

PROGRAMACIÓN ORIENTADA A OBJETOS – EXAMEN PRÁCTICO 15 ENERO 2016.

PROBLEMA 1 – 5 PUNTOS

Para la implementación de un Centro de Atención al Usuario necesitamos una clase que actúe como <<buffer>> de entrada para las solicitudes. Tras el correspondiente análisis se ha llegado a las siguientes conclusiones.

- Todos los elementos a insertar en el buffer implementan una interfaz entrada cuya única característica es una función que devuelve un entero, cero para los de prioridad alta, uno para los de prioridad media y dos para los de prioridad baja.
- Se utilizarán <<buffer>> de varios tipos de elementos, pero un buffer siempre almacena elementos del mismo tipo (llamadas, correos, sms)
- El buffer tiene que permitir la inserción de nuevos elementos de su tipo (mete), informarnos sobre cuál será el primero en ser atendido (primero) o eliminar el primer elemento (saca)
- El criterio para atender a los elementos del buffer es: Primero atendemos a los de mayor prioridad y, de entre ellos, al primero por orden de entrada.

Elabore una clase Java que implemente un <<buffer>>. Se valorará la documentación del código y especialmente la utilización de pre y post condiciones, descritos por asertos convenientemente documentados o contratos escritos mediante Cofaja.

PROBLEMA 2 – 5 PUNTOS

La empresa <<FutbolCable>> se dedica a vender partidos de fútbol en <<streaming>>. El sistema de facturación de la empresa utiliza una clase Contrato de la que nos han proporcionado la siguiente información.

/*Implementa la facturación de un contrato. Se omiten elementos de la implementación.

```
public class Contrato {
```

/*Inicializa un nuevo contrato para el cliente c

```
public Contrato (Cliente c)
```

/*Añade el partido p al conjunto de partidos a facturar en este periodo

```
public void compra (Partido p)
```

/*Devuelve el importe total de la factura

- El importe incluye tanto determinados gastos fijos como el importe de los partidos comprados
- El importe de un partido depende de varios criterios, entre ellos el equipo local y el visitante

```
public double importe()
```

Reinicia la información sobre los partidos consumidos para iniciar un nuevo periodo de facturación

```
public void reset()
```

Sobre la clase Partido sólo sabemos que dispone de métodos getLocal() y getVisitante() que devuelven un Equipo.

De cara a la campaña navideña, la empresa ha decidido tener nuevos tipos de contratos con condiciones especiales:

Seguidor: Dirigida a los seguidores de un equipo, fijado al crear el contrato. No se cobra ningún partido en que el equipo seguido actúe como visitante. Además el primer partido de cada periodo no se factura. El resto de los partidos y gastos fijos se facturan normalmente.

DosPorUno: Para aumentar el consumo de partidos. Sólo se cobra 1 de cada 2 partidos en cada periodo de facturación. El precio del resto de la factura (incluidos los partidos facturados) se incrementa en un 5%

TodoElFutbol: Pensado para grandes consumidores. No se factura ningún partido, pero los costes fijos se incrementan en un 20%

Elaborar el diagrama de clases de la solución del problema e implementar en Java las clases necesarias para resolver las nuevas necesidades del sistema, considerando que no podemos modificar las clases ya existentes, ni realizar suposiciones sobre la forma en que han sido implementadas.