SinDoWhile. В данной программе производится расчёт и вывод значений y=sin(x), при x1<=x<=x2, с помощью цикла с постусловием.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace SinDoWhile

{

class Program

{

static void Main(string[] args)

{

Console.Write("x1 = ");

double x1 = double.Parse(Console.ReadLine());

Console.Write("x2 = ");

double x2 = double.Parse(Console.ReadLine());

double x = x1;

double y;

Console.Write("x\t\tsin(x)\n");

do

{

y = Math.Sin(x);

Console.Write("{0}\t\t{1}\n", x, y);

x += 0.01;

} while (x <= x2);

}

}

}

Листинг 5 – исходный код программы SinDoWhile.cs

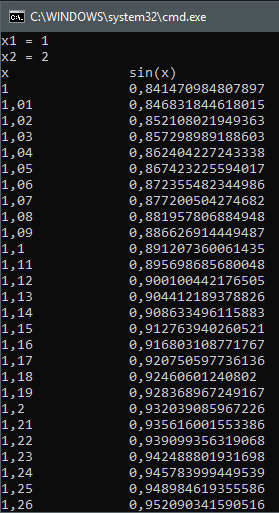


Рисунок 5 – результат работы программы

1. Был создан новый проект EvklidNod. В данной программе будет реализован алгоритм Евклида для нахождения НОД. А также будет проведено сравнение циклов с пред- и постусловием.
2. Был реализован ввод двух чисел с клавиатуры. На основе полученных данных с помощью алгоритма Евклида НОД двух чисел выводится в консоль.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace EvklidNod

{

class Program

{

static void Main(string[] args)

{

Console.Write("a = ");

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

Console.Write("b = ");

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

int temp = a;

while (temp != b)

{

a = temp;

if (a<b)

{

temp = a;

a = b;

b = temp;

}

temp = a - b;

a = b;

}

Console.WriteLine("nod = {0}", temp);

}

}

}

Листинг 6 – реализация алгоритма Евклида с помощью цикла с предусловием

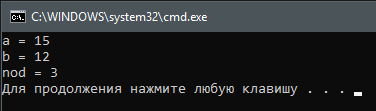


Рисунок 6 – результат работы программы

1. Алгоритм Евклида был реализован с помощью цикла с постусловием. Необходимо заметить, что в таком случае алгоритм находит неверный ответ для одинаковых чисел.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace EvklidNod

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("While:");

Console.Write("a = ");

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

Console.Write("b = ");

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

int temp = a;

while (temp != b)

{

a = temp;

if (a < b)

{

temp = a;

a = b;

b = temp;

}

temp = a - b;

a = b;

}

Console.WriteLine("nod = {0}", temp);

Console.WriteLine("Do-while:");

Console.Write("a = ");

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

Console.Write("b = ");

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

temp = a;

do

{

a = temp;

if (a < b)

{

temp = a;

a = b;

b = temp;

}

temp = a - b;

a = b;

} while (temp != b);

Console.WriteLine("nod = {0}", temp);

}

}

}

Листинг 7 – исходный код программы EvklidNod.cs

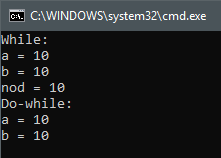


Рисунок 7 – в данном случае программа вошла в бесконечный цикл и не завершилась самостоятельно

1. Был создан новый проект SumKM. В данной программе будет реализовано нахождение суммы чисел от 1 до k и от m до 100.
2. Был реализован ввод чисел k и m с клавиатуры.
3. Был реализован алгоритм нахождения суммы с использованием оператора перехода continue.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace SumKM

{

class Program

{

static void Main(string[] args)

{

Console.Write("k = ");

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

Console.Write("m = ");

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

int sum = 0;

for (int i = 1; i <= 100; i++)

{

if (i > k && i < m) continue;

sum += i;

}

Console.WriteLine("sum = {0}", sum);

}

}

}

Листинг 8 – исходный код программы SumKM.cs

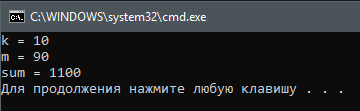


Рисунок 8 – результат работы программы

1. Был создан новый проект Shooting. В данной программе будет реализована стрельба по мишени с подсчётом баллов.
2. Был реализован ввод кол-ва выстрелов с клавиатуры. На основе этого был написан цикл, который запрашивает координаты n выстрелов и подсчитывает кол-во баллов, которые заработал стрелок.
3. Центр мишени определяется случайным образом. Генерация случайного числа обеспечивается объектом класса Random.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Shooting

{

class Program

{

static void Main(string[] args)

{

int n = 1;

int sum = 0;

Random rand = new Random();

int centerX = rand.Next(-1, 1);

int centerY = rand.Next(-1, 1);

Console.Write("Number of shots: ");

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

int[] x = new int[n], y = new int[n];

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

{

Console.Write("x = ");

x[i] = int.Parse(Console.ReadLine());

Console.Write("y = ");

y[i] = int.Parse(Console.ReadLine());

if (Math.Pow(x[i] - centerX, 2) + Math.Pow(y[i] - centerY, 2) <= 1)

{

Console.WriteLine("Wow! 10 points!");

sum += 10;

}

else if (Math.Pow(x[i] - centerX, 2) + Math.Pow(y[i] - centerY, 2) <= 4)

{

Console.WriteLine("Not bad. 5 point.");

sum += 5;

}

else if (Math.Pow(x[i] - centerX, 2) + Math.Pow(y[i] - centerY, 2) <= 9)

{

Console.WriteLine("1 point.");

sum += 1;

}

else

{

Console.WriteLine("Miss!");

}

}

Console.WriteLine("Total score: {0}", sum);

}

}

}

Листинг 9 – исходный код программы Shooting.cs

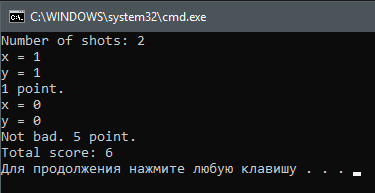


Рисунок 9 – пример работы программы. Стрельба в центр координат не гарантирует попадание в центр мишени.

**Вывод:**

В ходе выполнения данной лабораторной работы были получены новые знания в языке C#. Были рассмотрены условные операторы и циклы. Было проведено сравнение циклов с пред- и постусловием. В конце лабораторной работы все полученные знания были объеденены в программе, имитирующей стрельбу по мишени.

**Ответы на контрольные вопросы:**

1. Почему в алгоритме Евклида необходимо использовать while?

При использовании do-while проверка на равенство входных значений происходит только после первой итерации, что приводит к некорректному ответу.

1. Зачем в switch использовать break?

Если его не использовать, то программа будет переходить к выполнению следующего блока case.