**18 Коллекции и классы прототипы**

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:

class Program

{

static void Main(string[] args)

{

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

Console.Write("Введите строку: ");

string s = Console.ReadLine();

for (int i = 0; i < s.Length; i++)

{

str.Push(s[i]);

}

s = "";

char ch;

int count = str.Count;

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

{

ch = str.Pop();

if (ch == '#')

{

str.Pop();

i++;

}

else

{ s += ch;}

}

Console.WriteLine("Результат работы: {0}", s);

Console.ReadLine();

}

}

Таблица 1.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| fuyfy#rdt#k | kdrfyuf |

Анализ результатов:



Рисунок 1.1 – Результат работы программы

Задание 2. Дан текстовый файл. За один просмотр файла напечатать элементы файла в следующем порядке: сначала все символы, отличные от цифр, а затем все цифры, сохраняя исходный порядок в каждой группе символов.

Листинг программы:

class Program

{

static void Main(string[] args)

{

Queue<char> symbols = new Queue<char>();

Queue<char> digits = new Queue<char>();

using (StreamReader sr = new StreamReader(@"D:\pr18z2.txt"))

{

while (!sr.EndOfStream)

{

char s = (char)sr.Read();

if (char.IsDigit(s)) digits.Enqueue(s);

else symbols.Enqueue(s);

}

Console.WriteLine(string.Join("", symbols));

Console.WriteLine(string.Join("", digits));

Console.ReadKey();

}

}

Таблица 1.2 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
|  |  |

Анализ результатов:

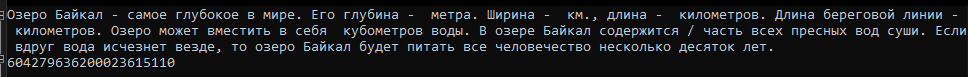


Рисунок 1.2 – Результат работы программы

Задание 3. Создать класс-прототип, с обобщенным методом, с методами добавления, удаления, клонирования экземпляров класса.

В основной программе продемонстрировать:

функциональность созданного класса и всех его методов;

использование обнуляемых типов;

использование абстрактных структур (List<P>; Dictionary<Key,Value>).

Вариант: спорт.

Листинг программы:

public class Program

{

static void Main(string[] args)

{

Sp SpList = new Sp();

Dictionary<string, string> city = new Dictionary<string, string>(2);

for (int i = 0; i < 2; i++)

{

Console.WriteLine("Введите по какому виду спорта будет проходить игра: ");

string s = null;

s = Console.ReadLine();

Console.WriteLine("Введите город");

string g = Console.ReadLine();

Console.WriteLine("Введите дату проведения (Например: 10 Марта): ");

string ds = Console.ReadLine();

SpList.Add(new DataS(s, g, ds));

var key = $"{i + 1}";

city[key] = g;

}

Console.WriteLine("=======================");

foreach (KeyValuePair<string, string> keyValue in city)

{

Console.WriteLine(keyValue.Key + " - " + keyValue.Value);

}

Console.WriteLine("=======================");

foreach (DataS x in SpList)

x.PrintSport();

SpList.RemoveAt(1);

Console.WriteLine("=======================");

Sp c1 = (Sp)SpList.Clone();

foreach (DataS x in c1)

x.PrintSport();

Console.ReadLine();

}

}

Таблица 1.3 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| теннису  Гродно  10 июля  футболу  Гродно  13 июля | =======================  1 - Гродно  2 - Гродно  =======================  В Гродно состоится игра по теннису: дата проведения 10 июля  В Гродно состоится игра по футболу: дата проведения 13 июля  ======================= |

Анализ результатов:

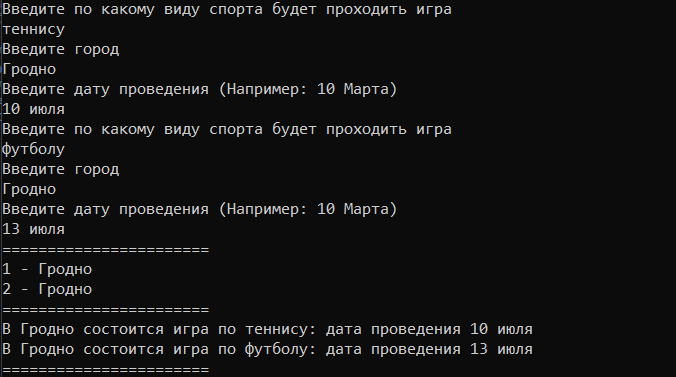


Рисунок 1.3 – Результат работы программы