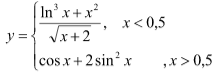
**1 Принципы ООП**

Задание №1. Написать программу, которая вычисляет значение функции у:



Листинг программы:

static void Main(string[] args)

{

double x, y;

while (true)

{

try

{

Console.Write("Enter x value (more or less than 0.5): ");

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

if (x < 0)

{

throw new Exception();

}

break;

}

catch (Exception)

{

Console.WriteLine("Value cannot be a negative number or letter");

}

}

if (x < 0.5)

{

y = (Math.Pow(Math.Log10(x), 3) + Math.Pow(x, 2)) / Math.Sqrt(x + 2);

}

else

{

y = Math.Cos(x) + 2 \* Math.Pow(Math.Sin(x), 2);

}

Console.WriteLine($"y = {Math.Round(y, 4)}");

}

Таблица 1.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 1 | 1,9564 |

Источник: собственная разработка

Анализ результатов представлен на рисунке 1.1.



Рисунок 1.1 – Результат работы программы

Источник: собственная разработка

Задание №2. Написать программу, которая определяет, все ли цифры трехзначного числа одинаковые.

Листинг программы:

static void Main(string[] args)

{

Console.WriteLine("Enter 3-digit value:");

string str = Console.ReadLine();

if (str.Substring(0, 1) == str.Substring(1, 1) && str.Substring(1, 1) == str.Substring(1, 1))

{

Console.WriteLine("digits in number are the same");

}

else Console.WriteLine("digits are not the same");

}

}

Таблица 1.2 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 888 | Digits in number are the same |

Источник: собственная разработка

Анализ результатов представлен на рисунке 1.2.



Рисунок 1.2 – Результат работы программы

Источник: собственная разработка

Задание №3. Составить программу (при решении данных задач использовать оператор switch или вложенные операторы if). В шеренге спортсмены стоят по росту. Определить по номеру спортсмена вид спорта, которым он занимается: 1, 2, 9 - баскетбол, 3, 4, 5 - бег, 6, 7, 8 - штанга.

Листинг программы:

static void Main(string[] args)

{

int number = 0;

while (true)

{

try

{

Console.Write("Enter value: ");

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

break;

}

catch (Exception)

{

Console.WriteLine("Value is incorrect!");

}

}

switch (number)

{

case 1:

case 2:

case 9:

Console.WriteLine("Backetball");

break;

case 3:

case 4:

case 5:

Console.WriteLine("Run");

break;

case 6:

case 7:

case 8:

Console.WriteLine("Barbell");

break;

default:

break;

}

}

Таблица 1.3 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 8 | Barbell |

Источник: собственная разработка

Анализ результатов представлен на рисунке 1.3.



Рисунок 1.3 – Результат работы программы

Источник: собственная разработка

Задание №4. Вывести на экран (задачу решите тремя способами - используя операторы цикла while, do while и for). все четные числа из диапазона от А до В, кратные пяти (A <= B).

Листинг программы:

static void Main(string[] args)

{

Console.Write("Enter A value: ");

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

Console.Write("Enter B value: ");

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

if (a % 2 != 0)

a++;

for (int i = a; i < b; i += 2)

if (i % 5 != 0)

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

Console.WriteLine();

int c = a;

while (c < b)

{

if (c % 5 != 0 && c % 2 == 0)

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

c++;

}

Console.WriteLine();

c = a;

do

{

if (c % 5 != 0 && c % 2 == 0)

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

c++;

}

while (c < b);

}

Таблица 1.4 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 1, 10 | 2, 4, 6, 8 |

Источник: собственная разработка

Анализ результатов представлен на рисунке 1.4.

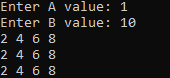


Рисунок 1.4 – Результат работы программы

Источник: собственная разработка

Задание №5. Дано вещественное число A и целое число N (N > 0). Найти A в степени N: AN = A·A· … ·A (числа A перемножаются N раз).

Входные данные: ввести вещественное число А (-5<= A <=5) и целое число N (1<= n <=10).

Выходные данные: вывести A в степени N с точностью до 4 цифр в дробной части.

Листинг программы:

static void Main(string[] args)

{

int N, A;

while (true)

{

try

{

Console.Write("Enter N value: ");

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

Console.Write("Enter A value: ");

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

if (!((A <= 5 && A >= -5) || (N <= 10 && N >= 1)))

{

throw new Exception();

}

break;

}

catch (Exception)

{

Console.WriteLine("Value is incorrect!");

}

}

Console.WriteLine($"Number {A} in Pow {N} = {Math.Round(Math.Pow(A, N), 4)}");

}

Таблица 1.5 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 19, 3 | 1162261467 |

Источник: собственная разработка

Анализ результатов представлен на рисунке 1.5:



Рисунок 1.5 – Результат работы программы

Источник: собственная разработка

Задание №6. 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. F(x) = Sin(x) - Tan(x), A = 0, B = π / 4, M = 20.

Листинг программы:

static void Main(string[] args)

{

const double A = 0;

const double B = Math.PI / 4;

const double M = 20;

double h = (B - A) / M;

double result = 0;

double x = 0;

for (double i = 0.1; i < 2.1; i += 0.1)

{

x = i + h;

result = Math.Sin(x) - Math.Tan(x);

Console.WriteLine($"x = {Math.Round(x,4)}");

}

}

Таблица 1.6 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
|  | 0.1393, 0.4393, 1.1393, 2.0393 |

Источник: собственная разработка

Анализ результатов представлен на рисунке 1.6.

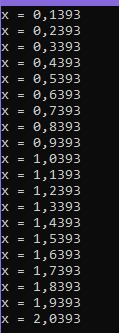


Рисунок 1.6 – Результат работы программы

Источник: собственная разработка

Задание №3.3-9. Так называемая сиракузская последовательность чисел для заданного числа n строится следующим образом. Если n чётное, то оно делится на 2, иначе заменяется на 3n + 1. Затем действия повторяются над полученным числом.

Листинг программы:

static void Main(string[] args)

{

double res;

Console.Write("Enter n value: ");

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

res = n;

Console.Write("Result: ");

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

{

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

if (n == 1)

break;

if (n % 2 == 0)

{

n /= 2;

}

else

{

n = 3 \* n + 1;

}

}

}

Таблица 1.7 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 5 | 5, 16, 8, 4, 2, 1 |

Источник: собственная разработка

Анализ результатов представлен на рисунке 1.7.



Рисунок 1.7 – Результат работы программы

Источник: собственная разработка

Задание №7. Даны два числа. Найти среднее арифметическое и среднее геометрическое их модулей.

Листинг программы:

static void Main(string[] args)

{;

double firstNumber = 0, secondNumber = 0;

while (true)

{

try

{

Console.Write("First value: ");

firstNumber = Math.Abs(double.Parse(Console.ReadLine()));

Console.Write("Second value: ");

secondNumber = Math.Abs(double.Parse(Console.ReadLine()));

break;

}

catch (Exception)

{

Console.WriteLine("Value is incorrect!");

}

}

Console.WriteLine($"Avarage - {(firstNumber + secondNumber) / 2}");

Console.WriteLine($"Geometric mean - {(firstNumber \* secondNumber) / 2}");

}

Таблица 1.8 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 6, 19 | 12.5, 57 |

Источник: собственная разработка

Анализ результатов представлен на рисунке 1.8:

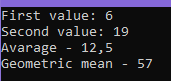


Рисунок 1.8 – Результат работы программы

Источник: собственная разработка

Задание №8. Дано целое число, лежащее в диапазоне от –999 до 999. Вывести строку – словесное описание данного числа вида «отрицательное двузначное число», «нулевое число», «положительное однозначное число» и т.д.

Листинг программы:

static void Main(string[] args)

{

int N;

while (true)

{

try

{

Console.Write("Enter value(from -999 to 999): ");

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

if (N < -999 || N > 999)

{

throw new Exception();

}

break;

}

catch (Exception)

{

Console.WriteLine("Value is incorrect!");

}

}

if (N == 0) Console.WriteLine("Zero number");

else

{

if (N < 0)

{

Console.WriteLine("Negative");

}

else Console.WriteLine("Positive");

if (Math.Abs(N) / 100 > 0) Console.WriteLine("3-digit");

else if (Math.Abs(N) / 10 > 0) Console.WriteLine("2-digit");

else Console.WriteLine("1-digit");

}

}

Таблица 1.9 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| -674 | Negative, 3-digit |

Источник: собственная разработка

Анализ результатов представлен на рисунке 1.9.



Рисунок 1.9 – Результат работы программы

Источник: собственная разработка

Задание №9. Дано вещественное число A и целое число N (N> 0). Используя один цикл, найти сумму 1 + A + A 2 + A 3 + … + A N.

Входные данные: ввести вещественное число А (-5<=A <=5) и целое число N (1<=n <=10).

Выходные данные: вывести сумму 1 + A + A 2 + A 3 + … + A N с точностью до 4 цифр в дробной части.

Листинг программы:

static void Main(string[] args)

{

double N;

double A;

while (true)

{

try

{

Console.Write("Enter N value: ");

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

Console.Write("Enter A value: ");

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

if (!((A <= 5 && A >= -5) || (N <= 10 && N >= 1)))

{

throw new Exception();

}

break;

}

catch (Exception)

{

Console.WriteLine("Value is incorrect!");

}

}

double sum = 1;

for (double i = 1; i <= N; i++)

{

sum += (double)(Math.Pow(A, i));

}

Console.WriteLine("Sum = {0:N4}", sum);

}

Таблица 1.10 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 2.2, 5.1 | 32.1100 |

Источник: собственная разработка

Анализ результатов представлен на рисунке 1.10.



Рисунок 1.10 – Результат работы программы

Источник: собственная разработка