Министерство образования и молодежной политики Свердловской области

ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Отчёт по программе «Практическое занятие Коллекции в с# »

Выполнил: Меньшиков Артур Андреевич

Группа: ПР-22

Преподаватель: Мирошниченко Г.В

2024

**Задание:**

**4 Решить задачу, используя класс 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.
* Осуществлять поиск всех записей заданного исполнителя по всему каталогу.

**Входные и выходные данные**

ht: хэш-таблица/Hashtable

songs: список/ArrayList

textbox1: string/строка

textbox2: string/строка

textbox3: string/строка

textbox4: string/строка

**Блок-схема**

**Листинг программы (если есть)**

namespace ind\_zad4

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

Music music = new Music();

private void add\_disk\_Click(object sender, EventArgs e)

{

try

{

if (textBox1.Text != "")

{

music.add\_disk(textBox1.Text);

listBox1.Items.Clear();

foreach (DictionaryEntry el in music.ht)

{

listBox1.Items.Add(el.Key);

}

}

else

{

MessageBox.Show("неверный ввод");

}

}

catch (Exception ex) { MessageBox.Show($"ошибка: {ex.Message}"); }

}

private void del\_disk\_Click(object sender, EventArgs e)

{

try

{

if (textBox1.Text != "" && listBox1.Items.Count > 0)

{

string disk = listBox1.SelectedItem.ToString();

music.del\_disk(disk);

listBox1.Items.RemoveAt(listBox1.SelectedIndex);

}

else

{

MessageBox.Show("выберите диск который хотите удалить");

}

}

catch (Exception ex) { MessageBox.Show($"ошибка: {ex.Message}"); }

}

private void add\_song\_Click(object sender, EventArgs e)

{

try

{

if (textBox2.Text != "" && listBox1.SelectedIndex != -1 && textBox3.Text != "")

{

label1.Text = "диск" + " " + textBox1.Text;

string disk = listBox1.SelectedItem.ToString();

string song = textBox2.Text + " - " + textBox3.Text;

music.add\_song(disk, song);

listBox2.Items.Clear();

foreach (var el in music.songs)

{

listBox2.Items.Add(el);

}

}

else { MessageBox.Show("выберите диск или введите все занчения"); }

}

catch (Exception ex) { MessageBox.Show($"ошибка: {ex.Message}"); }

}

private void del\_song\_Click(object sender, EventArgs e)

{

try

{

if (listBox1.SelectedIndex != -1 && listBox2.SelectedIndex != -1)

{

string disk = listBox1.SelectedItem.ToString();

string song = listBox2.SelectedItem.ToString();

music.del\_song(disk, song);

listBox2.Items.RemoveAt(listBox1.SelectedIndex);

}

else { MessageBox.Show("выберите песню"); }

}

catch (Exception ex) { MessageBox.Show($"ошибка: {ex.Message}"); }

}

private void listBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

label1.Text = "диск" + " " + listBox1.SelectedItem.ToString();

string disk = listBox1.SelectedItem.ToString();

music.song\_on\_disk(disk);

listBox2.Items.Clear();

foreach (var el in music.songs)

{

listBox2.Items.Add(el);

}

}

private void button1\_Click(object sender, EventArgs e)

{

try

{

if (textBox4.Text != "")

{

if (textBox4.Text != "")

{

listBox2.ClearSelected();

string searchTerm = textBox4.Text.ToLower();

List<int> indices = new List<int>();

for (int i = 0; i < listBox2.Items.Count; i++)

{

string currentSong = listBox2.Items[i].ToString().ToLower();

if (currentSong.Contains(searchTerm))

{

indices.Add(i);

}

}

if (indices.Count > 0)

{

foreach (var index in indices)

{

listBox2.SetSelected(index, true);

}

}

else { MessageBox.Show("Нет песен этого исполнителя"); }

}

} else { MessageBox.Show("Введите исполнителя"); }

}

catch (Exception ex) { MessageBox.Show($"Ошибка: {ex.Message}"); }

}

private void button2\_Click(object sender, EventArgs e)

{

label1.Text = "все диски";

listBox2.Items.Clear();

foreach (DictionaryEntry entry in music.ht)

{

ArrayList songs = (ArrayList)entry.Value;

foreach (string song in songs)

{

listBox2.Items.Add(song);

}

}

}

}

}

namespace ind\_zad4

{

class Music

{

public Hashtable ht = new Hashtable();

public ArrayList songs = new ArrayList();

public void add\_disk(string disk)

{

ht.Add(disk, new ArrayList());

}

public void del\_disk(string disk)

{

ht.Remove(disk);

}

public void add\_song(string disk, string song)

{

songs = (ArrayList)ht[disk];

songs.Add(song);

}

public void song\_on\_disk(string disk)

{

songs = (ArrayList)ht[disk];

}

public void del\_song(string disk, string song)

{

songs = (ArrayList)ht[disk];

songs.Remove(song);

}

}

}

**Тестовые ситуации**









