**2 Классы. Библиотеки FCL. Классы как типы и объекты этих типов**

Задание №1. Метод вычисления значения выражения , метод возведения в квадрат суммы a и b.

Листинг программы:

class A

{

public int a;

public int b;

public A(int a\_, int b\_)

{

this.a = a\_; this.b = b\_;

}

public double MakeSomethinWrong()

{

return (double)((Math.Sin(b) + 4) / (2 \* a));

}

public double MakeSomethinWrong2()

{

return (double)(Math.Pow(a + b, 2));

}

}

class Temp {

static void Main(string[] args)

{

A classA = new A(5, 5);

Console.WriteLine($"(Math.Sin(b) + 4) / (2 \* a) = {classA.MakeSomethinWrong()}");

Console.WriteLine($"(a + b)^2 = {classA.MakeSomethinWrong2()}");

}

}

Таблица 2.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 5, 5 | 0.30410, 100 |

Источник: собственная разработка

Анализ результатов представлен на рисунке 2.1.

C:\Users\Leo\Downloads\photo1652521229.jpeg

Рисунок 2.1 – Результат работы программы

Источник: собственная разработка

Задание №2. Построить иерархию классов в соответствии с вариантом задания: Игрушка, продукт, товар, молочный продукт.

Листинг программы:

class Goods

{

public string Name { get; set; }

public decimal Price { get; set; }

public string Status { get; set; }

public string Owner { get; set; }

public void GetStatus() { }

public void Buy() { }

}

class Toy : Goods

{

public string Type { get; set; }

public string Material { get; set; }

public void PlayWithToy() { }

}

class Product : Goods

{

public string Period { get; set; }

public string TypeOfProduct { get; set; }

public void GetTypeOfProduct() { }

}

class MilkProduct : Product

{

public string typeOfMilkProduct { get; set; }

public void GetTypeOfMilkProduct() { }

}

Таблица 2.2 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Name, Price, Status, Type, Material | GetStatus(), Buy(), PlayWithToy() |

Источник: собственная разработка

Анализ результатов представлен на рисунке 2.2.

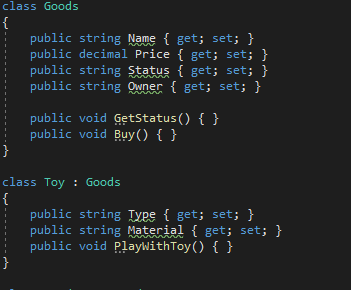


Рисунок 2.2 – Результат работы программы

Источник: собственная разработка

Задание №3. Составить описание класса многочлена вида ах2 + bх + с. Предусмотреть методы, реализующие:

вычисление значения многочлена для заданного аргумента;

операцию сложения, вычитания и умножения многочленов с получением

нового объекта-многочлена;

вывод на экран описания многочлена.

Написать программу, демонстрирующую все разработанные элементы класса.

Листинг программы:

class Mnogochlen

{

private double[] A;

public Mnogochlen(double[] parametrs)

{

A = parametrs;

}

public void VivodMnogochlena()

{

for (int i = 0; i < A.Length - 1; i++)

Console.Write("{0}x^{1} + ", A[i], A.Length - i - 1);

Console.Write(A[A.Length - 1] + "\n");

}

public double ZnachenieMnogochlena(double x)/// возвращает значение многочлена в точке х

{

double rezalt = 0;

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

{

double StepenX = 1;

for (int j = 0; j < A.Length - i - 1; j++)

StepenX \*= x;

rezalt += StepenX \* A[i];

}

return rezalt;

}

public static Mnogochlen operator +(Mnogochlen mn1, Mnogochlen mn2)

{

Mnogochlen mnRezalt = new Mnogochlen(new double[Math.Max(mn1.A.Length, mn2.A.Length)]); // создание массива, в который будет сложен результат. его размер будет соответсвовать большему из исходных.

for (int i = mnRezalt.A.Length - 1; i >= 0; i--)

{

if (i < mn1.A.Length)

mnRezalt.A[i] += mn1.A[i];

if (i < mn2.A.Length)

mnRezalt.A[i] += mn2.A[i];

}

return mnRezalt;

}

public static Mnogochlen operator \*(Mnogochlen mn1, Mnogochlen mn2)

{

Mnogochlen mnRezult = new Mnogochlen(new double[mn1.A.Length + mn2.A.Length - 1]);

for (int i1 = 0; i1 < mn1.A.Length; i1++)

{

for (int i2 = 0; i2 < mn2.A.Length; i2++)

mnRezult.A[i1 + i2] += mn1.A[i1] \* mn2.A[i2];

}

return mnRezult;

}

public static Mnogochlen operator -(Mnogochlen mn1, Mnogochlen mn2)

{

Mnogochlen mnRezalt = new Mnogochlen(new double[Math.Max(mn1.A.Length, mn2.A.Length)]); // создание массива, в который будет сложен результат. его размер будет соответсвовать большему из исходных.

for (int i = mnRezalt.A.Length - 1; i >= 0; i--)

{

if (i < mn1.A.Length)

mnRezalt.A[i] += mn1.A[i];

if (i < mn2.A.Length)

mnRezalt.A[i] -= mn2.A[i];

}

return mnRezalt;

}

}

class Temp

{

static void Main()

{

Mnogochlen testMnogochlen1 = new Mnogochlen(new double[] { 3, 4, 5 });

Mnogochlen testMnogochlen2 = new Mnogochlen(new double[] { 1, 1, 1 });

testMnogochlen1.VivodMnogochlena();

testMnogochlen2.VivodMnogochlena();

Console.WriteLine("введите переменную для нахождения значения первого многочлена");

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

Console.WriteLine("значение многочлена в точке x= {0} будет {1}", x, testMnogochlen1.ZnachenieMnogochlena(x));

Mnogochlen summa = testMnogochlen1 + testMnogochlen2;

Mnogochlen proizvedenie = testMnogochlen1 \* testMnogochlen2;

Mnogochlen difference = testMnogochlen1 - testMnogochlen2;

Console.Write("сумма многочленов равна: ");

summa.VivodMnogochlena();

Console.Write("произведение моногочленов равно:");

proizvedenie.VivodMnogochlena();

Console.Write("Разность моногочленов равна:");

difference.VivodMnogochlena();

}

}

Таблица 2.3 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 6 | 137, 4x22 + 5x1 + 6, 3x4 + 7x3 + 12x2 + 9x1 + 5, 2x2 + 3x1 + 4 |

Источник: собственная разработка

Анализ результатов представлен на рисунке 2.3.

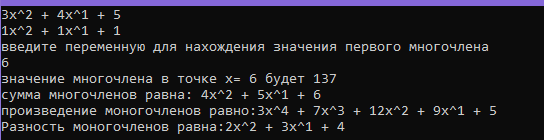


Рисунок 2.3 – Результат работы программы

Источник: собственная разработка