**17 Сборки, библиотеки, атрибуты, директивы**

Задание 1 Решить задачи с использованием класса Stack. Пусть символ # определен в текстовом редакторе как стирающий символ Backspace, т.е. строка abc#d##c в действительности является строкой ac. Дан текст, в котором встречается символ #. Преобразовать его с учетом действия этого символа.

Листинг программы:

using System;

using System.Collections.Generic;

class Program

{

static string ProcessString(string input)

{

Stack<char> stack = new Stack<char>();

foreach (char c in input)

{

if (c == '#')

{

if (stack.Count > 0) stack.Pop();

}

else

{

stack.Push(c);

}

}

return new string(stack.ToArray().Reverse().ToArray());

}

static void Main()

{

Console.Write("Введите строку: ");

string userInput = Console.ReadLine();

string result = ProcessString(userInput);

Console.WriteLine($"Преобразованная строка: {result}");

}

}

Анализ результатов:



Рисунок 1 – Результат работы программы

Задание 2 Дан файл, содержащий числа. За один просмотр файла напечатать элементы файла в следующем порядке: сначала все числа, из интервала [a,b], потом

все числа, меньшие a, потом все числа, большие b, сохраняя исходный порядок в каждой группе чисел.

Листинг программы:

using System;

using System.Collections.Generic;

using System.IO;

class Program

{

static void Main(string[] args)

{

string fileName = "numbers.txt";

File.WriteAllLines(fileName, new string[]

{

"12", "5", "20", "7", "15", "30", "3", "10", "25"

});

Console.WriteLine($"Файл {fileName} создан с тестовыми числами.");

Console.Write("Введите границы интервала [a, b]: ");

string[] intervalInput = Console.ReadLine().Split();

int a = int.Parse(intervalInput[0]);

int b = int.Parse(intervalInput[1]);

Queue<int> inRange = new Queue<int>();

Queue<int> lessThanA = new Queue<int>();

Queue<int> greaterThanB = new Queue<int>();

foreach (string line in File.ReadLines(fileName))

{

if (int.TryParse(line, out int number))

{

if (number >= a && number <= b)

inRange.Enqueue(number);

else if (number < a)

lessThanA.Enqueue(number);

else

greaterThanB.Enqueue(number);

}

}

Console.WriteLine("Числа из интервала [a, b]: " + string.Join(" ", inRange));

Console.WriteLine("Числа меньше a: " + string.Join(" ", lessThanA));

Console.WriteLine("Числа больше b: " + string.Join(" ", greaterThanB));

}

}

Анализ результатов:

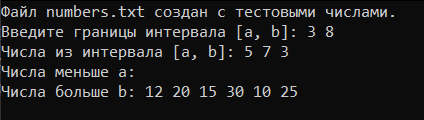


Рисунок 2 – Результат работы программы