

Examen Julio 2016

Programación 2

Duración 2hrs.

Pregunta 1 (45 pts)

Una cadena de supermercados necesita un programa capaz de gestionar sus sucursales y productos. De las sucursales sabemos su nombre, dirección y los productos que existen allí (en su stock). De los productos sabemos su código, descripción y precio.

Existen dos tipos de productos: aquellos que tienen una fecha de vencimiento y aquellos que no.

Además, existen productos que son combos. Esto quiere decir que un producto puede estar compuesto de uno o más productos (por ejemplo el pack "Festichola" compuesto de papas y maníes). El precio final de este tipo de productos es la sumatoria de los precios de sus subproductos aplicando un descuento dado al total de la sumatoria.

Realiza un programa capaz de

- Agregar una sucursal a la cadena de supermercados.
- Crear un producto y almacenarlo en una sucursal. No es posible agregar un producto ya existente (dado por su código).
- Vender de una sucursal un producto (sacarlo del stock). Cabe aclarar que no es posible vender inexistente (dado por su código) ni tampoco un producto ya vencido.
- Llevar el total de ventas efectuadas en cada sucursal.

Ejemplifica el modelo con un programa principal.

Pregunta 2 (30 pts)

```
class YouTube {
    public void VerVideo(String unVideo) { /*...*/ }
}

class Vimeo {
    public void ComenzarVideo(unVideo) { /*...*/ }
    public void IngresarUsuario() { /*...*/ }
}

class StreamingVideo {
    YouTube youTube = new YouTube();
    Vimeo vimeo = new Vimeo();

    public void Play (String unVideo, int cual) {
        if (cual == 0)
            youTube.VerVideo(unVideo)
        if (cual == 1) {
            vimeo.IngresarUsuario();
            vimeo.ComenzarVideo(unVideo);
        }
    }
}
```

Realiza los cambios en el código anterior para poder soportar varios tipos de streaming de video (como VeVo). Los cambios realizados deberán considerar que, agregar un streaming nuevo al programa, **no** implique la modificación de la clase StreamingVideo.

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 excepcion 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()).