ESCUELA SUPERIOR POLITECNICA DEL LITORAL
FACULTAD DE INGENIERIA EN ELECTRICIDAD Y COMPUTACION
EXAMEN PARCIAL - PROGRAMACION ORIENTADA A OBJETOS
I Término 2011, Julio 2011

Nombre:_____ **Paralelo:**_____
Matricula:_____

TEMA 1. (20 puntos) Se desea diseñar el sistema para la Copa América Argentina 2011. En este campeonato se encuentra organizados en grupos, cada grupo consta de varios equipos de fútbol, los mismos que representan a cada uno de los países participantes. El campeonato se desarrolla a través de partidos de fútbol los que se juegan en los estadios designados en las fechas establecidas acorde al cronograma de partidos, como resultado de los partidos se puede identificar si hubo un ganador o si fue un empate, así como la cantidad de puntos que gana cada equipo. También se desea mantener la información de los Jugadores de cada equipo (incluyendo cantidad de goles) y Directores Técnicos.

Defina el diagrama de clases UML para el problema presentado, mostrando:
a) Las clases requeridas para el sistema
b) Las relaciones entre las clases
c) Multiplicidad, atributos

TEMA 2. (15 puntos)
Considerando el modelo del tema 1, implementar

a. Las clases que representan a los grupos, equipos y jugadores.
b. Completar la clase CopaAmerica

```
public class CopaAmerica
{
        public ArrayList<Grupo> grupos;

        public CopaAmerica(){
                grupos = Campeonato.cargarGruposCopa();
        }

        public static void main(String args[]){
                CopaAmerica ca = new CopaAmerica();

                // Mostrar el listado de grupos ordenados alfabeticamente y por cada
                // grupo mostrar los equipos ordenados por puntos obtenidos









                // Por cada equipo mostrar el listado de jugadores ordenados de menor a
                // mayor por número de camiseta









        }
```

Elaborado por: Jorge Rodríguez E., Javier Tibau, Gonzalo Luzardo

TEMA 3. (15 puntos)

1. ¿Qué imprime por pantalla el siguiente código?

```
class X{
        int xpto(){return 5;}
}
class Y extends X{
        int xpto(){return 10;}

        void test(){
                X x = (X) this;
                System.out.print(this.xpto());
                System.out.print(x.xpto());
                System.out.print(((X)this).xpto());
                System.out.print(super.xpto());
        }

        public static void main(String[] args){
                new Y().test();
        }
}
```

a) 5 5 5 10
b) 10 5 5 10
c) 10 10 5 5
d) 10 10 10 5
e) Ninguna de las anteriores

2. Asuma que las clases A y B del código adjunto están definidas en el mismo paquete. ¿Cuál es el valor de los atributos de B luego de la sentencia "new B(10)"?

```
public   class    A {
          String str;

          public A() {
                  str="Xpto";
          }

          public A(String str) {
                  this.str=str;
          }
}
```

```
public class B extends A {
          int num;

          public B() {
                  super("Hello");
          }
          public B(int num) {
                  num++;
          }
}
```

a) str="Xpto" num=0
b) str="Xpto" num=11
c) str="Hello" num=0
d) str="Hello" num=1
e) Ninguna de las anteriores

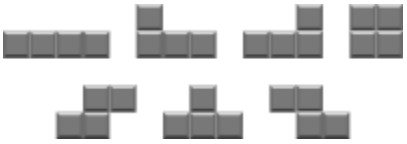
3. ¿Qué imprime por pantalla el siguiente código?

```
Integer i = new Integer(2);
Integer j = new Integer(2);
System.out.print(i==j);
System.out.print(i.equals(j));
```

a) true true
b) true false
c) false true
d) false false
e) Ninguna de las anteriores

Elaborado por: Jorge Rodríguez E., Javier Tibau, Gonzalo Luzardo

TEMA 4. (40 puntos) Usted es parte del equipo de desarrollo de una empresa dedicada a la implementación de Juegos de Video y participará en la implementación del juego **"TetrisPOO"**. Su líder de proyecto le indica las reglas básicas del juego. TetrisPOO es un juego compuesto por 7 tetrinos, figuras geométricas compuestas por cuatro bloques cuadrados unidos de forma ortogonal.



Existen 7 clases de tetrinos (piezas) **"I", "J", "L", "O", "S", "T", "Z"**. Todos los tetrinos tienen un color, ubicación y un indicador que establece si está en movimiento o no. Adicionalmente, tienen el ángulo de rotación, excepto la **"O"**.

Los tetrinos pueden rotar y moverse en tres posibles direcciones (abajo, izquierda, derecha). Además, cuenta con la interfaz **MovimientoTetrino**, la misma que se define de la siguiente forma:

```
public interface MovimientoTetrino {
void rotar(int grados);
void mover (Direccion d);
}
```

Una vez que le han explicado la metodología del juego, usted deberá implementar:

a) La superclase Tetrino. Considere que de esta clase no se pueden crear instancias.
b) La enumeración Direccion
c) La clase TetrisPoo que tiene como constantes el ancho (15) y alto (30) del tablero sobre el que se mueven los tetrinos.
d) Las clases derivadas que representen a las piezas O y Z
e) Constructores, use this y/o super adecuadamente. Por defecto: color → blanco,ubicación → (0,0), detenido → false
f) Implementar la interface MovimientoTetrino (validar que la nueva posición este dentro del tablero)
g) Cuando se requiera imprimir un tetrino se imprimirá en el siguiente formato: "<Tetrino>: {color = c, Ubicación=(X, Y), En movimiento=m}", donde c es el color del Tetrino, X es la coordenada x en el TetrisPoo y Y es la coordenada y, y m indica si se está moviendo o no. Adicionalmente, se imprime la figura. Por ejemplo si el Tetrino es una Z, la impresión por pantalla será de la siguiente forma:

```
Z: {color=rojo, Ubicación =(1,1), En moviento = Si}
00
 00
```

h) Realice la implementación considerando que las siguientes sentencias puedan compilar:

```
TetrinoO o;
TetrinoZ z;

ArrayList<Tetrino> tetrinos = new ArrayList<Tetrino>();
o = new TetrinoO("Cyan"); //color, ubicación y movimiento por defecto
tetrinos.add(o);
z = new TetrinoZ(1,1); // x, y, color por defecto y movimiento por defecto
tetrinos.add(o);
o = new TetrinoO(true); // moviento, por defecto x, y, color
tetrinos.add(o);
z = new TetrinoZ(verde,1,1, true); // x, y, color por defecto y movimiento por defecto
```

i) Complete la siguiente sentencia en el main:

```
for(Object o: tetrinos){
// imprimir solamente los tetrinos Z




}
```

TEMA 5. (10 puntos) Una función es una abstracción computacional matemática que toma una entrada, ejecuta algún cálculo y produce una salida. En la teoría de ciencias computacionales, una función es llamada "lambda." En POO, podemos modelar la abstracción de funciones a través de una interface llamada Lambda, que define un método **"apply"** que recibe un objeto x como entrada y devuelve un objeto.
Definir:

1. La interface Lambda
2. Implementar dos clases Lambda1 y Lambda2 que implementen Lambda

Elaborado por: Jorge Rodríguez E., Javier Tibau, Gonzalo Luzardo