

Grados en Ingeniería Informática, del Software y de Computadores (Grupos C, D, E),
y Doble Grado de Informática y Matemáticas (Grupo A)

Tecnología de la Programación - Curso 2016/2017

Examen de Junio (06/06/2017) - Duración: 3 horas **Puntuación máxima: 10 puntos**

Instrucciones

- En el examen debes partir de la última práctica entregada, que se te proporciona al principio del examen.
- **Crea un fichero de texto cambiosExamen.txt en la raíz los fuentes de tu proyecto (dentro de src).** En este fichero deberás incluir tu **nombre completo** y el nombre de todos los ficheros .java que modifiques. Además puedes incluir otros comentarios relevantes sobre tu solución del examen. Por ejemplo, si sabes cómo hacer algo, pero te atascas al escribirlo, en lugar de entregar código que no funciona o que ni siquiera compila, coméntalo e indica en este fichero lo que querías hacer y dónde está el código comentado.
- El código entregado *debe compilar*, y debe evitar *romper la encapsulación de las clases* (acceso a atributos privados y protegidos desde clases externas, utilización de atributos públicos, etc.). Si el código no compila o rompe encapsulación de clases, tendrá la *calificación de 0 puntos*.
- En la corrección del examen se valorará el funcionamiento, la claridad del código, el uso conveniente de los medios proporcionados por la Programación Orientada a Objetos (herencia, polimorfismo y vinculación dinámica) y el buen uso de comentarios.

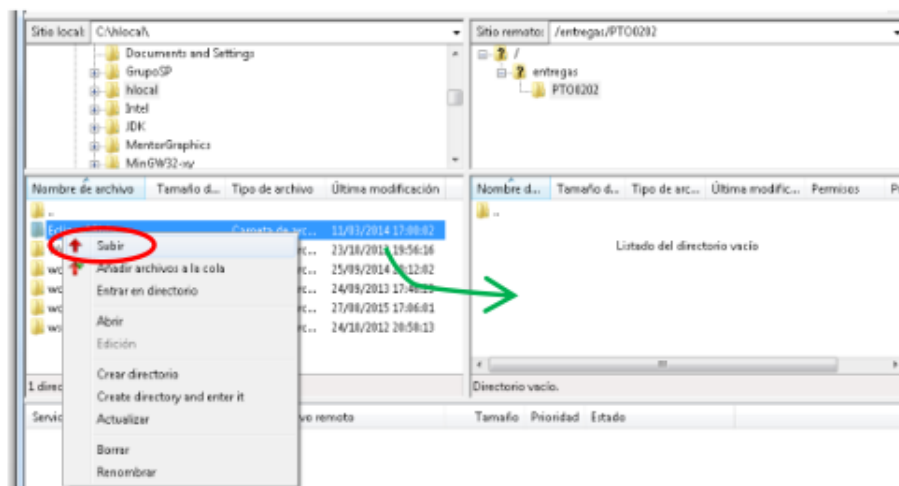
Consejos

- Realiza el examen en el orden en el que aparece en el enunciado. Comprueba periódicamente que la práctica sigue funcionando tras los cambios introducidos. Cada vez que realices un apartado, guarda una copia de la práctica por si necesitas volver a recuperarla.
- Incluye comentarios en cambiosExamen.txt donde expliques y justifiques muy brevemente las modificaciones que has introducido en tu código para cumplir con el enunciado del examen.

Instrucciones de entrega

- Para entregar la solución al examen, crea un fichero NombreApellido1Apellido2.zip. En él debes incluir todo el proyecto una vez limpiado de archivos intermedios (es decir, **sin** el directorio target).
- Haz doble clic en el icono del escritorio denominado: **“EXAMENES en LABs entregas...”**, y dentro de la ventana que aparece, doble clic en **“ALUMNOS entrega de practicas y examenes”**.

Se abre otra ventana en la que debes seleccionar el archivo zip en el panel inferior izquierdo y arrastrarlo al panel inferior derecho (o utiliza el botón derecho del ratón y la opción **Subir**), como se indica en la figura.

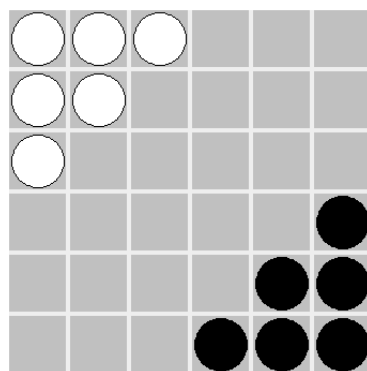


- Antes de abandonar el laboratorio debes pasar por el puesto del profesor para asegurarte de que lo que se ve en el puesto del profesor es lo que has entregado (*no cierres tu sesión hasta haberlo verificado*) y **firmar en la hoja de entregas**.

Enunciado

1. [4.5 puntos]

Se quiere añadir a la práctica 5 el juego *Damas Chinas* que es una variante simple en tablero cuadrado del juego del mismo nombre. Se juega entre dos jugadores (que juegan con fichas blancas y negras) y consiste en un tablero 6×6 en el que inicialmente están situadas 6 fichas de cada color, ocupando cada color una esquina del tablero, como se muestra en la siguiente figura:



Las reglas del juego son las siguientes:

- **Inicio del juego.** Las blancas son las primeras en jugar.
- **Movimientos.** En cada turno, el jugador puede mover una ficha de su color a una casilla adyacente libre en cualquiera de las 8 direcciones posibles (vertical, horizontal, diagonales). No se realizan capturas y no se cambia el color de las demás fichas del tablero.
- **Siguiente jugador.** Los dos jugadores se van alternando siempre que se pueda. Si un jugador no puede realizar ningún movimiento, se pasa el turno al oponente, quien por tanto realizará dos movimientos seguidos.
- **Fin del juego.** Se pueden producir el siguiente final de juego:
 - Un jugador gana si consigue situar todas sus fichas en la esquina opuesta del tablero (es decir, en las casillas que ocupaba el jugador contrario en la situación inicial de la partida).
 - En este juego no se puede producir la situación de empate.

Para realizar este ejercicio tendrás que:

- Crear clases específicas en el paquete `es.ucm.fdi.tp.damasChinas` dentro de tu práctica 5 con las clases necesarias para que se pueda jugar a este juego en modo consola, además de los juegos ya disponibles en la práctica. Estas clases deben sobreescribir el método `toString()` para poder ser usadas en *modo consola*.

- Modifica la clase `Main` para permitir lanzar una partida de damas chinas usando la opción “`damas`” desde la línea de comandos. También se debe modificar el método `usage()` para mostrar la información actualizada.
- Realiza los cambios necesarios para que se pueda jugar a este juego en el interfaz de ventana. Las casillas del tablero pueden tener colores alternos como en el ajedrez o un color uniforme.

Ten en cuenta que en este apartado **no está permitido cambiar ningún método del modelo** (`GameTable`) ni añadir métodos nuevos.

2. [3.5 puntos] Modifica el código de tu solución de la Práctica 5 para que muestre, además de las ventanas por cada jugador, una nueva ventana que contenga indicadores con el número de segundos que ha pasado pensando cada jugador desde que se inició la partida. Puedes calcular los tiempos usando `System.currentTimeMillis()`, que devuelve un número en milisegundos que representa el instante en el que se ejecuta la llamada a este método.

Para realizar este apartado debes **decidir con cuidado el diseño de tu programa**, puesto que esta decisión tendrá impacto en la calificación del apartado. La ventana creada **no debe ser accesible desde las demás ventanas del programa**.

3. [2 puntos] Añade a tu programa un componente gráfico, `Simulacion`, que extienda a `JPanel` y que contenga dos botones, `start` y `stop`. El botón `start` debe simular la ejecución de 10000 partidas de jugadores aleatorio-contra-aleatorio a partir del estado actual del juego al que se esté jugando. Cuando un jugador pulse este botón, se lanzará la simulación de forma que no se debe bloquear el funcionamiento de los demás componentes visuales. Además, las simulaciones se deberán ejecutar en paralelo, usando para ello 4 hilos de simulación (cada uno simulando 2500 partidas).

Al terminar la simulación se mostrará al usuario, utilizando el panel de mensajes, el número de victorias de cada jugador y el porcentaje de éxito del jugador actual. El botón `stop` debe cancelar la ejecución de la simulación.

Para realizar la simulación de **una partida** a partir de un estado puedes utilizar el método que se muestra a continuación:

```
public static <S extends GameState<S, A>, A extends GameAction<S, A>>
int randomGame(S initialState, int maxActions) {
    Random r = new Random();
    S currState = initialState;
    List<A> available = currState.validActions(currState.getTurn());
    for (int i=0; i<maxActions && !available.isEmpty()
        && !currState.isFinished(); i++) {
        A rndAction = available.get(r.nextInt(available.size()));
        currState = rndAction.applyTo(currState);
        available = currState.validActions(currState.getTurn());
    }
    return currState.getWinner();
}
```

Observa que en este método se debe indicar un número máximo de movimientos. Si la

partida no ha terminado en ese número de movimientos, se puede considerar que en ese caso ha terminado en tablas. Un buen valor de referencia para este número puede ser 100.