12. Перечислите этапы создания проекта консольной программы.

Запускаем Visual Studio.

Создание нового проекта: File —> New —> Project (в русской версии данное меню будет выглядеть так – Файл —> Создать —> Проект).

В окне New Project (Новый Проект) на левой панели (Project Types) (Типы Проектов) нужно выбрать язык (Visual С#) и платформу (Windows). На центральной панели выбрать вид приложения Console Application (Консольное Приложение).

В поле Name (Имя) можно напечатать вместо установленного по умолчанию имени свое. В поле Location (Путь) нужно ввести полное имя папки, в которой будет сохранено решение, например, С:Программы. По умолчанию решению приписывается стандартное имя, но его можно заменить на, например, Program\_l (Стандартным является ConsoleApplication (Цифра начиная от 1 в зависимости от того сколько приложений создано со стандартным именем проекта). Кнопкой ОКзапустить процесс создания проекта (и решения).

Среда Visual Studio создает решение, проект приложения и открывает окно редактора с готовой заготовкой исходного кода программы следующего вида

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Program\_l

{

class Program

{

static void Main(string[] args)

{

}

}

}

52. Опишите отношение клиенты – поставщики.

Класс является обобщенным понятием, определяющим характеристики и поведение некоторого множества конкретных объектов этого класса, называемых экземплярами класса. «Классический» класс содержит данные, задающие свойства объектов класса, и функции, определяющие их поведение. В последнее время в класс часто добавляется третья составляющая - события, на которые может реагировать объект класса.

Классы программной системы находятся в определенных отношениях друг с другом.

**Определение**: Классы A и В находятся в отношении "клиент - поставщик", если полем класса А является объект класса В, и хотя бы в одном из методов класса А происходит вызов свойств или методов класса В.

Следуя этому определению, объект класса B "вложен" в класс A. По этой причине отношение "клиент - поставщик" называют также отношением вложенности или встраивания.

65. Напишите программу, которая определяет число отрицатель­ных чисел во введенной с клавиатуры последовательности (длина последовательности неограниченна).

using System;

using System.Collections.Generic;

using System.Text;

namespace ZADANIE65

{

class Program

{

static void Main(string[] args)

{

Console.Title = "Программа определяющая число отрицательных чисел во введенной с клавиатуры последовательности";

Console.WriteLine("Задание 65 - Шифр 29 - \n Второй Курс Могилёвского Государственного Политехнического Колледжа - Сделано Савичем Андреем Олеговичем");

Console.WriteLine("Пожалуйста Введите последовательность");

double max = +1;

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

double t = 0.0;

int i = 0;

do

{

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

i ++;

}

while(t < max);

Console.WriteLine("Количество Отрицательных чисел - {0}", i);

Console.ReadLine();

}

}

}



76. Решите задачу. Дан двумерный массив. Заполните его по строкам с клавиатуры и определите:

количество строк, не содержащих ни одного нулевого элемента;

максимальное из чисел, в заданной строке массива.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

class Program

{

static void Main()

{

Console.Write("Введите количество строк: ");

int firstDimension = int.Parse(Console.ReadLine());

Console.WriteLine();

Console.Write("Введите количество столбцов: ");

int secondDimension = int.Parse(Console.ReadLine());

Console.WriteLine();

int[,] array = new int[firstDimension, secondDimension];

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

{

string enterString = Console.ReadLine();

string[] massiveString = enterString.Split(new Char[] { ' ' });

for (int j = 0; j < massiveString.Length; j++)

{

array[i, j] = int.Parse(massiveString[j]);

}

}

// Перебираем каждый элемент матрицы и если он равен 0, тогда инкрементируем локальную переменную kolElem и

// выводим потом на экран в каждой строке. Если строка не содержит нулевые элементы матрицы, инкрементируем

// локальную переменную kolStr

int kolElem = 0;

int kolStr = 0;

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

{

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

{

if (array[i, j] == 0)

{

++kolElem;

}

}

Console.WriteLine("Строка {0} содержит {1} нулевых элементов", (i+1), kolElem);

if (kolElem == 0)

{

++kolStr;

}

else

{

kolElem = 0;

}

}

// Выводим на экран локальную переменную kolStr

Console.WriteLine("Количество строк не содержащих нулевые элементы: " + kolStr);

Console.WriteLine();

//Находим максимальное значение в избранной строке матрицы

Console.Write("Введите строку в которой хотите произвести поиск: ");

int row = int.Parse(Console.ReadLine());

int max = array[row, 0];

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

{

if (array[row, j] > max)

{

max = array[row, j];

}

}

Console.WriteLine("Максимальный элемент в строке {0} равен {1}",row+1 , max);

Console.ReadKey();

}

}

