Министерство образования и молодежной политики Свердловской области

ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Отчёт по программе  «Практическое занятие 4»

Выполнил: Рудных Екатерина

Группа: ПР-21

Преподаватель: Мирошниченко Г.В

2025

**Задание 1:**

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 данные:**

1. Название диска
2. Информация о песне
3. Имя исполнителя

**Выходные данные:**

1. Список дисков
2. Результаты операций
3. Сообщения пользователю

**Листинг программы:**

using System;

using System.Collections;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace pr4

{

public partial class Form1 : Form

{

private Hashtable catalog = new Hashtable();

public Form1()

{

InitializeComponent();

}

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

{

string diskName = txtDiskName.Text.Trim();

if (string.IsNullOrEmpty(diskName))

{

MessageBox.Show("Введите название диска.");

return;

}

if (catalog.ContainsKey(diskName))

{

MessageBox.Show("Диск с таким названием уже существует.");

return;

}

catalog[diskName] = new ArrayList(); // Создаем новый список песен для диска

MessageBox.Show($"Диск '{diskName}' успешно добавлен.");

UpdateCatalogList();

}

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

{

string diskName = txtDiskName.Text.Trim();

if (string.IsNullOrEmpty(diskName))

{

MessageBox.Show("Введите название диска.");

return;

}

if (!catalog.ContainsKey(diskName))

{

MessageBox.Show("Диск с таким названием не найден.");

return;

}

catalog.Remove(diskName);

MessageBox.Show($"Диск '{diskName}' успешно удален.");

UpdateCatalogList();

}

private void button3\_Click(object sender, EventArgs e)

{

string diskName = txtDiskName.Text.Trim();

string songInfo = txtSongInfo.Text.Trim();

if (string.IsNullOrEmpty(diskName) || string.IsNullOrEmpty(songInfo))

{

MessageBox.Show("Введите название диска и информацию о песне.");

return;

}

if (!catalog.ContainsKey(diskName))

{

MessageBox.Show("Диск с таким названием не найден.");

return;

}

ArrayList songs = (ArrayList)catalog[diskName];

songs.Add(songInfo);

MessageBox.Show($"Песня '{songInfo}' успешно добавлена на диск '{diskName}'.");

UpdateDiskContent(diskName);

}

private void button4\_Click(object sender, EventArgs e)

{

string diskName = txtDiskName.Text.Trim();

string songInfo = txtSongInfo.Text.Trim();

if (string.IsNullOrEmpty(diskName) || string.IsNullOrEmpty(songInfo))

{

MessageBox.Show("Введите название диска и информацию о песне.");

return;

}

if (!catalog.ContainsKey(diskName))

{

MessageBox.Show("Диск с таким названием не найден.");

return;

}

ArrayList songs = (ArrayList)catalog[diskName];

if (!songs.Contains(songInfo))

{

MessageBox.Show("Песня не найдена на этом диске.");

return;

}

songs.Remove(songInfo);

MessageBox.Show($"Песня '{songInfo}' успешно удалена с диска '{diskName}'.");

UpdateDiskContent(diskName);

}

private void button5\_Click(object sender, EventArgs e)

{

lstResults.Items.Clear();

foreach (DictionaryEntry entry in catalog)

{

string diskName = entry.Key.ToString();

ArrayList songs = (ArrayList)entry.Value;

lstResults.Items.Add($"Диск: {diskName}");

foreach (string song in songs)

{

lstResults.Items.Add($" - {song}");

}

}

}

private void button6\_Click(object sender, EventArgs e)

{

string diskName = txtDiskName.Text.Trim();

if (string.IsNullOrEmpty(diskName))

{

MessageBox.Show("Введите название диска.");

return;

}

if (!catalog.ContainsKey(diskName))

{

MessageBox.Show("Диск с таким названием не найден.");

return;

}

UpdateDiskContent(diskName);

}

private void button7\_Click(object sender, EventArgs e)

{

string artist = txtArtist.Text.Trim();

if (string.IsNullOrEmpty(artist))

{

MessageBox.Show("Введите имя исполнителя.");

return;

}

lstResults.Items.Clear();

foreach (DictionaryEntry entry in catalog)

{

string diskName = entry.Key.ToString();

ArrayList songs = (ArrayList)entry.Value;

foreach (string song in songs)

{

if (song.Contains(artist))

{

lstResults.Items.Add($"Диск: {diskName}, Песня: {song}");

}

}

}

if (lstResults.Items.Count == 0)

{

MessageBox.Show("Записи с указанным исполнителем не найдены.");

}

}

private void UpdateCatalogList()

{

lstDisks.Items.Clear();

foreach (string diskName in catalog.Keys)

{

lstDisks.Items.Add(diskName);

}

}

private void UpdateDiskContent(string diskName)

{

lstResults.Items.Clear();

if (catalog.ContainsKey(diskName))

{

ArrayList songs = (ArrayList)catalog[diskName];

lstResults.Items.Add($"Диск: {diskName}");

foreach (string song in songs)

{

lstResults.Items.Add($" - {song}");

}

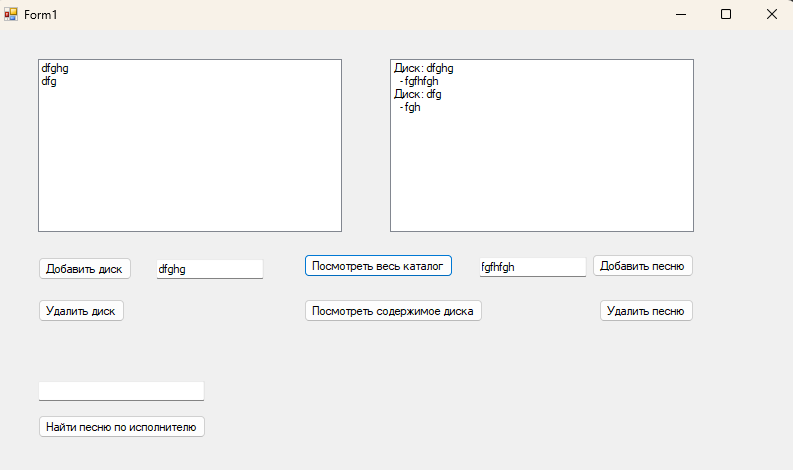
}

}

}

}

**Вывод:**

****

**Тестовые ситуации:**

