**Phase-End Project**

**Player and Team Project**

**Source code:**

**Program.cs:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Project

{

internal class Program

{

static void Main(string[] args)

{

string c;

Player play = new Player();

play.M4();

OneDayTeam team = new OneDayTeam();

OneDayTeam.addplayer\_1 c1 = new OneDayTeam.addplayer\_1(team.Add);

do

{

Console.WriteLine("Enter your choice:\n 1.To Add Player \n 2.To Remove Player by Id \n 3.Get Player By Id \n 4.Get Player by Name \n 5.Get All Players ");

int n1 = Convert.ToInt32(Console.ReadLine());

switch (n1)

{

case 1:

c1(play);

// team.Add(play);

break;

case 2:

Console.WriteLine("Enter the Player id : ");

int id1 = Convert.ToInt32(Console.ReadLine());

team.Remove(id1);

break;

case 3:

Console.WriteLine("Enter the Player id : ");

int id = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("----------------------------------");

team.GetPlayerById(id);

break;

case 4:

Console.WriteLine("Enter the name: ");

string s = Console.ReadLine();

Console.WriteLine("----------------------------------");

team.GetPlayerByName(s);

break;

case 5:

team.GetAllPlayers();

break;

}

Console.WriteLine("Do you want to continue press yes/no..");

c = Console.ReadLine();

} while (c == "yes" );

Console.ReadLine();

}

}

}

**Player.cs:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Project

{

public class Player

{

private int playerid;

public int Player\_ID

{

get { return playerid; }

set { playerid = value; }

}

private string playername;

public string Player\_Name

{

get { return playername; }

set { playername = value; }

}

private int age;

public int PlayerAge

{

get { return age; }

set { age = value; }

}

public List<Player> prod

{

get;

set;

}

}

public static class Extension2

{

public static void M4(this Player ss)

{

Console.WriteLine("Player and Team Requirements");

Console.WriteLine("-----------------------------");

}

}

}

**OneDayTeam.cs:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Net.Http.Headers;

using System.Text;

using System.Threading.Tasks;

namespace Project

{

internal class OneDayTeam : Player,ITeam

{

public static List<Player> oneDayTeam = new List<Player>();

Player p = new Player();

public static int Capacity = 11;

public delegate void addplayer\_1(Player y);

public OneDayTeam()

{

Console.WriteLine("Team Capacity: {0}", OneDayTeam.Capacity);

Console.WriteLine("-----------------------------");

}

public void Add(Player player)

{

if (oneDayTeam.Count < Capacity)

{

Player p = new Player();

Console.WriteLine("Enter id:");

p.Player\_ID = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Enter Name:");

p.Player\_Name = Console.ReadLine();

Console.WriteLine("Enter age:");

p.PlayerAge = Convert.ToInt32(Console.ReadLine());

oneDayTeam.Add(p);

player.prod = oneDayTeam;

Console.WriteLine("Added successfully!!");

Console.WriteLine("------------------------------");

// return player;

}

else

{

Console.WriteLine("Player cannot be added !! already 11 members are there");

Console.WriteLine("----------------------------------");

}

}

public List<Player> GetAllPlayers()

{

Console.WriteLine("List of members:");

Console.WriteLine("----------------------------------");

foreach (var item in oneDayTeam)

{

Console.WriteLine("Player id:" + item.Player\_ID);

Console.WriteLine("Player name:" + item.Player\_Name);

Console.WriteLine("Age:" + item.PlayerAge);

Console.WriteLine("----------------------------------");

}

return oneDayTeam;

}

public Player GetPlayerById(int playerId)

{

var ti = oneDayTeam.Where(mi => mi.Player\_ID == playerId).ToList();

if (ti.Count > 0)

{

foreach (var item in ti)

{

Console.WriteLine("Player id:" + item.Player\_ID);

Console.WriteLine("Name:" + item.Player\_Name);

Console.WriteLine("Age:" + item.PlayerAge);

Console.WriteLine("----------------------------------");

}

return p;

}

else

{

Console.WriteLine($"Id {playerId} not found !!");

// return ti[0];

Console.WriteLine("----------------------------------");

return p;

}

}

public Player GetPlayerByName(string playerName)

{

var ti = oneDayTeam.Where(mi => mi.Player\_Name == playerName).ToList();

if (ti.Count > 0)

{

foreach (var item in ti)

{

Console.WriteLine("Player id:" + item.Player\_ID);

Console.WriteLine("Name:" + item.Player\_Name);

Console.WriteLine("Age:" + item.PlayerAge);

Console.WriteLine("----------------------------------");

}

}

else

{

Console.WriteLine($"Player name {playerName} not found !!");

}

return p;

}

public void Remove(int playerId)

{

//oneDayTeam = oneDayTeam.Where(m1 => m1.Player\_ID != playerId).ToList();

var p1 = oneDayTeam.Find(item => item.Player\_ID == playerId);

if (p1 != null)

{

oneDayTeam.Remove(p1);

Console.WriteLine($"Player ID {playerId} Removed Successfully!!");

}

else

{

Console.WriteLine($"Player ID {playerId} Not found !");

}

Console.WriteLine("----------------------------------");

}

}

}

**ITeam.cs:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Project

{

internal interface ITeam

{

void Add(Player player);

void Remove(int playerId);

Player GetPlayerById(int playerId);

Player GetPlayerByName(string playerName);

List<Player> GetAllPlayers();

//void GetAllPlayers(List<Player> oneDayTeam);

}

}