

Programación Orientada a Objetos

Práctica 1

Inicio: 10 de Febrero

Entrega: 16 de Marzo a las 23:55 p.m.

El objetivo de esta práctica es familiarizarse con los conceptos básicos de orientación a objetos en el lenguaje Java (clases, objetos y encapsulación), así como practicar el manejo de arrays.

Nota: Puedes descargar una instalación conjunta de NetBeans 8 y JDK 8 en:

<http://www.oracle.com/technetwork/java/javase/downloads/jdk-netbeans-jsp-142931.html>

Se pide crear el juego de las damas, para lo que será necesario trabajar con una matriz de dos dimensiones sobre la que se irán añadiendo casillas blancas y negras de manera alternada. No podrá haber dos casillas con el mismo color ni horizontal ni verticalmente. Tras inicializar el tablero, se colocarán en él varias damas de forma aleatoria, y se intentarán algunos movimientos. Se trata de comprobar si dichos movimientos son posibles, y en caso afirmativo, efectuarlos. Para lograr completar la práctica, se deberán realizar los apartados que se detallan a continuación.

Apartado 1: La clase Casilla (2 puntos)

La clase `Casilla` servirá para crear el tablero de juego con las casillas de color blanco o negro de manera alternada. Su código (incompleto) es el siguiente:

```
public class Casilla {
    static final char BLANCO = 'B';
    static final char NEGRO = 'N';
    private char color;
}
```

Extiende la clase `Casilla` con métodos que permitan obtener la siguiente funcionalidad:

- Consultar el color de una casilla
- Cambiar el color de una casilla
- Imprimir por pantalla el color de una casilla
- Comparar una casilla con otra, teniendo en cuenta que dos casillas serán iguales si su color coincide

Construye un `main` que cree una casilla blanca y una casilla negra, y las muestre por pantalla si su color es distinto.

Apartado 2: La clase Dama (1 punto)

Crea la clase `Dama`, que representará una pieza dentro del tablero. Las damas pueden ser de color blanco o negro.

Apartado 3: La clase Tablero (4 puntos)

La clase `Tablero` será la clase que permitirá crear un tablero bien formado, permitiendo únicamente la ubicación de casillas de color alterno y correcto. Un tablero será siempre cuadrado, y su dimensión se especificará al construirlo. Construye la clase `Tablero` teniendo en cuenta lo siguiente:

- Deberá existir un método `rellenar()` que, partiendo del cálculo de un color al azar para la primera casilla, rellene el tablero con casillas de forma correcta, es decir, con colores alternos. No deberá quedar ninguna celda de la matriz sin casilla asignada. Puedes utilizar la clase `java.util.Random` para generar un número aleatorio en función del cual se elegirá el color de la casilla inicial.

- El constructor recibirá la dimensión del tablero (que permite crear la matriz cuadrada), el número de damas blancas que se colocarán en el tablero, y el número de damas negras que se colocarán en el tablero. El constructor colocará las damas aleatoriamente en el tablero, teniendo en cuenta que las damas sólo pueden ocupar casillas negras, y que una casilla sólo puede estar ocupada a la vez por una dama. Puedes utilizar la clase `java.util.Random` para generar posiciones aleatorias.
- Deberá existir un método `mostrar()` que mostrará por pantalla el tablero, la posición de las damas dentro del mismo, y si hay un ganador. Con una 'N' se indicará que una casilla vacía del tablero es de color negro; con una 'B' se indicará que una casilla vacía del tablero es de color blanco; con '[b]' se indicará que una casilla del tablero contiene una dama blanca; y con '[n]' se indicará que una casilla del tablero contiene una dama negra. El juego tiene un ganador si sólo hay damas de un color sobre el tablero, siendo dicho color el ganador. Un ejemplo de salida del método para un tablero de tamaño 5x5 con 3 damas es el siguiente:

```

Tablero:
    [b]  B   N   B   N
      B   N   B   N   B
      N   B  [b]  B  [b]
      B   N   B   N   B
      N   B   N   B   N
Damas:
    Dama blanca en (1, 1)
    Dama blanca en (3, 3)
    Dama blanca en (3, 5)
Ganador:
    Blancas

```

- Además de los métodos previos, deberá deducirse cualquier otro método que sea necesario para realizar estas tareas, incluyendo sus modificadores de acceso. Se deberá respetar el principio de encapsulamiento.

Apartado 4: Las reglas del juego (3 puntos)

Extiende la clase `Tablero` con un método `mover(int x1, int y1, int x2, int y2)` que mueva la dama que está en la posición (x1, y1) a la posición (x2, y2). El método devolverá un valor indicando si el movimiento se ha realizado o no. Para poder realizar el movimiento se tendrán que cumplir las siguientes reglas:

- Las posiciones inicial y final deben estar dentro de los límites del tablero.
- Debe haber una dama en la posición (x1, y1), y ninguna dama en la posición (x2, y2).
- El movimiento debe ser siempre en diagonal (ej. de la posición (1,2) a la posición (2,3)).
- Las damas blancas sólo pueden moverse hacia adelante, es decir, la y de la nueva posición debe ser mayor que la y de la posición actual. Por el contrario, las damas negras sólo pueden moverse hacia “atrás”.
- Una dama no puede saltar damas de su mismo color.
- Una dama puede saltar a lo sumo una dama de otro color, en cuyo caso, además de realizar el movimiento, se eliminará la dama del otro color del tablero.

Construye una clase `PruebaTablero` con un método `main` que cree un tablero de tamaño 8x8 con 5 damas de cada color, realice varios movimientos, e imprima el tablero tras cada movimiento. Además, elabora el diagrama de clases de tu aplicación, que deberás entregar junto con el código resultante en los distintos apartados.

Normas de Entrega

Los ficheros **Casilla.java**, **Dama.java**, **Tablero.java** y **PruebaTablero.java**, junto con el **diagrama de clases** de la aplicación, deben comprimirse en un único fichero zip/rar con nombre *pl-nia1-nia2.zip*, donde **nia1** y **nia2** serán los números de identificación de cada alumno de la pareja. La finalidad de las clases y métodos deberá estar documentada adecuadamente como comentarios javadoc en el código. Uno de los miembros de la pareja entregará el fichero a través de moodle antes de la fecha de entrega indicada en la cabecera del enunciado.