

Auxiliar Extra Control 2 – Profesor Nelson Baloian
Jueves 30 de abril de 2009

Auxiliares: Óscar Álvarez, Roberto Vargas

1. (Pregunta 1 Control 2 2002)

Para mantener una pieza de Ajedrez en el tablero de 8 filas y 8 columnas, se dispone de la clase:

```
class Pieza{
    protected int fila, columna;
    public Pieza( int x,int y){ fila=x; columna=y; }
    public boolean mover(int x,int y){fila=x; columna=y; return true;}
    public int obtenerFila(){return fila;}
    public int obtenerColumna(){return columna;}
}
```

A) (ponderación: 2/3) Escriba las clases Caballo y Torre que extiendan la clase Pieza con los métodos:

ejemplo	significado
C=new Caballo(1,1)	constructor que ubica un caballo en fila 1 y columna 1
C.mover(2,3) C.mover(1,2)	Mueve C a fila 2 y col. 3, y entrega true (movimiento válido) Entrega false y no mueve el caballo (movimiento inválido). Nota. Un movimiento de un caballo es válido si avanza 2 filas y 1 columna o 2 columnas y 1 fila dentro del tablero.
T = new Torre(5,6)	constructor que ubica una torre en fila 5 y columna 6
T.mover(5,8) T.mover(5,9)	Mueve T a fila 5 y col 8, y entrega true (movimiento válido). Entrega false y no mueve la torre (movimiento inválido). Nota. Es válido si avanza en la misma fila o columna dentro del tablero.

B) (ponderación: 1/3) Escriba el método **boolean comer(Pieza X,Pieza Y)** que devuelva true si la Pieza X “come” en una jugada a la Pieza Y, considerando que no hay más piezas en el tablero. Por ejemplo, si **Caballo C=new Caballo(1,2)** y **Torre T=new Torre(3,1)**, entonces **comer(C,T)** devuelve true y **comer(T,C)** devuelve false.

2. (Pregunta 1 Control 2 2006)

Considerando la siguiente interfaz:

```
interface I{
    public void leer(String x)
    public String escribir();
    public int perimetro();
}
```

a) Escriba las clases Rectangulo y Triangulo que implementen la interfaz I para representar, calcular el perímetro, leer y escribir las figuras correspondientes.

Notas:

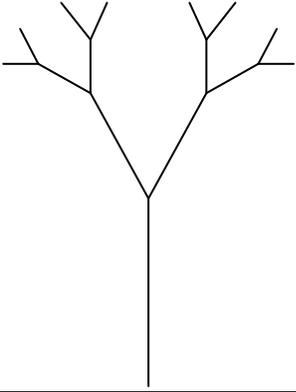
- Las figuras se representan por las longitudes de sus lados (2 valores enteros para rectángulos y 3 valores para triángulos)
- El método leer, debe obtener desde el String los valores (separados por espacios) de los lados de la figura (2 ó 3 valores enteros según corresponda)
- El método escribir debe entregar un String con los valores de los lados separados por espacios (2 ó 3 valores enteros)

b) Escriba un programa que lea el archivo “figuras.txt” que en cada línea contiene una letra (R o T) y a continuación 2 ó 3 números (valores de los lados separados por espacios). El programa debe usar las clases anteriores para escribir como resultado los calores de los lados de la figura de mayor perímetro. Por ejemplo, si el archivo contiene las líneas:

```
R 10 20
T 30 40 5
R 20 30
T 1 2 3
```

entonces el resultado es 20 30, puesto que 100 es el mayor perímetro y corresponde al rectángulo de lados 20 y 30.

3. (Pregunta 2 Control 2 2006)

Profundidad	10
Ángulo (grados)	30
	
Dibujar	Cerrar

La figura muestra una interfaz gráfica que permite dibujar árboles. El dibujo de un árbol está compuesto de una rama inicial (tronco) que se divide en dos ramas separadas por un ángulo y que miden $\frac{2}{3}$ de la longitud de la rama inicial.

Escriba un programa que implementa la interfaz ilustrada. Los dos campos de texto (objetos de tipo TextField) permiten determinar los parámetros del dibujo: la profundidad de ramas en que se abre el árbol y el ángulo en que se abren las ramas.

Notas:

- El botón etiquetado “Dibujar” dibuja el árbol, teniendo en cuenta los dos parámetros indicados en los campos de texto (que tienen valores iniciales 10 y 30)
- Utilice un BorderLayout, con un Canvas al centro y paneles al norte (2 filas y dos columnas) y sur (1 fila y dos columnas)
- El botón cerrar termina la ejecución del programa
- Se recomienda usar y escribir un método recursivo de encabezamiento:

```
void dibujarArbol(int profundidad, int largo, double angulo, int x, int y, Graphics g)
//coordenadas iniciales (horizontal y vertical) del tronco en el objeto g.
```