**3 Процедуры и функции – методы класса**

Задание №1. Разработать метод f(x), который в трехзначном числе меняет местами первую с последней цифрой, а остальные числа оставляет без изменения. Продемонстрировать работу данного метода на примере.

Листинг программы:

class Program

{

static void F(int number)

{

int firstDigit = (number / 100) % 10;

int secondDigit = (number / 10) % 10;

int thirdDigit = (number / 1) % 10;

int[] array = new int[3];

array[0] = thirdDigit;

array[1] = secondDigit;

array[2] = firstDigit;

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

{

Console.Write(array[i]);

}

}

static void Main()

{

Console.Write("Enter three-digit number: ");

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

F(number);

}

}

Таблица 1.1 – Входные и выходные данных задание №1

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 123 | 321 |

Анализ результатов:



Рисунок 1.1 – Результат работы программы из задания №1

Источник: собственная разработка

Задание №2. Постройте таблицу значений функции y=f(x) для x ∈ [a,b] с шагом h. Для решения задачи используйте вспомогательный метод.

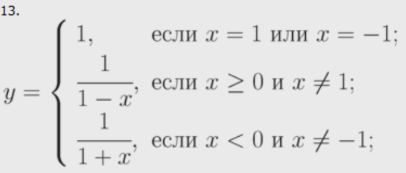


Рисунок 1.2 – Формулы к заданию 2

Источник: задание 2

Листинг программы:

namespace z2

{

class Program

{

static void Main(string[] args)

{

double x; double a; double b; double h;

Console.Write("a=: ");

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

Console.Write("b=: ");

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

Console.Write("h=: ");

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

for (double i = a; i <= b; i += h)

{

Console.Write($"f({i})={F(i)}");

}

}

static double F(double x)

{

double y = 0;

if (x == 1 || x == -1) y = 1;

if (x >= 0 && x != 1) y = (1 / (1 - x));

if (x < 0 && x != -1) y = (1 / (1 + x));

return y;

}

}

}

Таблица 1.2 – Входные и выходные данных задание №2

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 3  4  3 | -0.5 |

Анализ результатов:

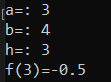


Рисунок 1.3 – Результат работы программы из задания №2

Источник: собственная разработка

Задание №3. Перегрузите метод f из предыдущего раздела так, чтобы его сигнатура (заголовок) соответствовала виду static void f (double x, out double y). Продемонстрируйте работу перегруженных методов.

Листинг программы:

class Program

{

static void Main(string[] args)

{

double x; double a; double b; double h;

Console.Write("a=: ");

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

Console.Write("b=: ");

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

Console.Write("h=: ");

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

for (double i = a; i <= b; i += h)

{

F(i, out double y);

Console.WriteLine($"f({i})={y}");

}

}

static void F(double x, out double y)

{

y = 0;

if (x == 1 || x == -1) y = 1;

if (x >= 0 && x != 1) y = (1 / (1 - x));

if (x < 0 && x != -1) y = (1 / (1 + x));

}

}

Таблица 1.3 – Входные и выходные данных задание №3

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 3  4  3 | -0.5 |

Анализ результатов:

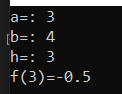


Рисунок 1.4 – Результат работы программы из задания №3

Источник: собственная разработка

Задание №4. Создайте проект, в котором опишите класс для решения задачи Вашего варианта. Разрабатываемый класс должен содержать следующие элементы: скрытые и открытые поля, конструкторы, перегруженные операции. В программе должна выполняться проверка всех разработанных элементов класса. Описать класс для работы с двумерными массивами чисел. Реализовать возможность выполнения для согласованных массивов комбинированных операций присваивания (+=, -=).

Листинг программы:

namespace ConsoleApplication3

{

class Program

{

public class Matrix

{

private double[,] matrix;

int Row = 0, Col = 0;

public Matrix(int row, int col)

{

matrix = new double[row, col];

Row = row; Col = col;

}

public Matrix(int N)

{

matrix = new double[N, N];

Row = Col = N;

}

public double this[int i, int j]

{

get { return matrix[i, j]; }

set { matrix[i, j] = value; }

}

public static Matrix operator +(Matrix first, Matrix second)

{

Matrix mat = new Matrix(first.Row, first.Col);

for (int i = 0; i < first.Row; i++)

for (int j = 0; j < first.Col; j++)

mat[i, j] = first[i, j] + second[i, j];

return mat;

}

public static Matrix operator -(Matrix first, Matrix second)

{

Matrix mat = new Matrix(first.Row, first.Col);

for (int i = 0; i < first.Row; i++)

for (int j = 0; j < first.Col; j++)

mat[i, j] = first[i, j] - second[i, j];

return mat;

}

public void PrintMatrix()

{

for (int i = 0; i < this.Row; i++)

{

for (int j = 0; j < this.Col; j++)

Console.Write("{0} ", this[i, j]);

Console.Write("\n");

}

}

}

static void Main(string[] args)

{

int N = 3;

Random rand = new Random();

Matrix first = new Matrix(N);

Matrix second = new Matrix(N);

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

for (int j = 0; j < N; j++)

{

first[i, j] = rand.Next(1, 5);

second[i, j] = rand.Next(1, 5);

}

Console.WriteLine("First matrix :\n\n");

first.PrintMatrix();

Console.WriteLine("\n\nSecond matrix:\n\n");

second.PrintMatrix();

Console.WriteLine("\n\nMatrix sum :\n\n");

(first + second).PrintMatrix();

Console.WriteLine("\n\nMatrix difference:\n\n");

(first - second).PrintMatrix();

Console.ReadKey();

}

}

}

Таблица 1.4 – Входные и выходные данных задание №4

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
|  | First matrix:  1 2 1  1 1 4  2 1 2  Second matrix:  2 4 2  2 1 3  2 2 3  Matrix sum:  3 6 3  3 2 7  4 3 5  Matrix difference:  -1 -2 -1  -1 0 1  0 -1 -1 |

Анализ результатов:

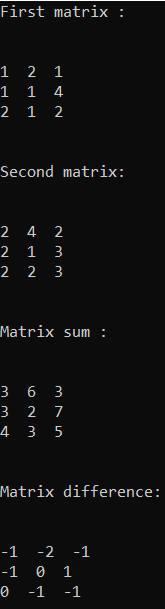


Рисунок 1.5 – Результат работы программы из задания №4

Источник: собственная разработка

Задание №5. Каждый разрабатываемый класс должен, как правило, содержать следующие элементы: скрытые поля, конструкторы с параметрами и без параметров, методы; свойства, перегруженные операции. В программе должна выполняться проверка всех разработанных элементов класса.

Листинг программы:

using z5;

class Program

{

static void Main()

{

Store groshik = new Store();

Product milk = new Product("Молоко", "Грошик", 2.08);

Product bread = new Product("Хлеб купаловский", "Грошик", 0.49);

Product candy = new Product("Бешенные пчёлки", "Грошик", 3.28);

Product[] products = new Product[3];

products[0] = milk;

products[1] = bread;

products[2] = candy ;

groshik.setTmp(products);

groshik.printInformation();

}

}

namespace z5

{

class Store

{

private Product[] products;

public void setTmp(Product[] product)

{

this.products = product;

}

public void printInformation()

{

foreach (Product product in products)

{

Console.WriteLine(product.getName());

Console.WriteLine($"Цена: {product.getPrice()}p");

Console.WriteLine($"Продается в: {product.getNameOfStore()}");

Console.WriteLine();

}

}

}

}

internal class Product

{

private string name;

private string nameOfStore;

private double price;

public Product(string name, string nameOfStore, double price)

{

this.name = name;

this.nameOfStore = nameOfStore;

this.price = price;

}

public string getName()

{

return name;

}

public string getNameOfStore()

{

return nameOfStore;

}

public double getPrice()

{

return price;

}

}

Таблица 1.5 – Входные и выходные данных задание №5

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
|  | Молоко  Цена: 2.08p  Продается в: Грошик  Хлеб купаловский  Цена: 0.49p  Продается в: Грошик  Бешенные пчёлки  Цена: 3.28p  Продается в: Грошик |

Анализ результатов:

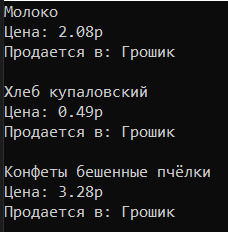


Рисунок 1.6 – Результат работы программы из задания №5

Источник: собственная разработка