

## EXAMEN PROGRAMACION 2 – II PARCIAL – 24 MAYO 2011

Para una compañía de celular:

**Crear** una interfaz llamada IPlan que tiene funciones como ser: (10%)

- double getPrecio( int cantmins,int cantmsgs). Retorna el monto a pagar final del plan.
- Atributos String TIPO\_40 y TIPO\_80 para identificar cada plan.

**SUPONER:** Clase PlanAbstracto que contiene dos atributos double: precioxmin y precioxmsg. Ambos son protected y se inicializan en el constructor de la clase.

- 1- Crear las clases Plan40 y Plan80 (10% c/u) que hereda de PlanAbstracto e implementa IPlan. Cada plan cobra según su tipo asignado es decir los de tipo 40 pagan \$40 fijos mensuales. El cobro de cada plan es diferente para cada clase:
  - a. Plan40 se le permite hablar hasta 160 mins gratis al mes, los minutos restantes se cobra según el precioxmin definido. El plan tiene no tiene msgs gratis al mes. Además se le rebaja el 5% del subtotal final si el cliente hablo menos de 30 minutos en el mes.
  - b. Plan80 permite hablar hasta un total de 300 mins gratis al mes, lo restante se cobra según el precioxmin, lo mismo sucede con los msgitos que tiene máximo 150, la diferencia se cobra. Además este plan tiene la posibilidad de optar por un plan ilimitado de internet (atributo de la clase y se inicializa por un parámetro en el constructor). Si este plan esta activado se cobra 15 USD adicionales.
- 2- Crear la clase Cliente que tiene como atributos públicos el nombre y el número telefónico, además de un atributo privado de tipo **PlanAbstracto**. Tiene Funciones como **void setPlan(PlanAbstracto plan)** que inicializa el atributo de tipo PlanAbstracto que posee además dispara una excepción TipoInvalidoException (SUPONGA QUE YA EXISTE y se hereda de **Exception**) si el parámetro enviado no es de de tipo Plan40 ni Plan80, si se dispara el error la función lo propaga el mismo hacia arriba. Los demás valores se le asignan desde el constructor de la clase. (20%)
- 3- Crear una clase llamada UNICELL que tiene un atributo final estático llamado MAX\_CLIENTS (2%), además un atributo llamado contador que me dice cuantos clientes están registrados actualmente, en el constructor se inicializa en cero. La clase tiene un arreglo de tipo Cliente como atributo también que es de una longitud de acuerdo a MAX\_CLIENTS y se inicializa en el constructor de la clase (3%). Y tiene las siguientes funciones:
  - a. Void AdicionCliente(Cliente c ). Trata de guardar el Cliente c dentro del arreglo siempre y cuando haya espacio en el mismo. Si hay espacio se inicializa el plan para ese cliente. Se valida con un Try Catch que aun haya espacio en el arreglo.(15%)
  - b. boolean setPlan(String nombre). Busca el cliente con el nombre según el parámetro, para asignarle el tipo de plan que va tener dicho cliente. La función retorna true si lo pudo cambiar y false si no.(15%)
  - c. double PagoTotalMes( ). Recorre el arreglo asignando la cantidad de minutos y mensajitos que consumieron **los clientes registrados** y llevando un acumulado final del monto total a cobrar por la empresa para retornarlo.(15%)

BONO. 2 oro (10%) = Hacer la función que me imprima toda la información de los clientes, con la ayuda de un foreach. Void printClientes(String tipo), si el tipo es de TIPO\_40 se imprima solos los que tienen el plan 40 y si es del TIPO\_80 solo se imprimirá los del 80, si es otro no se imprimirá nada.