**19 ДОКУМЕНТИРОВАНИЕ В ФОРМАТЕ XML**

Задание 1. Выполнить XML документирование кода.

Листинг программы 4.1:

class Temp

{

static void Main(string[] args)

{

/// <summary>

/// Formula value calculation method

/// </summary>

/// <param name="n">x power value</param>

/// <param name="x">new x value</param>

/// <returns>

/// Returns the value of the calculated formula

/// </returns>

static double Formula(int n, double x)

{

return Math.Pow(x, n) / n;

}

Console.WriteLine(Formula(2, 4) + Formula(4, 4) + Formula(6, 4));

}

}

Листининг программы 4.3:

class Array

{

/// <summary>

/// Array

/// </summary>

private double[,] \_data;

/// <summary>

/// Number of row.

/// </summary>

public int RowCount => \_data.GetLength(0);

/// <summary>

/// Number of column.

/// </summary>

public int ColumnCount => \_data.GetLength(1);

/// <summary>

/// Enter or return value.

/// </summary>

/// <param name="x">row</param>

/// <param name="y">col</param>

/// <returns>value</returns>

/// <exception cref="Exception">x, y not in row or(/and) col</exception>

public double this[int x, int y]

{

get { if (x < 0 || x > RowCount || y < 0 || y > ColumnCount) { throw new Exception(); } return \_data[x, y]; }

set { \_data[x, y] = value; }

}

/// <summary>

/// Array creater.

/// </summary>

/// <param name="dim1">number of rows</param>

/// <param name="dim2">number of cols</param>

public Array(int dim1, int dim2)

{

\_data = new double[dim1, dim2];

}

/// <summary>

/// Print mass.

/// </summary>

public void Print()

{

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

{

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

{

Console.Write(\_data[i, j] + " ");

}

Console.WriteLine();

}

}

/// <summary>

/// Get length, this array.

/// </summary>

/// <param name="d">type</param>

/// <returns>length this array</returns>

public int GetLength(int d)

{

return \_data.GetLength(d);

}

/// <summary>

/// Get element value.

/// </summary>

/// <param name="i1">row</param>

/// <param name="i2">col</param>

/// <returns>value in this row and col</returns>

public double GetValue(int i1, int i2)

{

return \_data[i1, i2];

}

/// <summary>

/// Setting a value.

/// </summary>

/// <param name="i1">row</param>

/// <param name="i2">col</param>

/// <param name="val">value</param>

public void SetValue(int i1, int i2, int val)

{

\_data[i1, i2] = val;

}

/// <summary>

/// Аdding to the first column of the given number column.

/// </summary>

/// <param name="array">Source array</param>

/// <param name="index">column number</param>

/// <returns>Returns an array with new values</returns>

/// <exception cref="ArgumentException">Column outside array dimension</exception>

public static Array operator +(Array array, int index)

{

if (index < 0 || index > array.ColumnCount)

{

throw new ArgumentException();

}

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

{

array[i, 0] += array[i, index];

}

return array;

}

}

class Temp

{

static void Main(string[] args)

{

Console.Write("Создаём двумерный массив [{0},{1}]...", 5, 5);

Array array1 = new Array(5, 5);

Console.Write("\nСоздаём двумерный массив [{0},{1}]...", 5, 5);

Array array2 = new Array(5, 5);

Console.WriteLine("\n");

int min = -10, max = 10;

Random r = new Random();

Console.WriteLine("Заполняем массив 1 случайными числами из отрезка [{0},{1}]", min, max);

for (int i = 0; i < array1.GetLength(0); i++)

{

for (int j = 0; j < array1.GetLength(1); j++)

{

array1.SetValue(i, j, r.Next(min, max));

Console.Write("{0,3} ", array1.GetValue(i, j));

}

Console.WriteLine();

}

Console.WriteLine("Заполняем массив 2 случайными числами из отрезка [{0},{1}]", min, max);

for (int i = 0; i < array2.GetLength(0); i++)

{

for (int j = 0; j < array2.GetLength(1); j++)

{

array2.SetValue(i, j, r.Next(min, max));

Console.Write("{0,3} ", array2.GetValue(i, j));

}

Console.WriteLine();

}

Console.WriteLine("Добавление 4-ого столбца к первому для перваго массива: ");

array1 += 4;

array1.Print();

Console.WriteLine();

}

}