Name: Aryan Patel

SIN:- 301226774

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Week\_02\_lab\_05\_06

{

public enum MedalColor

{

Bronze,

Silver,

Gold

}

class Program

{

// Medal class driver function

public static void medalcontroller()

{

TextWriter writer = new StreamWriter("Medal.txt");

// create a medal object

Medal m1 = new Medal("Horace Gwynne", "Boxing", MedalColor.Gold, 2012, true);

//print the object

Console.WriteLine(m1.ToString());

//print only the name of the medal holder

Console.WriteLine("\nName: " + m1.Name);

//create another object

Medal m2 = new Medal("Michael Phelps", "Swimming", MedalColor.Gold, 2012, false);

//print the updated m2

Console.WriteLine(m2.ToString());

//create a list to store the medal objects

List<Medal> medals = new List<Medal>() { m1, m2 };

medals.Add(new Medal("Ryan Cochrane", "Swimming", MedalColor.Silver, 2012, false)); // Create and add object in madal list

medals.Add(new Medal("Adam van Koeverden", "Canoeing", MedalColor.Silver, 2012, false)); // Create and add object in madal list

medals.Add(new Medal("Rosie MacLennan", "Gymnastics", MedalColor.Gold, 2012, false)); // Create and add object in madal list

medals.Add(new Medal("Christine Girard", "Weightlifting", MedalColor.Bronze, 2012, false)); // Create and add object in madal list

medals.Add(new Medal("Charles Hamelin", "Short Track", MedalColor.Gold, 2014, true)); // Create and add object in madal list

medals.Add(new Medal("Alexandre Bilodeau", "Freestyle skiing", MedalColor.Gold, 2012, true)); // Create and add object in madal list

medals.Add(new Medal("Jennifer Jones", "Curling", MedalColor.Gold, 2014, false)); // Create and add object in madal list

medals.Add(new Medal("Charle Cournoyer", "Short Track", MedalColor.Bronze, 2014, false)); // Create and add object in madal list

medals.Add(new Medal("Mark McMorris", "Snowboarding", MedalColor.Bronze, 2014, false)); // Create and add object in madal list

medals.Add(new Medal("Sidney Crosby ", "Ice Hockey", MedalColor.Gold, 2014, false)); // Create and add object in madal list

medals.Add(new Medal("Brad Jacobs", "Curling", MedalColor.Gold, 2014, false)); // Create and add object in madal list

medals.Add(new Medal("Ryan Fry", "Curling", MedalColor.Gold, 2014, false)); // Create and add object in madal list

medals.Add(new Medal("Antoine Valois-Fortier", "Judo", MedalColor.Bronze, 2012, false)); // Create and add object in madal list

medals.Add(new Medal("Brent Hayden", "Swimming", MedalColor.Bronze, 2012, false)); // Create and add object in madal list

//prints a numbered list of 16 medals.

Console.WriteLine("\n\nAll 16 medals");

foreach (Medal m in medals)

{

Console.WriteLine(m.ToString());

}

//prints a numbered list of 16 names (ONLY)

Console.WriteLine("\n\nAll 16 names");

foreach (Medal m in medals)

{

Console.WriteLine("\nName: " + m.Name);

}

//prints a numbered list of 9 gold medals

Console.WriteLine("\n\nAll 9 gold medals");

foreach (Medal m in medals)

{

if (m.Color == MedalColor.Gold)

{

Console.WriteLine(m.ToString());

}

}

//prints a numbered list of 9 medals in 2012

Console.WriteLine("\n\nAll 9 medals on 2012");

foreach (Medal m in medals)

{

if (m.Year == 2012)

{

Console.WriteLine(m.ToString());

}

}

//prints a numbered list of 4 gold medals in 2012

Console.WriteLine("\n\nAll 4 gold medals in 2012");

foreach (Medal m in medals)

{

if (m.Year == 2012 && m.Color == MedalColor.Gold)

{

Console.WriteLine(m.ToString());

}

}

//prints a numbered list of 3 world record medals

Console.WriteLine("\n\nAll 3 world record medals");

foreach (Medal m in medals)

{

if (m.IsRecord)

{

Console.WriteLine(m.ToString());

}

}

//saving all the medal to file Medals.txt

Console.WriteLine("\n\nSaving to file");

foreach (Medal m in medals)

{

writer.WriteLine(m.ToString());

}

writer.Close();

}

// pet class driver function

public static void petController()

{

Pet pet1 = new Pet("Bruno", 1, "black husky dog"); // Create pet object

Pet pet2 = new Pet("Mars", 2, "black doberman dog"); // Create pet object

Pet pet3 = new Pet("Pluto", 3, "black labrador dog"); // Create pet object

Pet pet4 = new Pet("Jecky", 4, "black husky dog"); // Create pet object

List<Pet> petArray = new List<Pet>() { pet1, pet2, pet3, pet4 }; // List of Pet Object

petArray[1].SetOwner("Aryan"); // set Property

petArray[1].Train(); // set Property

petArray[2].SetOwner("Aryan"); // set Property

petArray[3].SetOwner("Mike"); // set Property

petArray[0].Train(); // set Property

// print all pet Objects

foreach (Pet p in petArray)

{

Console.WriteLine(p.ToString());

}

Console.WriteLine("write owner name\n");

string ownerName = Console.ReadLine();

var results = petArray.Where(p => p.Owner.ToLower() == ownerName.ToLower()); // filter all object according to give owner name

// print all pets of given owner name

if (results.Count() > 0)

{

Console.WriteLine($"All {ownerName}'s pet list are below\n ");

foreach (Pet p in results)

{

Console.WriteLine(p);

}

}

else

{

Console.WriteLine("this owner doesn't own any pets");

}

}

// main function

static void Main(string[] args)

{

petController(); // pet driver call

medalcontroller(); // medal driver call

}

// pet class

public class Pet

{

public string Name { get; } // field

public string Owner { get; private set; } // field

public string Description { get; } // field

public int Age { get; } // field

public bool InHouseTrained { get; private set; } // field

// Constructor

public Pet(string name, int age, string description)

{

this.Name = name;

this.Age = age;

this.Description = description;

this.Owner = "no one";

this.InHouseTrained = false;

}

// set owner name

public void SetOwner(string owner)

{

this.Owner = owner;

}

// set IsTrained property

public void Train()

{

this.InHouseTrained = true;

}

// override Tostring()

public override string ToString()

{

return $"Pet Object\n\t Name:- {this.Name}\n\t Age:-{this.Age}\n\t Description:- {this.Description}" +

$"\n\t Owner:- {this.Owner}\n\t Trained:- {this.InHouseTrained}";

}

}

// Medal Class

public class Medal

{

public string Name { get; } // field

public string TheEvent { get; } // field

public MedalColor Color { get; } // field

public int Year { get; } // field

public bool IsRecord { get; } // field

// Constructor

public Medal(string name, string theEvent, MedalColor color, int year,bool isRecord)

{

this.Name = name;

this.TheEvent = theEvent;

this.Color = color;

this.Year = year;

this.IsRecord = isRecord;

}

// override Tostring()

public override string ToString()

{

var Record = this.IsRecord ? "(R)" : "";

return $"{this.Year} - {this.TheEvent}{Record} {this.Name}({this.Color})";

}

}

}

}