

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.ComponentModel.Design;
using System.Globalization;
using System.Net.Http.Headers;
using System.Numerics;
using System.Text.RegularExpressions;
using System.Threading.Channels;
using System.Xml.Linq;
using Microsoft.VisualBasic;
using static System.Runtime.InteropServices.JavaScript.JSType;
using System.Net;
using System.Net.Mail;

namespace KinoGame
{

    public class UserAccount
    {
        public string FirstName { get; set; }
        public string LastName { get; set; } // Changed to string for LastName
        public int Age { get; set; }
        public DateTime BirthDay { get; set; }
        public string Mail { get; set; }
        public string ID { get; set; }

        static List<Dictionary<string, object>> userAccounts = new List<Dictionary<string, object>>();

        public UserAccount(string FirstName, string LastName, int Age, DateTime BirthDay, string
Mail, string ID)
        {
            this.FirstName = FirstName;
            this.LastName = LastName;
            this.Age = Age;
            this.BirthDay = BirthDay;
            this.Mail = Mail;
            this.ID = ID;
        }

        public static void MessagesPersonalInformationInput()
        {
            Console.WriteLine("\n Welcome");
            Console.WriteLine("\n You need to register so you can play the game.");
            Console.WriteLine("\n Fill in your personal information:");
        }

        public static void PersonalInformationInput()
        {
            string FirstName = string.Empty;
            string LastName = string.Empty;
            string AgeOfuser = string.Empty;

```

```

DateTime BirthDay = DateTime.MinValue;
string Mail = string.Empty;
string ID = null;

while (true)
{
    Console.WriteLine("\n Write your First Name:");
    FirstName = Console.ReadLine();

    if (string.IsNullOrEmpty(FirstName) || FirstName.Any(char.IsDigit) ||
FirstName.Any(char.IsSymbol) || FirstName.Any(char.IsNumber))
    {
        Console.WriteLine("You have inserted invalid characters. Please give a valid Name.");
        continue;
    }
    break;
}

while (true)
{
    Console.WriteLine("\n Write your Last Name:");
    LastName = Console.ReadLine();

    if (string.IsNullOrEmpty(LastName) || LastName.Any(char.IsDigit) ||
LastName.Any(char.IsSymbol) || LastName.Any(char.IsNumber))
    {
        Console.WriteLine("You have inserted invalid characters. Please give a valid Last
Name.");
        continue;
    }
    break;
}

while (true)
{
    Console.WriteLine("\n Write your Age:");
    AgeOfuser = Console.ReadLine();

    if (string.IsNullOrEmpty(AgeOfuser) || !int.TryParse(AgeOfuser, out int age) || age <
0)
    {
        Console.WriteLine("Please give a valid age.");
        continue;
    }

    if (age < 18)
    {
        Console.WriteLine("You cannot register, because you are under age.");
        Console.WriteLine("We must close the application.");
        Environment.Exit(0);
    }
    break;
}

```

```

    }

    while (true)
    {
        Console.WriteLine("\n Write your Birth Day (dd-mm-yyyy):");

        if (!DateTime.TryParse(Console.ReadLine(), out BirthDay))
        {
            Console.WriteLine("Please give a valid date.");
            continue;
        }
        break;
    }

    while (true)
    {
        Console.WriteLine("\n Write your Mail:");
        Mail = Console.ReadLine();
        string emailPattern = @"^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9-
9.-]+\.(com|gr|net|org|edu|gov|mil|biz|info|mobi|name|aero|asia|jobs|museum|eu)$";
        if (string.IsNullOrEmpty(Mail) || !Mail.Contains("@"))

        {
            Console.WriteLine("Please give a valid email address.");
            continue;
        }

        if (Regex.IsMatch(Mail, emailPattern))
        {
            Console.WriteLine(Mail);
        }

        break;
    }

    while (true)
    {
        Console.WriteLine("\n Write your ID:");
        if (string.IsNullOrEmpty(ID) || string.IsNullOrEmpty(ID))
        {
            break;
        }
        Console.WriteLine("Please give a valid ID.");
        continue;
    }

    var userRegistry = new Dictionary<string, object>
    {
        { "FirstName", FirstName },
    }

```

```

        { "LastName", LastName },
        { "Age", int.Parse(AgeOfuser) },
        { "BirthDay", BirthDay },
        { "Mail", Mail },
        { "ID", ID }
    };

    userAccounts.Add(userRegistry);

    Console.WriteLine("\n Your registration is completed successfully.");

    foreach (var user in userAccounts)
    {
        Console.WriteLine($"First Name: {user["FirstName"]}, Last Name: {user["LastName"]},
Age: {user["Age"]}, Birth Day: {user["BirthDay"]}, Mail: {user["Mail"]}, ID: {user["ID"]}");
    }
}
}
}

```