**1 Принципы объектно-ориентированного программирования**

Задание 1. Написать программу, которая вычисляет значение функции у:

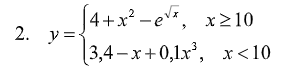


Рисунок 1.1 – Уравнение функции

Листинг программы:

using static System.Console;

namespace zad1

{

class Program

{

public static void Main()

{

Console.Write("Введите значение x = ");

double x = Convert.ToInt32(Console.ReadLine());

double y;

double e = 2.718;

if (x >= 10)

{

y = 4 + Math.Pow(x, 2) - Math.Pow(e, Math.Sqrt(x));

Console.WriteLine("Значение Y: "+ y);

}

else if (x < 10)

{

y = 3.4 - x + 0.1 \* Math.Pow(x, 3);

Console.WriteLine("Значение Y: "+ y);

}

}

}

}

Таблица 1.1 – Выходные и входные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 2 | 2,2 |

Анализ результатов:



Рисунок 1.2 – Результат работы программы

Источник: собственная разработка

Задание 2. Написать программу, которая определяет является ли заданное целое число четным.

Листинг программы:

using System;

using static System.Console;

namespace zad2

{

class Program

{

public static void Main()

{

Console.Write("Введите число: ");

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

if (a % 2 == 0)

Console.WriteLine("число чётное");

else

Console.WriteLine("число нечётное");

}

}

}

Таблица 1.2 – Выходные и входные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 3 | Нечётное |

Анализ результатов:



Рисунок 1.3 – Результат работы программы

Источник: собственная разработка

Задание 3. Составить программу (при решении данных задач использовать оператор switch или вложенные операторы if). Дан год. Вывести на экран название животного, символизирующего заданный год по восточному календарю.

Листинг программы:

using System;

using static System.Console;

namespace secondPart\_zad1

{

class Program

{

static void Main(string[] args)

{

Console.Write("Введите год: ");

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

year %= 12;

year+=1;

switch (year)

{

case 1: Console.WriteLine("Обезьяна"); break;

case 2: Console.WriteLine("Петух"); break;

case 3: Console.WriteLine("Собака"); break;

case 4: Console.WriteLine("Кабан"); break;

case 5: Console.WriteLine("Крыса"); break;

case 6: Console.WriteLine("Бык"); break;

case 7: Console.WriteLine("Тигр"); break;

case 8: Console.WriteLine("Кролик"); break;

case 9: Console.WriteLine("Дракон"); break;

case 10: Console.WriteLine("Змея"); break;

case 11: Console.WriteLine("Лошадь"); break;

case 12: Console.WriteLine("Овца"); break;

default: Console.WriteLine("ВЫ ОШИБЛИСЬ"); break;

}

Console.ReadKey();

}

}

}

Таблица 1.3 – Выходные и входные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 2005 | Петух |

Анализ результатов:



Рисунок 1.4 – Результат работы программы

Источник: собственная разработка

Задание 4. Вывести на экран (задачу решите тремя способами - используя операторы цикла while, do while и for): целые числа 10, 12, 14, …, 80 в обратном порядке в столбик;

Листинг программы:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Runtime.CompilerServices;

using System.Text;

using System.Threading.Tasks;

namespace secondPart\_zad1

{

class Program

{

static void Main(string[] args)

{

for (int i = 80; i >= 10; i -= 2)

{

Console.WriteLine(i);

}

int k = 80;

while (k >= 10)

{

Console.WriteLine(k);

k -= 2;

}

int n = 80;

do

{

Console.WriteLine(n);

n -= 2;

}

while (n >= 10);

}

}

}

Таблица 1.4 – Выходные и входные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
|  | 80-10 |

Анализ результатов:

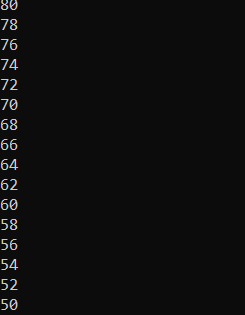


Рисунок 1.5 – Результат работы программы

Источник: собственная разработка

Задание 5. Решить задачу. Дано целое число N (N&gt; 0). Найти значение выражения 1.1 – 1.2 + 1.3 – … (N слагаемых, знаки чередуются). Условный оператор не использовать.

Листинг программы:

using System;

namespace ConsoleApp

{

class Program

{

static void Main(string[] args)

{

Console.Write("n=");

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

double rez = 0;

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

rez += Math.Pow(-1, i + 1) \* (1.0 + (double)i / 10);

Console.WriteLine("Result=" + rez);

Console.ReadKey();

}

}

}

Таблица 1.5 – Выходные и входные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 5 | 1,3 |

Анализ результатов:



Рисунок 1.6 – Результат работы программы

Источник: собственная разработка

Задание 6. Табулирование функций.

Постановка задачи: Составить программу вычисления значений функции F(x) на отрезке [A, B] в точках x i =x+H, где H=(B-A)/M, M – заданное целое число.

Листинг программы:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace MyNameSpace

{

internal class Program

{

static void Main(string[] args)

{

Console.Write("Введите M: ");

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

double A = (Math.Sqrt(2)) / (2);

double B = 1;

double H = (B - A) / M;

double x = 0;

x = x + H;

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

{

double y = 2 - Math.Sin(x);

Console.WriteLine($"{i} - " + y);

x = x + H;

}

Console.ReadLine();

}

}

}

Таблица 1.6 – Выходные и входные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 4 | 1,92  1,85  1,78  1,71 |

Анализ результатов:

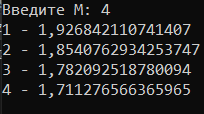


Рисунок 1.7 – Результат работы программы

Источник: собственная разработка