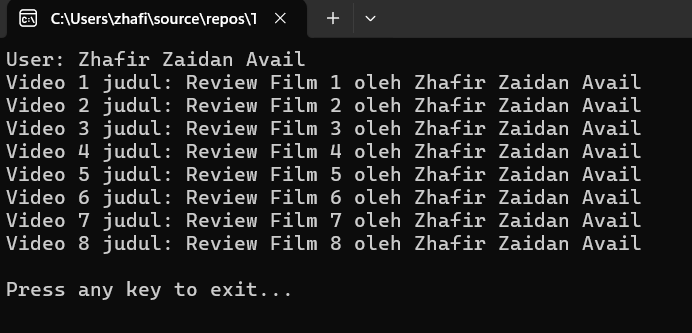
**TP\_MODUL\_6**

Zhafir Zaidan Avail

2311104059

Hasil Run:



Keknya salah sih tapi gatau lah, bingung njir

Source code:

1. SayaTubeVideo.cs

using System;

using System.Diagnostics;

public class SayaTubeVideo

{

private int id;

private string title;

private int playCount;

public SayaTubeVideo(string title)

{

Debug.Assert(title != null && title.Length <= 200, "Judul video harus tidak null dan maksimal 200 karakter");

Random rand = new Random();

this.id = rand.Next(10000, 99999); // Generate ID random 5 digit

this.title = title;

this.playCount = 0;

}

public void IncreasePlayCount(int count)

{

if (count <= 0 || count > 25000000)

{

Console.WriteLine("ERROR: Play count harus antara 1 hingga 25 juta!");

return;

}

try

{

checked

{

playCount += count;

}

}

catch (OverflowException)

{

Console.WriteLine("ERROR: Play count melebihi batas integer maksimum!");

}

}

public void PrintVideoDetails()

{

Console.WriteLine($"Video ID: {id}");

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

Console.WriteLine($"Play Count: {playCount}");

}

public int GetPlayCount()

{

return playCount;

}

public string GetTitle()

{

return title;

}

}

Penjelasan:

Kelas SayaTubeVideo merepresentasikan sebuah video dengan atribut id, title, dan playCount yang mencerminkan identitas, judul, serta jumlah tayangan video tersebut. Konstruktor memastikan bahwa title tidak null dan tidak melebihi 200 karakter menggunakan Debug.Assert() untuk validasi. id dihasilkan secara acak dengan nilai antara 10000 hingga 99999, sedangkan playCount dimulai dari nol. Metode IncreasePlayCount() memungkinkan penambahan jumlah tayangan dengan batas minimal 1 dan maksimal 25 juta, serta menangani kemungkinan OverflowException menggunakan checked. Metode PrintVideoDetails() menampilkan informasi video, sementara GetPlayCount() dan GetTitle() memungkinkan pengambilan data terkait jumlah tayangan dan judul video. Implementasi ini mengikuti prinsip Design by Contract, memastikan bahwa setiap metode mematuhi kondisi precondition, exception handling, dan postcondition agar data tetap valid serta program berjalan dengan aman.

1. SayaTubeUser.cs

using System;

using System.Collections.Generic;

using System.Diagnostics;

public class SayaTubeUser

{

private int id;

private string username;

private List<SayaTubeVideo> uploadedVideos;

public SayaTubeUser(string username)

{

Debug.Assert(username != null && username.Length <= 100, "Username harus tidak null dan maksimal 100 karakter");

Random rand = new Random();

this.id = rand.Next(10000, 99999); // Generate ID random 5 digit

this.username = username;

this.uploadedVideos = new List<SayaTubeVideo>();

}

public void AddVideo(SayaTubeVideo video)

{

Debug.Assert(video != null, "Video yang ditambahkan tidak boleh null");

uploadedVideos.Add(video);

}

public int GetTotalVideoPlayCount()

{

int totalPlayCount = 0;

foreach (var video in uploadedVideos)

{

totalPlayCount += video.GetPlayCount();

}

return totalPlayCount;

}

public void PrintAllVideoPlaycount()

{

Console.WriteLine($"User: {username}");

int maxPrint = Math.Min(8, uploadedVideos.Count); // Maksimal print 8 video

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

{

Console.WriteLine($"Video {i + 1} judul: {uploadedVideos[i].GetTitle()}");

}

}

}

Penjelasan:

Kelas SayaTubeUser merepresentasikan pengguna yang memiliki daftar video yang diunggah. Atribut yang dimiliki adalah id (identitas unik yang dihasilkan secara acak), username (nama pengguna dengan batasan maksimal 100 karakter), dan uploadedVideos (list yang menyimpan objek SayaTubeVideo). Konstruktor memastikan bahwa username valid menggunakan Debug.Assert(). Metode AddVideo() memungkinkan penambahan video dengan validasi agar tidak null. GetTotalVideoPlayCount() menghitung total jumlah tayangan dari semua video yang diunggah. Metode PrintAllVideoPlaycount() mencetak daftar video milik pengguna dengan maksimal 8 video untuk menghindari tampilan berlebihan, sehingga tetap terorganisir dan mudah dibaca. Dengan prinsip Design by Contract, kelas ini memastikan bahwa semua operasi dilakukan dengan aman dan sesuai dengan batasan yang telah ditentukan.

1. Program.cs

using System;

class Program

{

static void Main()

{

SayaTubeUser user = new SayaTubeUser("Zhafir Zaidan Avail");

for (int i = 1; i <= 10; i++)

{

SayaTubeVideo video = new SayaTubeVideo($"Review Film {i} oleh Zhafir Zaidan Avail");

user.AddVideo(video);

}

user.PrintAllVideoPlaycount();

// Agar terminal tidak langsung tertutup

Console.WriteLine("\nPress any key to exit...");

Console.ReadKey();

}

}

Penjelasan:

Kelas Program berisi metode Main() yang berfungsi sebagai titik awal eksekusi program. Di dalamnya, dibuat objek SayaTubeUser dengan nama "Zhafir Zaidan Avail". Kemudian, menggunakan loop for, program membuat 10 objek SayaTubeVideo dengan judul "Review Film X oleh Zhafir Zaidan Avail" dan menambahkannya ke dalam daftar video milik pengguna. Setelah itu, metode PrintAllVideoPlaycount() dipanggil untuk menampilkan daftar video yang telah diunggah, dengan batas maksimal 8 video yang ditampilkan. Terakhir, program menambahkan Console.ReadKey() untuk memastikan terminal tidak langsung tertutup setelah eksekusi, menunggu pengguna menekan tombol sebelum keluar.