**Module 04 – DVD Collection**

Tim Mastarone

Rasmussen University

Microsoft C# Programming

Instructor: Jim Barringer

Module 04 - Lab 1

April 23, 2024

The DVD collection application displays a welcome message when the user runs the executable. They can enter one of 4 options (Open, New, Modify, and Close).

In the snip below, I ran the app and selected ‘o’ to open the file (DVD-Collection.csv) and display the contents (the main menu is shown again after the list is printed):

A screenshot of a computer program

Description automatically generated

If the user selects ‘n’ (‘N’ or a case insensitive ‘new’ are also accepted) they will enter a new title to be added to the file:

A screenshot of a computer program

Description automatically generated

In the snip above, an invalid year was entered. The program displays and erro message to the user and looks for new input.

The user can also choose to modify and entry. They will enter a DVD title name, and the program will search the file for a case insensitive match for the title. If it is found, the entry is deleted, and the user enters new information:

A screenshot of a computer program

Description automatically generated

The user can close the program from the main menu by entering ‘c’. There is no need to save the file as it is written as the entries are accepted as valid:

A screenshot of a computer program

Description automatically generated

All functionality of the application is in the Program.cs code below:

using System;

using System.Diagnostics.Metrics;

using System.IO;

using System.Reflection;

using System.Reflection.Metadata.Ecma335;

class Program

{

static void Main(string[] args)

{

string csvFile = "DVD-Collection.csv";//file is local in the project directory

// Check if the CSV file exists, create it if it doesn't

if (!File.Exists(csvFile))

{

// Create the CSV file with headers

using StreamWriter writer = new(csvFile);

writer.WriteLine("Name, Genre, Year, Rating");

}

//display the greeting

printGreeting();

//loop to get user's choice of action

bool exitLoop = false;

string? menuChoice;

while (!exitLoop)

{

printOptionsMenu();

menuChoice = Console.ReadLine();

if (menuChoice != null)

{

if (menuChoice.ToLower() == "o" || menuChoice.ToLower() == "open")

{

//open file and display the contents

List<DvdCollection> titles = LoadCsv(csvFile);// Load CSV data

// Display loaded data

foreach (DvdCollection title in titles)

{

Console.WriteLine($"Name: {title.Name}");

Console.WriteLine($"Genre: {title.Genre}");

Console.WriteLine($"Release Year: {title.Year}");

Console.WriteLine($"Rating: {title.Rating}");

Console.WriteLine("");

}

}

else if (menuChoice.ToLower() == "n" || menuChoice.ToLower() == "new")

{

//open the file and give the option to create a new entry

EnterData(csvFile);

}

else if (menuChoice.ToLower() == "m" || menuChoice.ToLower() == "modify")

{

//modify an entry in the file (if it exists)

Console.WriteLine("Enter the title of the DVD:");

string? enteredTitle = Console.ReadLine();

bool titleFound = false;

List<DvdCollection> titles = LoadCsv(csvFile);// Load CSV data

DvdCollection title;//temp holder for accessing data in the for loop

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

{

title = titles[i];

if (title.Name.ToLower() == enteredTitle)

{

titleFound = true;

//remove the found entry

titles.RemoveAt(i);

i--;//compensate for the removed entry

}

}

if (titleFound)

{

Console.WriteLine($"{enteredTitle} was found, please enter the new information:");

SaveData(csvFile, titles);

//prompt the user the enter a replacement

EnterData(csvFile);

}

}

else if (menuChoice.ToLower() == "c" || menuChoice.ToLower() == "close")

{

Console.WriteLine("Closing the program.");

exitLoop = true;

}

else

{

Console.WriteLine("Please enter a valid menu option.");

}

}

}//end of main menu loop

}//end of main() function

//Functions and Classes

//Function to load and print the saved csv data

static List<DvdCollection> LoadCsv(string csvFile)

{

List<DvdCollection> titles = [];

// Read lines from the CSV file

string[] lines = File.ReadAllLines(csvFile);

Console.WriteLine("");

Console.WriteLine("DVD Collection List:");

// Skip the header line (first line)

for (int i = 1; i < lines.Length; i++)

{

string[] fields = lines[i].Split(',');

if (fields.Length == 4)

{

string name = fields[0].Trim();

string genre = fields[1].Trim();

if (int.TryParse(fields[2].Trim(), out int year))

{

string email = fields[2].Trim();

string rating = fields[3].Trim();

DvdCollection title = new DvdCollection(name, genre, year, rating);

titles.Add(title);

}

}

//handle cases where the number of fields is not as expected

else

{

Console.WriteLine("Invalid data file. Please check the format of DVD-Collection.csv.");

Console.WriteLine("The file should have exactly four columns for each row entry.");

}

}

return titles;

}

//Function to enter s title and save it to the file

static void EnterData(string csvFile)

{

Console.WriteLine("Enter data:");

Console.Write("Name: ");

string? name = Console.ReadLine();

Console.Write("Genre: ");

string? genre = Console.ReadLine();

bool validInt = false;

string? enteredYear;

int year = 0;

while (!validInt)

{

Console.Write("Year: ");

enteredYear = Console.ReadLine();

if (int.TryParse(enteredYear, out year))

{

validInt = true;

}

else

{

Console.WriteLine("Please enter a valid year (numbers only)");

}

}

Console.Write("Rating: ");

string? rating = Console.ReadLine();

// Append the entered data to the CSV file

using (StreamWriter writer = new StreamWriter(csvFile, true))

{

writer.WriteLine($"{name}, {genre}, {year}, {rating}");

}

Console.WriteLine($"{name} has been added to the collection.");

Console.WriteLine("Data saved successfully.");

Console.WriteLine("");

}

// Function to save data to the CSV file with an instance of DvdCollection (a title)

static void SaveData(string csvFile, List<DvdCollection> titles)

{

// Overwrite the existing file with the modified data

using (StreamWriter writer = new StreamWriter(csvFile))

{

// Write header

writer.WriteLine("Name, Genre, Year, Rating");

// Write each DVD entry

foreach (DvdCollection title in titles)

{

writer.WriteLine($"{title.Name}, {title.Genre}, {title.Year}, {title.Rating}");

}

}

}

static void printGreeting()

{

Console.WriteLine("Welcome to the DVD collection application!");

Console.WriteLine("You will have options to view or modify a DVD collection to a file.");

Console.WriteLine("The contents of the collection will be saved in a csv file.");

Console.WriteLine("");

}

static void printOptionsMenu()

{

//show options for the user to choose from

Console.WriteLine("Select from an option below to view or modify the DVD Collection:");

Console.WriteLine("(O) Open - Load and display");

Console.WriteLine("(N) New - Add a new DVD");

Console.WriteLine("(M) Modify - Alter a DVD entry");

Console.WriteLine("(C) Close - Close file and quit the program");

Console.WriteLine("");

}

}

//class to hold a DVD entry for the collection

class DvdCollection

{

public string Name { get; set; }

public string Genre { get; set; }

public int Year { get; set; }

public string Rating { get; set; }

public DvdCollection(string name, string genre, int year, string rating)

{

Name = name;

Genre = genre;

Year = year;

Rating = rating;

}

}