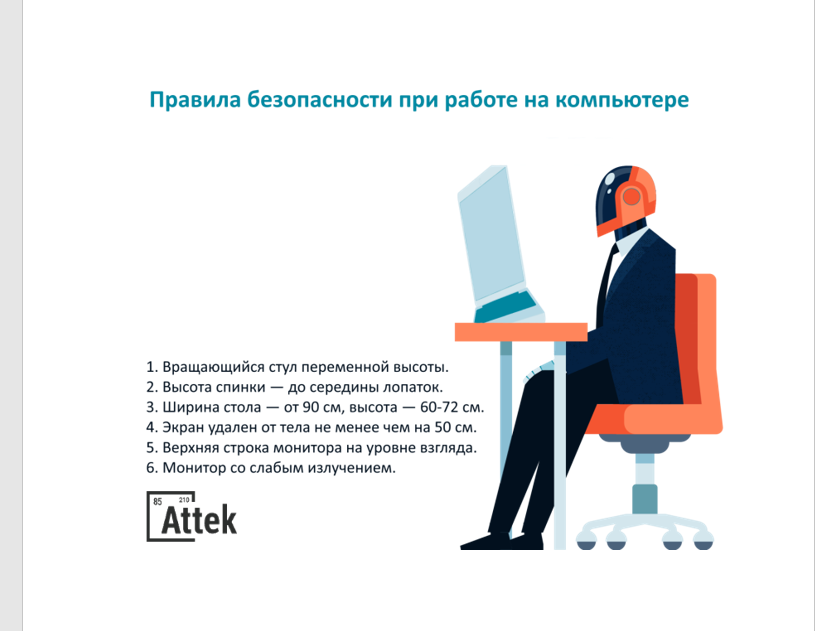
*ПРИЛОЖЕНИЯ*

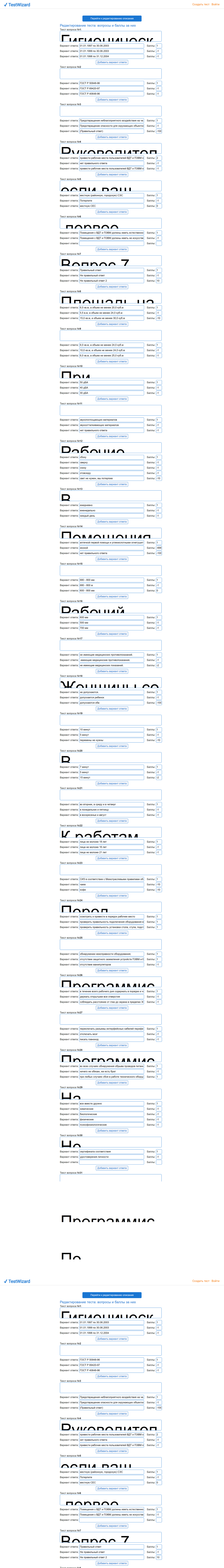
*ПРИЛОЖЕНИЕ А*

*Плакат:*

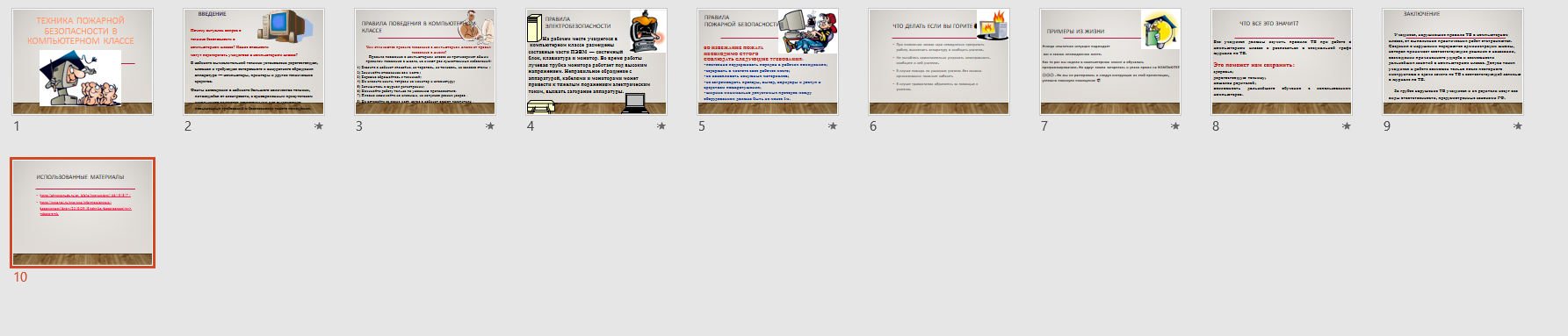


*Тест*

*https://www.testwizard.ru/result.php?tid=59104*



*Презентация*

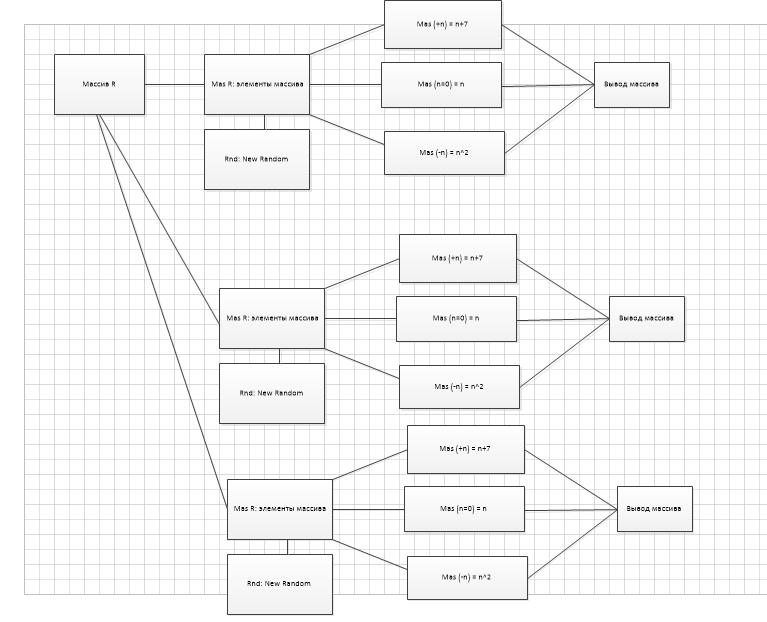


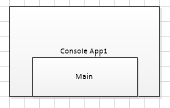
*ПРИЛОЖЕНИЕ Б*

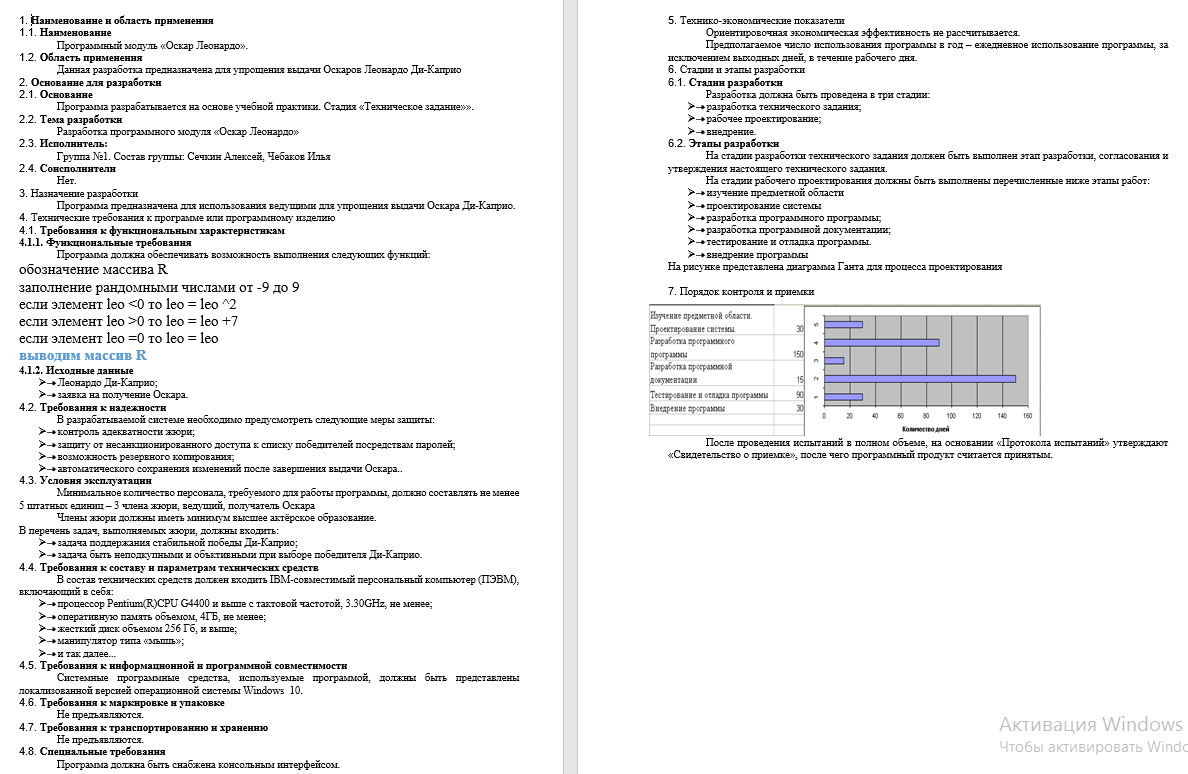
|  |  |
| --- | --- |
| 4 | Задача №1  В массиве R, содержащем 25 элементов, заменить значения отрицательных элементов квадратами значений, значения положительных увеличить на 7, а нулевые значения оставить без изменения. Вывести массив R.  Задача №2  Написать метод, вычисляющий значение *n*/*xn*. С его помощью вычислить выражение: |

|  |  |
| --- | --- |
| Параметр | Значение |
| 1 | 2 |
| Основные предметно-значимые сущности | mas R 25 целых элементов (5;5) |
| Основные предметно-значимые атрибуты сущностей | обозначение массива R  заполнение рандомными числами от -9 до 9  если элемент leo <0 то leo1 = leo ^2  если элемент leo >0 то leo2 = leo +7  если элемент leo =0 то leo3 = leo  выводим массив R |
| Основные требования к функциям системы: | leo1 – отрицательный элемент (<0); leo2 – положительный элемент(>0); leo3 – нулевое значение элемента(=0) |
| Дополнительно |  |

|  |  |
| --- | --- |
| Параметр | Значение |
| 1 | 2 |
| Основные предметно-значимые сущности | x, n;с |
| Основные предметно-значимые атрибуты сущностей | ввод переменных n, x;  вставка переменных в данную формулу *n*/*xn*;  вводим арифметический цикл for (i=1, i<=10, i++); для вычисления выражения ;  с=c+i/x^I;  выводим результат с; |
| Основные требования к функциям системы: | x – любое вещественное число !=0  n – любое вещественное число  с=0 |
| Дополнительно |  |







*ПРИЛОЖЕНИЕ В*

Составить диаграмму прецендентов в соответствии с разработанным техническим заданием.



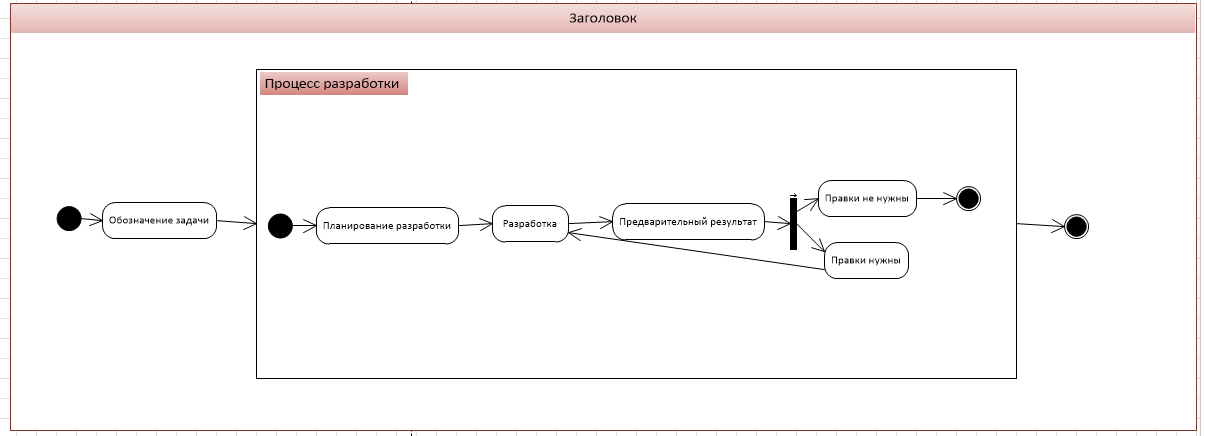
Составить диаграмму последовательностей в соответствии с разработанным техническим заданием.



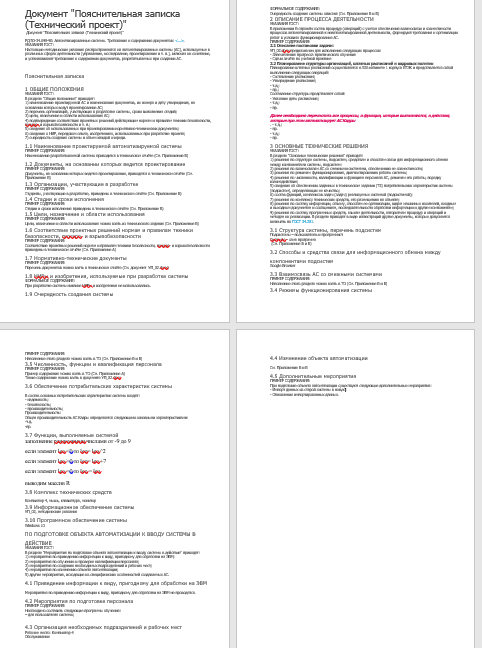
Оформить внешнюю спецификацию к задаче по плану:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Название подсистемы | Название функции | Информационная среда | | | |
| Входные данные | | Выходные данные | |
| Назначение (наименование) | Тип, ограничения | Назначение (наименование) | Тип, ограничения |
| Mas R | main | mas R  25 целых элементов (5;5); | заполнение рандомными числами от -9 до 9 | mas R | ‑ |
| main |  | если элемент leo <0 то leo = leo ^2  если элемент leo >0 то leo = leo +7  если элемент leo =0 то leo = leo |
| Код ошибки | целое |

1. Разработать диаграмму состояний для каждой подсистемы в спецификации(<https://flexberry.github.io/ru/fd_statechart-diagram.html> )

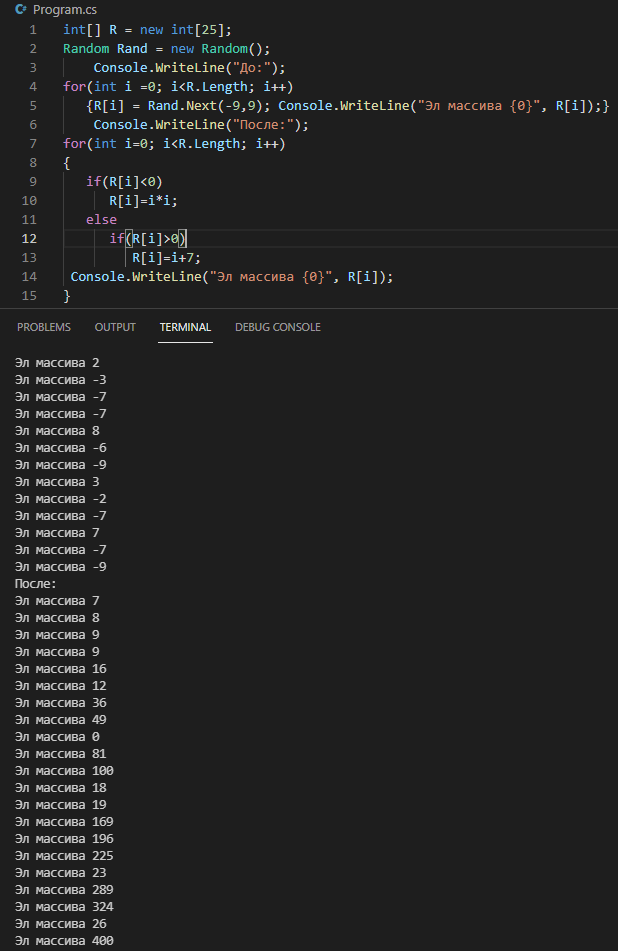


1. Разработать пояснительную записку по разработанному техническому заданию



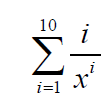
*ПРИЛОЖЕНИЕ Г*

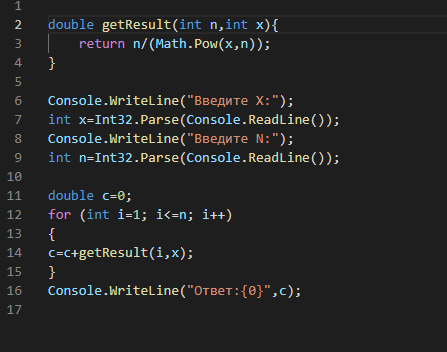
Задача №1

В массиве R, содержащем 25 элементов, заменить значения отрицательных элементов квадратами значений, значения положительных увеличить на 7, а нулевые значения оставить без изменения. Вывести массив R. 

Задача №2

Написать метод, вычисляющий значение *n*/*xn*. С его помощью вычислить выражение:





Задача №3

Написать метод, находящий сумму матриц одинакового размера и возвращающий новую матрицу. С помощью этого метода обработать пары матриц и отобразить результаты на экране.

using System;

// класс с методами расширения

static class MatrixExt

{

    // метод расширения для получения количества строк матрицы

    public static int RowsCount(this int[,] matrix)

    {

        return matrix.GetLength(0);

    }

    // метод расширения для получения количества столбцов матрицы

    public static int ColumnsCount(this int[,] matrix)

    {

        return matrix.GetLength(1);

    }

}

class Program

{

    // метод для получения матрицы из консоли

    static int[,] GetMatrixFromConsole(string name)

    {

        Console.Write("Количество строк матрицы {0}:    ", name);

        var r = int.Parse(Console.ReadLine());

        Console.Write("Количество столбцов матрицы {0}: ", name);

        var c = int.Parse(Console.ReadLine());

        var matrix = new int[r, c];

        for (var i = 0; i < r; i++)

        {

            for (var j = 0; j < c; j++)

            {

                Console.Write("{0}[{1},{2}] = ", name, i, j);

                matrix[i, j] = int.Parse(Console.ReadLine());

            }

        }

        return matrix;

    }

    // метод для вывода матрицы на экран

    static void PrintMatrix(int[,] matrix)

    {

        for (var i = 0; i < matrix.RowsCount(); i++)

        {

            for (var j = 0; j < matrix.ColumnsCount(); j++)

            {

                Console.Write(matrix[i, j].ToString().PadLeft(4));

            }

            Console.WriteLine();

        }

    }

    // метод для сложения двух матриц

    static int[,] MatrixSum(int[,] matrixA, int[,] matrixB)

    {

        if ((matrixA.ColumnsCount() != matrixB.ColumnsCount()) || (matrixA.RowsCount() != matrixB.RowsCount()))

        {

            throw new Exception("Для матриц с разным размером сложение не возможно!");

        }

        var matrixC = new int[matrixA.RowsCount(), matrixB.ColumnsCount()];

        for (var i = 0; i < matrixA.RowsCount(); i++)

        {

            for (var j = 0; j < matrixB.ColumnsCount(); j++)

            {

                matrixC[i, j] = matrixA[i, j] + matrixB[i, j];

            }

        }

        return matrixC;

    }

    static void Main(string[] args)

    {

        Console.WriteLine("Программа для сложения двух матриц");

        var a = GetMatrixFromConsole("A");

        var b = GetMatrixFromConsole("B");

        Console.WriteLine("Матрица A:");

        PrintMatrix(a);

        Console.WriteLine("Матрица B:");

        PrintMatrix(b);

        var result = MatrixSum(a, b);

        Console.WriteLine("Сумма матриц:");

        PrintMatrix(result);

        Console.ReadLine();

    }

}

