

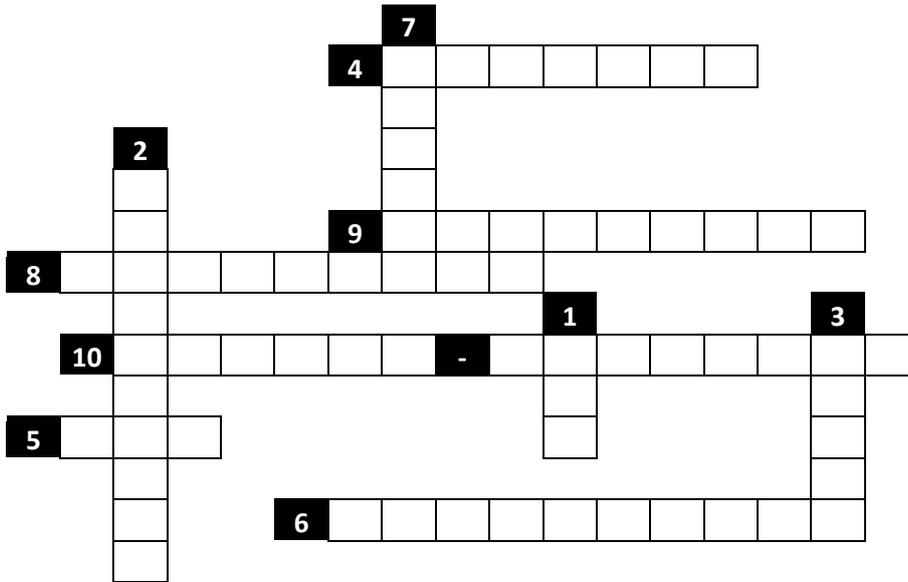
ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN
PROGRAMACION ORIENTADA A OBJETOS
PRIMERA EVALUACIÓN -2011-11-30

Nombre: _____ Matrícula _____ Paralelo _____

TEMA 1. (10 puntos)

Complete el siguiente crucigrama de conceptos orientados a objetos:



Verticales

1. Tipo de dato primitivo
2. Métodos con el mismo nombre
3. No es un método de instancia
7. Nivel de acceso

Horizontales

4. Agrupa clases, enumerados, etc.
5. Ejecutables de java
6. Orden natural
8. Clase incompleta
9. Asegura comportamiento
10. Habilidad para hacer referencia a objetos de diferentes tipos y ejecutar métodos apropiados

TEMA 2. (20 puntos)

El API de Java define la interfaz Comparable, donde `x.compareTo(y)` retorna -1 si `x` es menor que `y`, 0 si son iguales y 1 si no lo son.

```
interface Comparable {  
    int compareTo(Object);  
}
```

- a) Defina la clase Onda que implementa Comparable, la misma que se comportan como onda en el espacio. Una onda se crea conociendo su amplitud y su longitud de onda. La comparación entre ondas de su sistema se realiza en base a su amplitud. Por ejemplo:

```
ArrayList<Onda> ondas = new ArrayList<Onda>(); // Crea una lista  
ondas.add(new Onda(10)); // Agrega una onda  
ondas.add(new Onda(5)); // Agrega una onda  
  
Onda o1 = new Onda(6);  
Onda o2 = new Onda(8);  
  
ondas.add(o1);  
ondas.add(o2);  
  
System.out.println(o1.compareTo(o2)); // Imprime -1 por pantalla
```

- b) Escriba un bloque de código que muestre ordenados por pantalla todos los objetos de la lista ondas.

TEMA 3. (20 puntos)

Acorde a la definición del proyecto parcial:

- Grafique su diagrama de clases
- Escriba el método **boolean colocarFicha(Ficha f, int x, int y)**, el mismo que intenta colocar una ficha f en una coordenada (x,y) de su tablero (asuma que este es un atributo de la clase donde está definiendo el método) de SudokuPOO. Si la ficha puede ser colocada, se la coloca dentro del tablero y retorna true, caso contrario retorna false.

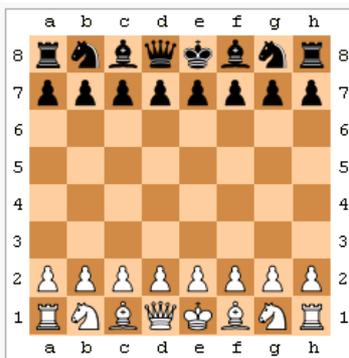
Nota:

- La clase ficha contiene el método getNumero() que le retorna el número que contiene la ficha (número que se quiere colocar).
- Si desea puede crear más métodos adicionales para modularizar el proceso de colocar una ficha.
- Si utiliza variables de instancia que usted ha definido en su proyecto las deberá especificar con un comentario.

Importante: El literal b será considerado como su sustentación de proyecto, es decir que la nota obtenida en este literal será ponderada al factor de calificación de sustentación.

TEMA 4. (50 puntos)

Usted es parte del grupo de Video-juegos de la Facultad y participará en la implementación del juego "Ajedrez-POO". Su líder de proyecto le indica las reglas básicas del juego.



El ajedrez se juega entre dos jugadores; cada uno posee 16 trebejos o piezas, siendo las de un jugador de color claro, llamadas blancas, y las de su oponente de color oscuro, llamadas negras. Los trebejos se mueven sobre un tablero de ajedrez cuadrado de $8 \times 8 = 64$ casillas, con los mismos colores que los trebejos colocados alternativamente, 32 claras y 32 oscuras, también llamadas escaques.

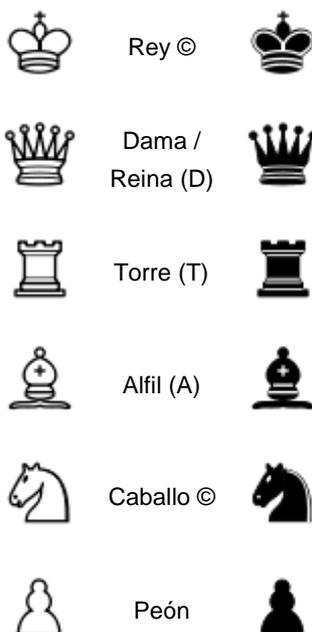
Los trebejos de cada jugador al principio de la partida son: Rey, Reina, Alfil (2), Caballo (2), Torre (2), Peon (8).

Cada tipo de trebejo se puede mover de una forma diferente, lo que determinará su potencia y su importancia en el desarrollo del juego

Los Caballos en particular se mueven avanzando 2 casillas en vertical y una en horizontal, o viceversa, realizando un movimiento de L, siendo el único trebejo que puede saltar por encima de los demás trebejos.

Usted cuenta con las siguientes clases que ya han sido implementadas:

Trebejos



```
class Celda {
    private int fila;
    private char columna;
    private Trebejo trebejo;
    private Color c;

    public Celda(int fil, char col, Color color){
        fila = fil;
        columna = col;
        trebejo = null;
        c = color;
    }

    void setTrebejo(Trebejo f){ trebejo = t; }
    Trebejo getTrebejo () { return trebejo; }
}
```

<pre> public class Tablero { private ArrayList<Celda> celdas; private Jugador j1, j2; public Tablero(){ int i = 0, j = 0; celdas = new ArrayList<Celda>(); j1 = new Jugador(); inicializaTrebejos(j1, Color.BLANCO); setTrebejosEnCeldas(j1); j2 = new Jugador(); inicializaTrebejos(j2, Color.NEGRO); setTrebejosEnCeldas(j2); for(i=0;i<8;i++) for (j=0;j<8;j++) celdas.add(new Celda(i+1, (char)(97+j))); } private void setTrebejosEnCeldas(Jugador j){ for(Trebejo f : j.getTrebejos()) getCelda(f.getFila(), f.getColumna()).setTrebejo(f); } </pre>	<pre> private Celda getCelda(int fila, char columna) { /*metodo ya implementado*/ } private void inicializaTrebejos(Jugador j, Color c){ int i = 0, fila = 0; fila = c == Color.BLANCO ? 1 : 8; j.addTrebejo(new Rey(fila,'e', c)); j.addTrebejo(new Reina(fila,'d', c)); j.addTrebejo(new Alfil(fila,'c', c)); j.addTrebejo(new Alfil(fila,'f', c)); j.addTrebejo(new Caballo(fila,'b', c)); j.addTrebejo(new Caballo(fila,'g', c)); j.addTrebejo(new Torre(fila,'a', c)); j.addTrebejo(new Torre(fila,'h', c)); fila = c == Color.BLANCO ? 2 : 7; for (i=0;i<8;i++) j.addTrebejo(new Peon(fila, (char)(97+i), c)); } } </pre>
---	---

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

- a) **(7 pts.)** La superclase Trebejo, la cual es una clase que no se puede instanciar; como atributos contiene la fila y columna en la que se encuentra colocada en el tablero y el color. Adicionalmente, está compuesta de los prototipos de los métodos que deberán implementar las clases derivadas de esta.
- b) **(7 pts.)** La clase Jugador que tendrá como variables de instancia la lista de trebejos que actualmente tiene y la lista de trebejos que ha ganado de su contrincante. No olvide crear el método para inicializar las instancias de esta clase, así como los métodos para agregar trebejos a la lista de trebejos del jugador, para perder un trebejo y para agregar un trebejo a la lista de trebejos ganados.
- c) **(4 pts.)** El enumerado Color que contenga los posibles colores de los trebejos.
- d) **(15 pts.)** La clase derivada que representen al Caballo (incluya constructores, use this y/o super adecuadamente). Realice la implementación considerando que las siguientes sentencias puedan compilar:

```

Caballo c1, c2, c3, c4;

c1 = new Caballo(1,'b', Color.BLANCO); // x, y
c2 = new Caballo(8,'b'); // x, y, color por defecto negro

c3 = new Caballo(1, Color.BLANCO); // fila, columna por defecto = 'g', color
c4 = new Caballo(8); // fila, columna por defecto = 'g', color por defecto negro

```

- e) **(7 pts.)** El método **ArrayList<Celda> getPossiblePosicionFinal(Trebejo t)** de la clase Tablero para el caso en que el trebejo es un Caballo; este método debe devolver un listado de Celdas en las que se podría colocar el trebejo dependiendo del tipo de movimiento que esta tenga. Un trebejo se puede mover de una celda a otra siempre y cuando la celda este vacía o contenga un trebejo del otro jugador. Considere los métodos implementados en la clase Tablero.
- f) **(10 pts.)** El método **void moverTrebejo(Trebejo f, Celda cInicial, Celda cFinal)** el mismo que mueve un trebejo de una celda a otra e imprime por pantalla la jugada realizada.
 - Una jugada se escribe con la inicial del trebejo y la casilla a la cual se mueve, excepto por el peón, del que se menciona sólo la casilla (Ce4, por ejemplo, significa que un caballo mueve a la casilla e4; d5 representa la jugada de un peón de la columna d a la quinta fila).
 - Una captura se indica con una "x" entre la inicial del trebejo (o la columna del peón) y la casilla del trebejo capturado (Cxb5 indica que un caballo captura un trebejo en b5; dxex6 indica que el peón de la columna d captura en e6).