

Prueba de Prácticas de Fundamentos de Programación

I. Introducción

La práctica consiste en la programación de un Instagram personal. Para ello tenemos 2 clases (Usuario e Instagram) y un programa principal que utilizará la red social.

II. Definición de las Clases

a) Estructura de la clase Usuario

La clase Usuario se utiliza para almacenar la información de un seguidor en tu perfil de Instagram. Dicha información es el username, password, bio y DatosPersonales.

La estructura de dicha clase es la siguiente:

```
struct DatosPersonales {
    cad name, lastName, email;
    int phone;
}

class Usuario {
    cad username [15];
    cad password [30];
    cad bio [50];
    bool reportado;
    DatosPersonales misDatos;
public:
    Usuario(cad usr[15], cad pass[30]);
    void Get_Username(cad usr[15]);
    void Set_Username(cad usr[15]);
    void Get_Password(cad pass[15]);
    void Set_Password(cad pass[15]);
    void Get_Bio(cad bi[15]);
    void Set_Bio(cad bi[15]);
    bool Get_Reported();
    void Set_Reportado(bool rpt);
    DatosPersonales Get_DatosPersonales();
    void Set_DatosPersonales(DatosPersonales dp);
};

void MostrarUsuario (Usuario u);
Usuario crearPerfil();
```

La descripción de las funciones y los métodos es la siguiente:

Usuario::Usuario(cad usr[15], cad pass[30]);

Método constructor que inicializa los atributos de la clase al valor proporcionado en los parámetros, siempre y cuando cada uno de los valores no estén vacíos. Por defecto al crearse una cuenta en Instagram y tener 0 followers, el usuario no puede estar reportado.

```
void Usuario::Get_Username(cad usr[15]);
void Usuario::Get_Password(cad pass[15]);
void Usuario::Get_Bio(cad bi[15]);
bool Usuario::Get_Reported();
DatosPersonales Get_DatosPersonales();
```

Métodos que devuelven por parámetro respectivamente el username, el password, la bio, y con return los datos personales.

```
void Usuario::Set_Username(cad usr[15]);
void Usuario::Set_Password(cad pass[15]);
void Usuario::Set_Bio(cad bi[15]);
void Usuario::Set_Reported(bool rpt);
void Usuario::Set_DatosPersonales(DatosPersonales dp);
```

Métodos para asignar el username, el password, la bio y los datos personales.

```
void MostrarUsuario (Usuario u);
```

Función genérica que mostrará por pantalla cada uno de los atributos del usuario pasado por parámetro.

```
Usuario crearPerfil();
```

Función genérica que pedirá por teclado tanto el username como el password. Esta función devuelve el objeto de la clase Usuario en cuyos atributos se han asignado la información introducida por teclado.

b) Estructura de la clase Instagram

La clase Instagram permitirá al usuario tener hasta 50 followers (al ser usuario inicial en la plataforma) y el propio Influencer que subirá contenido audiovisual en la red social de moda. Instagram permitirá poder añadir, borrar y buscar followers.

La estructura de la clase Instagram es la siguiente:

```
class Instagram {
    Usuario influencer;
    Usuario followers[50];
    int NFollowers;
public:
    Instagram();
    bool AddFollower(Usuario follower);
    bool RemoveFollower(Usuario follower);
    int SearchFollower(Usuario follower);
    void MostrarFollower(Usuario follower);
    void MostrarListaFollowers();
    int NumFollowers();
};
```

La descripción de las funciones y los métodos es la siguiente:

```
Instagram();
```

Método constructor que debe inicializar los datos de la red social. El método debe solicitar los datos del influencer para poder acceder a la plataforma.

bool AddFollower(Usuario follower);

Método que hará que un follower pasado por parámetro siga al influencer y que se añada a la lista de followers, siempre y cuando haya espacio. El método devuelve true si ha podido seguir el follower correctamente al influencer, y false en caso contrario.

Para que un follower pueda seguir a un influencer, éste no puede estar previamente reportado.

bool RemoveFollower(Usuario follower);

Método que hará que un follower pasado por parámetro deje de seguir al influencer y se borre de la lista de followers. Se comprobará previamente si el usuario pasado por parámetro es verdaderamente un follower del influencer, y de ser así, podrá dejar de seguirlo. Si realmente lo deja de seguir, el método devuelve true, false en el caso de que no exista el follower o que no se haya podido realizar correctamente la operación.

Si el follower que va a dejar de seguir al influencer es el único seguidor del influencer, se deberá de mostrar un mensaje pidiendo disculpas por el retraso, y hacer que su cuenta sea reportada.

int SearchFollower(Usuario follower);

Método que buscará el follower pasado por parámetro en la lista de followers. Devolverá la posición en la lista del elemento que coincide o bien -1 si no existe.

void MostrarFollower(Usuario follower);

Método que mostrará el perfil de un follower siempre y cuando el follower pasado por parámetro exista. En caso contrario no mostrará nada.

void MostrarListaFollowers();

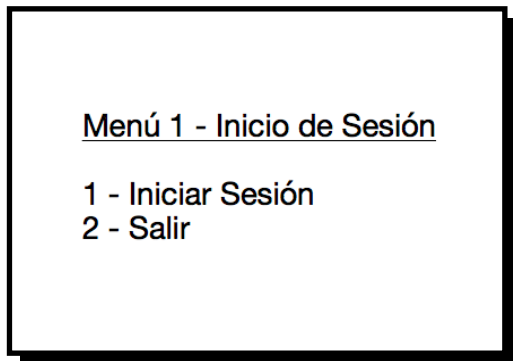
Método que mostrará por pantalla los datos del perfil del influencer y una lista de los followers que consumen su contenido. En caso de no tener followers, se le recordará que su cuenta o bien es recién creada o que está reportada.

int NumFollowers();

Método que devolverá el número actual de followers del influencer.

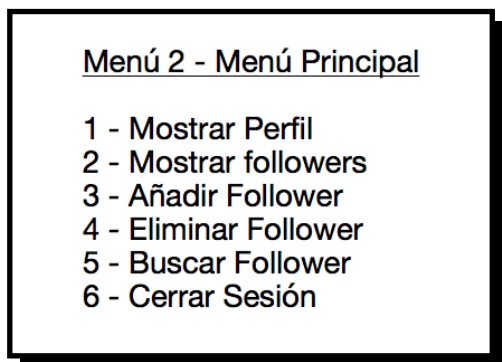
III. Programa Principal

La aplicación de Instagram está compuesta por 2 menús cuya estructura y descripción es la siguiente:



“Iniciar Sesión”: Esta opción pedirá por teclado tanto el username como el password para poder crear de manera temporal el influencer.

“Salir”: Esta opción hace que la aplicación se cierre.



“Cerrar Sesión”: Esta opción hará mostrar el menú 1 para poder volver de nuevo a iniciar sesión o salir de la aplicación.