MAYANK KAUSHAL

EMP ID=2579352

Github link= https://github.com/mayank5654/dotnetphase-end-project1

Dot Net Phase 1: Player and Team Project

SOURCE CODE

PROGRAM.CS

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace FastPace_Cricket_Academy
{
    internal class Program
    {
        static void Main(string[] args)
        {

            TeamManager teamManager = new TeamManager();
            bool shouldContinue = true;
            const int AddPlayerOption = 1;
            const int RemovePlayerOption = 2;
            const int GetPlayerByIdOption = 3;
            const int GetPlayerByNameOption = 4;
            const int DisplayPlayerOption = 5;


            while (shouldContinue)
            {
                Console.WriteLine("1:To Add Player \n2:To Remove Player by Id 
\n3.Get Player By Id \n4.Get Player by Name \n5.Get All Players:\n");
                Console.Write("Enter your Choice: ");
                int choice = teamManager.GetValidInt();
                switch (choice)
                {
                    case AddPlayerOption:
                        teamManager.AddPlayer();
                        break;

                    case RemovePlayerOption:
                        teamManager.RemovePlayer();
                        break;

                    case GetPlayerByIdOption:
                        teamManager.GetPlayerById();
                        break;

                    case GetPlayerByNameOption:
                        teamManager.GetPlayerByName();
                        break;
                case DisplayPlayerOption:
                    teamManager.DisplayAllPlayers();
                    break;

                default:
                    Console.WriteLine("Invalid Entry!!");
                    break;
            }
            Console.Write("Do you want to continue? (yes/no): ");
            string userResponse = Console.ReadLine().ToLower();
            Console.WriteLine();

            if (userResponse != "yes")
            {
                shouldContinue = false;
            }
        }
    }
}

class TeamManager
{
    private OneDayTeam team = new OneDayTeam();

    public void AddPlayer()
    {
        if (OneDayTeam.oneDayTeam.Count >= team.Capacity)
        {
            Console.WriteLine("Team is full!");
            return;
        }

        Console.Write("Enter Player ID: ");
        int id = GetValidInt();
        Console.Write("Enter Player Name: ");
        string name = GetValidString();
        Console.Write("Enter Player Age: ");
        int age = GetValidInt();

        Player player = new Player();
        player.PlayerID = id;
        player.PlayerName = name;
        player.PlayerAge = age;

        team.Add(player);
        Console.WriteLine("Player added successfully");
    }

    public void RemovePlayer()
    {
        Console.Write("Enter Player ID to Remove: ");
        int id = GetValidInt();
        team.Remove(id);
    }

    public void GetPlayerById()
    {
        Console.Write("Enter Player ID: ");
        int id = GetValidInt();

        Player player = team.GetPlayerById(id);
        if (player != null)
```

```
            {
                Console.WriteLine(player.PlayerID + "    " + player.PlayerName + "
" + player.PlayerAge);
            }
            else
            {
                Console.WriteLine("Player not found!");
            }
        }

        public void GetPlayerByName()
        {
            Console.Write("Enter Player Name: ");
            string name = GetValidString();

            Player player = team.GetPlayerByName(name);

            if (player != null)
            {
                Console.WriteLine(player.PlayerID + "    " + player.PlayerName + "
" + player.PlayerAge);
            }
            else
            {
                Console.WriteLine("Player not found!");
            }
        }

        public void DisplayAllPlayers()
        {
            List<Player> players = team.GetAllPlayers();
            if (players.Count > 0)
            {
                foreach (var player in players)
                {
                    Console.WriteLine(player.PlayerID + "    " + player.PlayerName
+ "    " + player.PlayerAge);
                }
            }
            else
            {
                Console.WriteLine("No players in team! Add some players...");
            }

        }

        public int GetValidInt()
        {
            int value;
            bool intNum;
            do
            {
                intNum = int.TryParse(Console.ReadLine(), out value);
                if (!intNum)
                {
                    Console.Write("Enter an Integer value! --> ");
                }
            } while (!intNum);
            return value;
        }

        public string GetValidString()
        {
            string input;
            do
            {
                input = Console.ReadLine();
                if (string.IsNullOrWhiteSpace(input))
                {
                    Console.Write("Enter a valid string value! --> ");
                }
            } while (string.IsNullOrWhiteSpace(input));
            return input;
        }

    }

}
```

PLAYER.CS

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace FastPace_Cricket_Academy
{
    internal class Player
    {
        public int PlayerID { get; set; }
        public string PlayerName { get; set; }
        public int PlayerAge { get; set; }
    }
}
```

ONE DAY TEAM.CS

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace FastPace_Cricket_Academy
{
    internal class OneDayTeam
    {
        public static List<Player> oneDayTeam = new List<Player>();

        public int Capacity { get; }

        public OneDayTeam()
        {
            Capacity = 11;
        }

        public void Add(Player player)
        {
            oneDayTeam.Add(player);
        }

        public void Remove(int playerId)
        {
            if (oneDayTeam.Exists(player => player.PlayerID == playerId))
            {
                oneDayTeam.RemoveAll(player => player.PlayerID == playerId);
                Console.WriteLine("Player removed successfully");
            }
            else
            {
                Console.WriteLine("Player not found in the list!");
            }
        }

        public Player GetPlayerById(int playerId)
        {
            return oneDayTeam.Find(player => player.PlayerID == playerId);
        }

        public Player GetPlayerByName(string playerName)
        {
            return oneDayTeam.Find(player => player.PlayerName == playerName);
        }

        public List<Player> GetAllPlayers()
        {
            return oneDayTeam;
        }
    }
}
```

ITEAM.CS

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace FastPace_Cricket_Academy
{
    internal interface ITeam
    {
        void Add(Player player);
        void Remove(int pLayerId);
        Player GetPlayerById(int playerId);
        Player GetPlayerByName(string playerName);
        List<Player> GetAllPlayers();
    }
}
```