**18. Коллекции. 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:

Console.Write("Enter line:");

string line = Console.ReadLine();

Stack stack = new Stack();

foreach (char c in line)

{ if (c == '#' && stack.Count > 0)

{ stack.Pop(); }

else if (c != '#')

{ stack.Push(c);}}

char[] arr = new char[stack.Count];

for (int i = stack.Count - 1; i >= 0; i--)

{ arr[i] = (char)stack.Pop();}

string result = new string(arr);

Console.WriteLine("Результат: " + result);

Таблица 18.1 – Входные и выходные данных задание №1

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| abc#d##c | ас |

Анализ результатов:

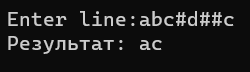


Рисунок 18.1 – Результат работы программы из задания №1

Задание №2. Решить следующие задачи с использованием класса Queue. 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:

string path = @"..\..\..\..\txt1.txt";

Console.WriteLine("Введите значения a:");

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

Console.WriteLine("Введите значения b:");

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

string[] input =path.Split(' ');

StreamReader reader = new StreamReader(path);

Queue interval = new Queue();

Queue lessA = new Queue();

Queue moreB = new Queue();

while (!reader.EndOfStream)

{ int num = int.Parse(reader.ReadLine());

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

{ interval.Enqueue(num); }

else if (num < a)

{ lessA.Enqueue(num);}

else if (num > b)

{ moreB.Enqueue(num); }}

reader.Close();

Console.WriteLine("Числа в интервале [a, b]:");

while (interval.Count > 0)

{ Console.Write(interval.Dequeue() + " "); }

Console.WriteLine();

Console.WriteLine("Числа меньше a:");

while (lessA.Count > 0)

{ Console.Write(lessA.Dequeue() + " ");}

Console.WriteLine();

Console.WriteLine("Числа больше b:");

while (moreB.Count > 0)

{ Console.Write(moreB.Dequeue() + " "); }

Console.WriteLine();

Таблица 18.2 – Входные и выходные данных задание №2

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| A=1  B=4 |  |

Анализ результатов:

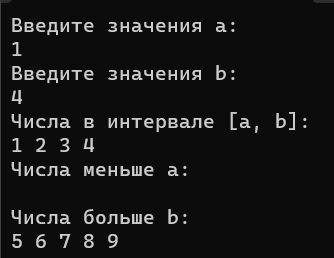


Рисунок 18.2 – Результат работы программы из задания №2

Задание №3. Решить задачу с использованием класса HashTable. Реализовать простейший каталог музыкальных компакт-дисков, который позволяет: добавлять и удалять диски, добавлять и удалять песни, просматривать содержимое целого каталога и каждого диска в отдельности.

Листинг программы:

class MusicCatalogue

{ private Hashtable discs;

public MusicCatalogue()

{ discs = new Hashtable(); }

public void AddDisc(string title)

{ discs[title] = new Hashtable(); }

public void RemoveDisc(string title)

{ discs.Remove(title); }

public void AddSong( string songTitle, string artist)

{ Hashtable songs = new Hashtable();

if (songs != null)

{ songs[songTitle] = artist}}

public void ViewCatalogue()

{ foreach (DictionaryEntry disc in discs)

{ Console.WriteLine(disc.Key);

Hashtable songs = (Hashtable)disc.Value;

if (songs != null)

{ foreach (DictionaryEntry song in songs)

{ Console.WriteLine(" " + song.Key + " - " + song.Value); }}}}

static void Main(string[] args)

{ MusicCatalogue catalogue = new MusicCatalogue();

catalogue.AddDisc("Песни 80х");

catalogue.AddDisc("Песни 90х");

catalogue.AddDisc("Песни 0х");

Console.WriteLine($"Каталог дисков");

catalogue.ViewCatalogue();

Console.WriteLine();

Console.WriteLine("Выберите операцию");

Console.WriteLine("1.Добавить диск \n2.Удалить диск");

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

switch (op)

{ case 1:

Console.WriteLine("Сколько дисков вы хотите добавить?");

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

if (n >= 1)

{ for (int j = 0; j < n; j++)

{ Console.WriteLine("Введите название диска для добавления");

string disk = Console.ReadLine();

catalogue.AddDisc(disk); } }

else

{ Console.WriteLine("Вы не хотите добавлять диски");}break;

case 2:

Console.WriteLine("Ведите диск, который хотите добавить");

string disk2 = Console.ReadLine();

catalogue.RemoveDisc(disk2);break;

default: Console.WriteLine("Такого пункта нет"); break; }

Console.WriteLine();

Console.WriteLine($"Каталог дисков");

catalogue.ViewCatalogue();

Console.WriteLine("Выберите номер диска, чтобы просмотреть информацию");

int disk3 = int.Parse(Console.ReadLine());

if (disk3 == 1)

{ Console.WriteLine("Песни на этом диске:");

catalogue.AddSong("Ветер с моря дул","Татьяна Буланова");

catalogue.AddSong("Чистый","Скриптонит");

catalogue.AddSong("Половинка","Танцы минус");}

else if (disk3==2)

{ Console.WriteLine("Песни на этом диске:");

catalogue.AddSong("Ясный мой свет", "Татьяна Буланова");

catalogue.AddSong("Танцуй", "Скриптонит");

catalogue.AddSong("Паранойа", "Танцы минус"); }

else if (disk3==3)

{ Console.WriteLine("Песни на этом диске:");

catalogue.AddSong("Ненаглядный", "Татьяна Буланова");

catalogue.AddSong("Baby", "Скриптонит");

catalogue.AddSong("Подожди", "Танцы минус");}

catalogue.ViewCatalogue();}}

Таблица 18.3 – Входные и выходные данных задание №3

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

Анализ результатов:

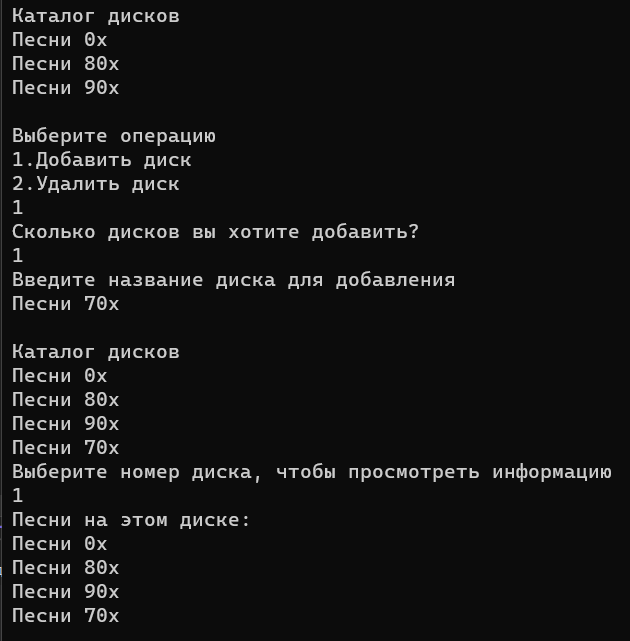


Рисунок 18.3 – Результат работы программы из задания №3