PROGRAMACIÓN ORIENTADA A OBJETOS **02/12/2005**

NOMBRE:_____Titulación:_____

1. El Balneario de Archena ofrece diferentes servicios a sus clientes: Lodo, Masaje, … cada uno con un precio. Además ofrecen la posibilidad de contratar programas. Un programa es un tipo de servicio que incluye otros servicios (por ejemplo, el programa denominado descanso puede incluir un masaje y un lodo). En el caso de los programas el precio se calcula como la suma de los servicios que incluye descontando sobre el total un 10%.

a) (**1 pto**) Especifica la jerarquía de clases en un lenguaje con herencia múltiple como Eiffel que represente el problema propuesto e implementa el método `getPrecio` en la clase que represente los programas.

b) (**1 pto**) Especifica cual sería la solución en Java (con herencia simple) para el mismo problema e implementa también el método `getPrecio` que calcule el precio de los programas.

2. (**1 pto**) Sea **`servicios`** la variable que hace referencia a la lista que contiene todos los servicios ofertados por el Balneario de Archena. Implemente tanto en Java como en Eiffel un método **`getPromocion`** que devuelva el programa ofertado más barato

3. Dada definición de la clase Eiffel LINEAR_ITERATOR vista en clase:

a) (**0'5 ptos**) Define el iterador ITERADOR_SERVICIOS encargado de sacar un listado de todos los servicios que ofrece el Balneario.

b) (**0'5 ptos**) Explica la diferencia entre la implementación de los iteradores propuesta para el lenguaje Eiffel y los iteradores que ofrece el lenguaje Java.

c) (**0'5 ptos**) ¿Es un ejemplo del el patrón de diseño del Método Plantilla? Justifica la respuesta.

4. Sea la clase `ControladorTanque` la clase encargada de interactuar con el software de la tarjeta hardware que abre y cierra la válvula para la entrada y salida de líquido de un tanque de agua. El software de la tarjeta lanza una excepción cuando encuentra algún error al abrir o cerrar la válvula (por ejemplo, si está atascada). La especificación parcial de esta clases sería:

```
class ControladorTanque feature
     valvula: TarjetaValvula;
     capacidad: REAL;
     volumenActual: REAL;

     quitarAgua (cantidad: REAL) is
     require
              no_vacio: volumenActual >0
              hay_suficiente: cantidad <= volumenActual
     do
       valvula.open(cantidad)
       volumenActual := volumenActual – cantidad;
     ensure
       volumenActual = old volumenActual - cantidad
     end

     …
}
```

a) (**0'5 ptos**) Utiliza el método **`quitarAgua`** para explicar la técnica del Diseño por Contrato propuesta por Bertrand Meyer, especificando las obligaciones y beneficios de cada una de las partes implicadas en el contrato.

b) (**1 pto**) Supuesta la implementación de la clase **`ControladorTanque`** en Java ¿De qué tipo es la excepción que lanza el método **`open`**, comprobada o runtime? Implemente el método **`quitarAgua`** en Java explicando TODAS las decisiones de diseño tomadas.

5. (**1 pto**) Vamos a definir una clase `Arbol` reutilizando las clases `Nodo` y `Lista` que ya tenemos implementadas haciendo uso de la herencia múltiple. Un objeto árbol **es un** `Nodo` con una referencia a su hermano y **es una** `Lista` enlazada de sus hijos. Tanto la clase `Nodo` como la clase `Lista` incluyen un método `add`, que debemos conservar en la implementación de la clase `Arbol`. El significado del método `add` de `Nodo` en la clase `Arbol` es añadir un hermano, mientras que el método `add` definido en la clase `Lista` tiene el significado de añadir un hijo. Dado que necesitamos mantener las dos versiones en la clase Arbol. Explica la solución al conflicto de nombres que se ocasiona en C++ y compara la solución dada con la manera de resolverlo en el lenguaje Eiffel.

6. a) (**1 pto**) Dada la implementación parcial de la clase `Bola` en Eiffel que representa una bola en un juego de billar, especificar una clase Java equivalente, explicando TODAS las decisiones y los problemas encontrados.

```
class Bola feature
      direccion: Real  --en radianes
      energia: Real
feature {NONE}
      calculaCentro: Punto is do …end
feature {MesaBillar}
      region:Rectangulo
…
end
```

b) (**0'5 ptos**) Explica si es posible exportar de manera selectiva el atributo `region` a la clase `MesaBillar` en el lenguaje C++.

7. Sea la clase Java `TorreControl` la encargada de mandar avisos a los aviones, personal de vuelo, … en definitiva a todo el que se registre como interesado en recibir las notificaciones de la torre. Por tanto, una implementación parcial de esta clase podría ser:

```
class TorreControl {
      private LinkedList oyentes;
      …
      public void notificarMet1(){
            Iterator it = oyentes.iterator();
            while (it.hasNext())
                  it.next().met1();
      }

      public void notificarMet2(){
            Iterator it = oyentes.iterator();
            while (it.hasNext())
                  it.next().met2();
      }
}
```

a) (**0'5 ptos**) ¿Es correcto el código anterior? En el caso de que no lo sea explica la manera de solucionarlo.

b) (**1 pto**) Factoriza el comportamiento común de los métodos `notificarMet1` y `notificarMet2` de manera que sólo haya un método `notificar` al que se le pase como parámetro el método que debe notificar.