**2 Классы**

Задание 1. Создать класс А с целочисленными полями а и b и двумя методами согласно варианту. Внутри класса реализовать конструктор для инициализации a и b. Создать объект класса и продемонстрировать работу со всеми элементами класса.

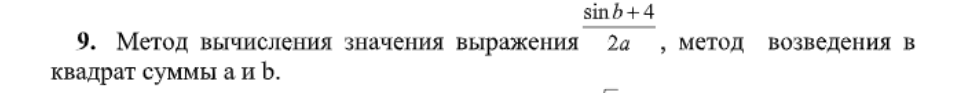


Рисунок 2.1 – Выражение

Листинг программы:

namespace zad1

{

class Program

{

public static void Main()

{

Console.WriteLine("Введите число а: ");

double num1 = Convert.ToDouble(Console.ReadLine());

Console.WriteLine("Введите число b: ");

double num2 = Convert.ToDouble(Console.ReadLine());

A arel = new A(num1, num2);

arel.vzv();

arel.vkz();

Console.ReadLine();

}

class A

{

public double a = 1; // поле данных

public double b = 2; // поле данных

public A(double a, double b)

{

this.a = a;

this.b = b;

}

public void vzv()

{

Console.WriteLine($"{Math.Pow((a + b), 2)}");

}

public void vkz()

{

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

}

}

}

}

Таблица 1.1 – Выходные и входные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 2, 5 | 49, 0.7 |

Анализ результатов:

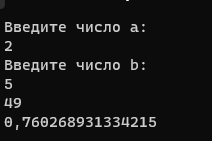


Рисунок 2.1 – Результат работы программы

Источник: собственная разработка

Задание 2. Построить иерархию классов в соответствии с вариантом задания: Игрушка, продукт, товар, молочный продукт.

Листинг программы:

namespace zad2

{

internal class Program

{

static void Main(string[] args)

{

Product[] products = new Product[]

{

new MilkTovar("Cheese",5.15M,14),

new Igrushka("Mishka",20M,"Sweet"),

new Tovar("Soufa", 10000M)

};

foreach (Product product in products)

{

Console.WriteLine(product);

}

//Tovar arel = new Igrushka("Pony");

//arel.Naz();

Console.ReadLine();

}

class Igrushka:Tovar

{

public string Kind { get; set; }

public Igrushka(string name,decimal cost, string kind):base(name,cost)

{

Kind = kind;

}

public override string ToString()

{

return $"Kind - {Kind} ";

}

//public void naz()

//{

// console.writeline($"игрушка - {name}");

//

}

class Product

{

public string Name { get; set; }

//public int Cost { get; set; }

//public int day = 14;

//public string Nazvanie\_e = "Tv";

//public int Huge = 15;

public Product(string name)

{

Name = name;

//Cost = cost;

}

public override string ToString()

{

return $"Name - {Name} ";

}

//public void AboutProduct()

//{

// Console.WriteLine($"Продукт: {Name}");

//}

}

class Tovar:Product

{

public decimal Cost { get; set; }

public Tovar(string name, decimal cost):base(name)

{

Cost = cost;

}

public override string ToString()

{

return base.ToString()+

$"Cost - {Cost} ";

}

//public void AboutTovar()

//{

// base.AboutProduct();

// Console.WriteLine($"Cost - {Cost} ");

//}

}

class MilkTovar : Tovar

{

public int Days { get; set; }

public MilkTovar(string name, decimal cost, int days):base(name,cost)

{

Days = days;

}

public override string ToString()

{

return base.ToString()+

$"Days - {Days} ";

}

//public void AboutMilkTovar()

//{

// Console.WriteLine($"Название молочного товара: {Nazvanie}, Срок годности в течении {day} дней ");

//}

}

}

}

Таблица 2.2 – Выходные и входные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| - | Name – Cheese Cost – 5,15 Days –14  Kind – Sweet  Name – Soufa Cost -10000 |

Анализ результатов:

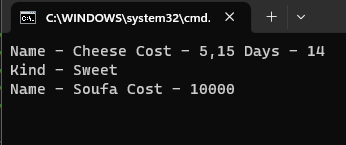


Рисунок 2.2 – Результат работы программы

Источник: собственная разработка

Задание 3. Составить описание класса многочлена вида ах 2 + 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:

namespace zad3

{

class Program

{

class Mnogochlen

{

private double[] A;//единственное поле класса(массив коэффициэнтов)

public Mnogochlen (double[] VvodA)//конструктор класса, вызывается при создании нового элемента класса, для поддержки инкапсуляции, если она будет нужна

{

A=VvodA;

}

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 SlogitMnogochleni(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 UmnogitMnogochleni(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;

}

}

static void Main(string[] args)

{

Mnogochlen testMnogochlen1 = new Mnogochlen(new double[] {1,2,3,4,5}); //создание представителя класса с именем testMnogochlen и массивом таким-то.

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

testMnogochlen1.VivodMnogochlena();

testMnogochlen2.VivodMnogochlena();

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

double x= double.Parse(Console.ReadLine()); //ввод с клавиатуры переменной X;

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

Mnogochlen summa = Mnogochlen.SlogitMnogochleni(testMnogochlen1,testMnogochlen2);

Mnogochlen proizvedenie = Mnogochlen.UmnogitMnogochleni(testMnogochlen1,testMnogochlen2);

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

summa.VivodMnogochlena();

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

proizvedenie.VivodMnogochlena();

}

}

}

Таблица 1.3 – Выходные и входные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 2 | Значение многочлена, сумма многочленов, произведение многочленов. |

Анализ результатов:

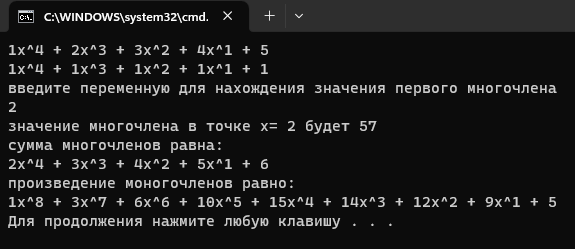


Рисунок 2.3 – Результат работы программы

Источник: собственная разработка