

Examen Diciembre 2016

Programación 2

Duración 2hrs.

Pregunta 1 (45 ptos)

Una cadena de hoteles necesita un programa capaz de gestionar sus sucursales y huéspedes.

De las sucursales sabemos su nombre, dirección, país y los huéspedes que se hospedan allí. De los huéspedes sabemos su identificación, nombre completo, nro. de habitación y cantidad de días que se va a hospedar inicialmente. Por ahora existen dos tipos de huéspedes: aquellos que son VIP y aquellos que no. A los huéspedes VIP se les regala días de cortesía bajo la siguiente reglas:

- Nros. de Habitaciones entre 1 y 10, se les regala 1 día.
- Nros. de Habitaciones entre 11 y 50, se les regala 2 días.
- Nros. de Habitaciones mayores a 50 se les regala 3 días.

Realiza un **modelo** y un **programa** capaz de:

- Agregar una sucursal a la cadena de hoteles.
- Crear huéspedes y hospedarlos en sucursales. No es posible agregar un huésped ya existente (dado por su identificación).
- Sacar un huésped de una sucursal (check out). No es posible sacar un huésped no existente (dado por su identificación).
- Reporte que indique, dada una sucursal, cantidad de huéspedes hospedados.

Ejemplifica el modelo con un programa principal.

Pregunta 2 (30 ptos)

```
class GooglePlus {
    public String GenerarToken (String usuario) { /*...*/ }
    public boolean ValidarCredenciales (String key) { /*...*/ }
    public void LogIn(String usuario, String clave) { /*...*/ }
}

class Twitter {
    public boolean ValidarMail(String usuario) { /*...*/ }
    public void LogIn(String usuario, String clave) { /*...*/ }
}

class SocialNetLogIn {
    GooglePlus googlePlus = new GooglePlus();
    Twitter twitter = new Twitter();
    public void Loggears (String usuario, String clave, int cual) {
        String token = googlePlus.GenerarToken(usuario);
        if ((cual == 0) && googlePlus.ValidarCredenciales (token))
            googlePlus.LogIn(usuario, clave);
        if ((cual == 1) && twitter.ValidarMail(usuario))
            twitter.LogIn(usuario, clave);
    }
}
```

Realiza los cambios en el código anterior para que la clase SocialNetLogIn cumpla con OCP.

Pregunta 3 (25 ptos)

```
class UCUAntiVirusAPI {
    public bool ContieneVirus (Algoritmo algoritmo, Programa programa) {
        if (algoritmo.ContieneVirus(programa) == true)
            MandarACuarentena(programa);
        return false;
    }
    else
        return true;
}
private void MandarACuarentena (Programa programa) { ... }
public void MostrarReporte() { ... }
}

class Program {
    public static void Main (String[] args) {
        UCUAntiVirusAPI uAPI = new UCUAntiVirusAPI();
        Algoritmo a = new Algoritmo();
        Programa p = new Programa();
        if (uAPI.ContieneVirus (a, p))
            Console.WriteLine("VIRRUSSSSS")
        else
            Console.WriteLine("Todo OK con el programa")
        uAPI.MostrarReporte();
    }
}
```

Agrega una excepción llamada "VirusExcepcion", que se lance al momento de verificar si contiene un virus o no un programa y que sea atrapada por el Main. Recordar que siempre se debe mostrar el reporte (mediante el método MostrarReporte()).