**3 Процедуры и функции – методы класса**

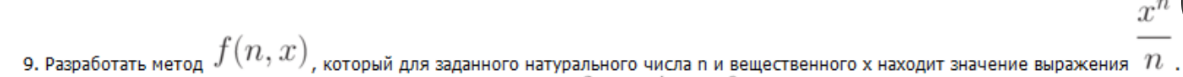
Задание 1. Разработать метод f(n, x), который для заданного натурального и вещественного числа и вещественного x находит значение выражения (x\*\*n) / n

Рисунок 3.1 – Значение выражения

Листинг программы:

namespace zad1

{

internal class Program

{

static void F(double x, double n)

{

double y;

y = (Math.Pow(x,n)) / (n);

Console.WriteLine(y);

}

static void Main(string[] args)

{

double n;

double x;

Console.Write("Введите x: ");

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

Console.Write("Введите n: ");

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

F(x,n);

Console.ReadLine();

}

}

}

Таблица 3.1 – Выходные и входные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| X = 3, n = 2 | 4,5 |

Анализ результатов:

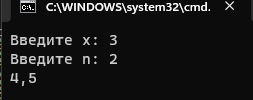


Рисунок 3.2 – Результат работы программы

Источник: собственная разработка

Задание 2. 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.

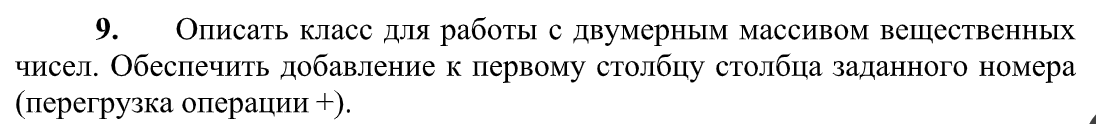


Рисунок 3.3 – Условие задания

Источник: собственная разработка

Листинг программы:

namespace zad2

{

public class Array

{

public int[,] Arr;

public Array(int row, int col)

{

this.Arr = new int[row, col];

}

public int GetValue(int row, int col)

{

return Arr[row, col];

}

public int SetValue(int row, int col, int val)

{

return Arr[row, col] = val;

}

public int GetLength(int d)

{

return Arr.GetLength(d);

}

public Array GetArr(Array arr)

{

Random random = new Random();

for (int i = 0; i < arr.GetLength(0); i++)

{

for (int j = 0; j < arr.GetLength(1); j++)

{

SetValue(i, j, random.Next(10));

Console.Write("{0,4}", arr.GetValue(i, j));

}

Console.WriteLine();

}

return arr;

}

public static Array operator +(Array Arr, int num)

{

for (int j = 0; j < Arr.GetLength(0); j++)

{

Arr.SetValue(j, 0, Arr.GetValue(j, 0) + Arr.GetValue(j, num));

}

return Arr;

}

public void Print(Array Arr)

{

for (int i = 0; i < Arr.GetLength(0); i++)

{

for (int j = 0; j < Arr.GetLength(1); j++)

{

Console.Write("{0,4}", Arr.GetValue(i, j));

}

Console.WriteLine();

}

}

}

class Program

{

static void Main(string[] args)

{

Array Arr = new Array(3, 4);

Arr.GetArr(Arr);

Console.WriteLine();

Arr.Print(Arr + 2);

Console.ReadLine();

}

}

}

Таблица 3.2 – Выходные и входные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 9, 2, 0 | 16, 2, 9 |

Анализ результатов:

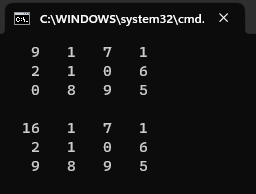


Рисунок 3.4– Результат работы программы

Источник: собственная разработка

Задание 3. 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.

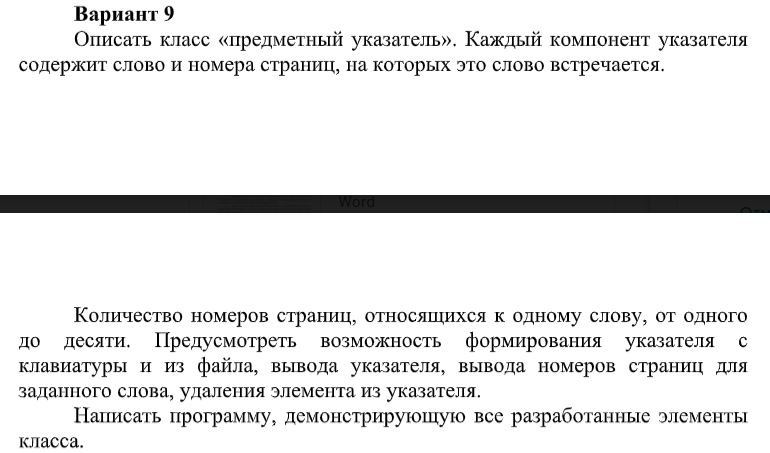


Рисунок 3.5 – Условие задания

Источник: собственная разработка

Листинг программы:

namespace zad3

{

struct Params

{

public string word; //слово из указателя

public int[] page; //массив с номерами страниц

public Params(string word, int[] page) //конструктор с параметрами

{

this.word = word;

this.page = page;

}

}

class Program

{

static void Main(string[] args)

{

Params[] to = new Params[4];

//ввод данных

Console.Write("Введите количество слов, которое вы желаете ввести: ");

string buf = Console.ReadLine();

for (int i = 1; i <= Convert.ToInt32(buf); i++)

{

Console.Write("Введите слово в указатель: ");

to[i].word = Console.ReadLine();

Console.Write("Введите количество страниц для этого слова: ");

string buf1 = Console.ReadLine();

int buf11 = Convert.ToInt32(buf1);

Console.Write("Введите номера страниц через запятую(,): ");

string TempPage = Console.ReadLine();

string[] TempPageMassive = TempPage.Split(',');

int[] page = new int[] {

int.Parse(TempPageMassive[0].ToString()),

int.Parse(TempPageMassive[1].ToString()),

int.Parse(TempPageMassive[2].ToString()) };

to[i].page = new int[buf11];

to[i].page = page;

}

//вывод введеных с клавиатуры данных

Console.WriteLine();

for (int k = 1; k <= Convert.ToInt32(buf); ++k)

{

Console.WriteLine("{0}\t\t{1},{2},{3}", to[k].word, to[k].page[0], to[k].page[1], to[k].page[2]);

}

Console.ReadLine();

}

}

}

Таблица 3.3 – Выходные и входные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Два слова: Привет, как – 3 страницы | Привет – стр. 1, 2, 3  Как – стр. 1, 2, 3 |

Анализ результатов:

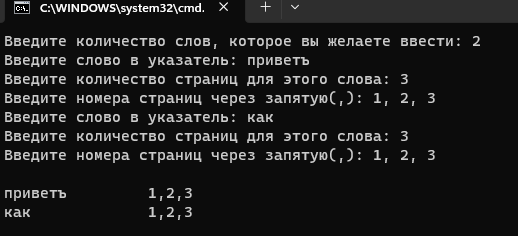


Рисунок 3.6 – Результат работы программы

Источник: собственная разработка

разработка