

PROYECTO DE PROGRAMACIÓN - PRIMER TRIMESTRE

Descripción del proyecto

En este proyecto tendrás que poner en práctica los conocimientos adquiridos durante el primer trimestre del curso, en las unidades 1, 2, 3 y 4. El proyecto consiste en la realización de un juego de tablero. La descripción completa y detallada de la implementación del proyecto se encuentra en la siguiente sección.

Debes aplicar todas las buenas prácticas aprendidas:

- Validar todos los datos de entrada, tanto los introducidos por teclado por el usuario, como los parámetros de las funciones.
- Evitar código duplicado y reutilizar código mediante funciones.
- Nombrar las variables, clases y métodos de acuerdo a las convenciones del lenguaje, y usar identificadores descriptivos y representativos de lo que almacena o realiza la variable/función.
- Seguir las especificaciones descritas en este guion sin cambiarlas, salvo justificación y previa consulta con el profesor.

Importante: No se corregirá ningún proyecto en el que haya errores de compilación.

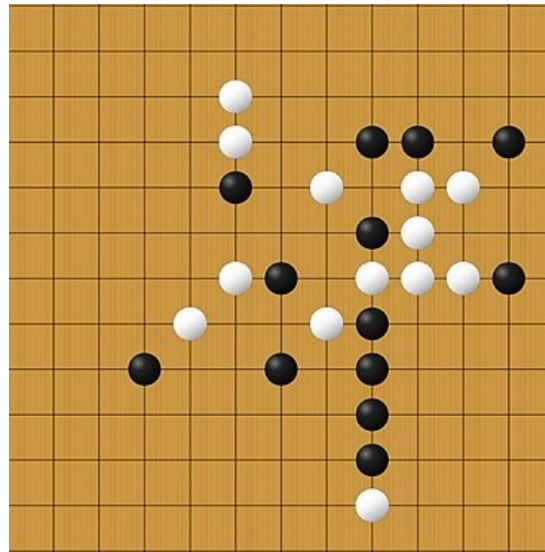
Juego del Pente (10 puntos)

El juego **Pente** es un juego de tablero de 19 filas y 19 columnas en donde dos jugadores se turnan para colocar su ficha en una casilla libre. En la imagen se muestran las fichas en intersecciones, pero para facilitar la interpretación e implementación en Java, se considerará que las fichas se posicionan en casillas de coordenadas (*fila*, *columna*), es decir, en los cuadros del tablero.

Los jugadores pueden capturar fichas del oponente si colocan sus propias fichas a ambos lados de un par de fichas del oponente (exactamente dos fichas). Por ejemplo, si las fichas están colocadas en la posición '**X O O _**' y es el turno de **X** y coloca su ficha de modo que quede '**X O O X**', las dos fichas del jugador **O** serían capturadas: '**X _ _ X**'. Esto puede ocurrir en cualquier dirección del tablero. Además, en una misma tirada, se podrían capturar más de dos fichas si logramos capturas simultáneas en varias direcciones del tablero.

El **juego termina** en alguna de las circunstancias siguientes:

- Cuando un jugador obtiene **5 fichas en línea** (horizontal, vertical o diagonal) gana la partida.
- Cuando un jugador **captura un número de fichas** del oponente, gana la partida.
- Cuando **no quedan más casillas libres**, se produce un empate.



Vamos a realizar la implementación de este juego en Java siguiendo las siguientes especificaciones.

(2 puntos) Programa principal

La función **main** será la encargada de crear el tablero como una matriz de enteros de NxN elementos inicializados a 0 (casilla vacía). Para representar las fichas de los jugadores (O) y (X) usaremos valores enteros. Para ello, definiremos las siguientes constantes para trabajar en el tablero a través de ellas.

- `public static final int CASILLA_VACIA=0;`
- `public static final int FICHA_CIRCULO=1;`
- `public static final int FICHA_EQUIS=2;`

Se deben incluir las siguientes consideraciones en el programa principal:

- a) **(0.5 puntos)** El programa deberá solicitar el tamaño del tablero (mínimo 10 y máximo 19 filas y columnas) y el número de fichas capturadas con el que se gana la partida (mínimo 5 y máximo 10). El programa funcionará de forma genérica independientemente de los valores seleccionados.
- b) **(0.25 puntos)** El programa solicitará el nombre de los jugadores para pedir por turnos que introduzcan la posición del tablero donde quieren colocar la ficha (fila y columna).
- c) **(0.25 puntos)** El programa informará al usuario si la casilla introducida es errónea (no existe o está ya ocupada) para que inserte una nueva posición.
- d) **(0.5 puntos)** Si la casilla introducida es correcta, se mostrará el tablero con la nueva ficha en su posición. Si se han producido capturas, se mostrará un mensaje por pantalla informando de ello y del número de fichas totales capturadas por los jugadores. Es deseable que la última ficha colocada aparezca resaltada en el tablero.
- e) **(0.5 puntos)** El programa continuará hasta que se produzca el final de la partida, bien porque algún jugador ha conseguido colocar 5 fichas en línea, bien porque ha logrado las capturas necesarias o bien porque no quedan casillas libres.

(8 puntos) Las funciones del juego

Dentro de la clase donde se implementa el programa principal se crearán las funciones necesarias para manejar la lógica del juego. En particular, se crearán al menos las siguientes funciones:

- (0.5 puntos) **public static void mostrarTablero(int [][] tablero)**. Visualizará el tablero en pantalla de forma apropiada para que el usuario lo entienda. Habrá que escribir O o X para representar las fichas de los jugadores, en lugar de valores numéricos.
- (0.5 puntos) **public static void mostrarTablero(int [][] tablero, int fila, int columna)**. Igual que la función anterior, pero además recibe por parámetro la posición de la última ficha colocada para poder mostrarla de forma resaltada.
- (0.5 puntos) **public static boolean colocarFicha(int [][] tablero, int ficha, int fila, int columna)**. Coloca la ficha en la posición indicada por las coordenadas (*fila*, *columna*). Devuelve *true* si ha habido éxito y *false* si no.
- (3 puntos) **public static int capturarFichas(int [][] tablero, int fila, int columna)**. Comprueba si en la última posición donde se ha colocado una ficha se pueden realizar capturas. En caso afirmativo, elimina las fichas capturadas del tablero y devuelve el número de fichas capturadas.
- (3 puntos) **public static boolean comprobarLinea(int [][] tablero, int fila, int columna)**. Devuelve *true* si en el tablero hay cinco fichas en línea (horizontal, vertical o diagonal), *false* en caso contrario. La comprobación se realizará desde la posición de la última ficha colocada.
- (0.5 puntos) **public static boolean casillasLibres(int [][] tablero)**. Devuelve *true* si en el tablero hay alguna casilla libre, o *false* si el tablero está completo.

Si es necesario, las funciones anteriores se pueden dividir en funciones más pequeñas (divide y vencerás) que permitan reutilizar al máximo el código creado. Esta reutilización del código es valorada en la corrección de la tarea.

Entrega de la tarea

El proyecto se desarrollará con el IDE Netbeans. Se creará un proyecto con el nombre **JuegoPente**. La **carpeta del proyecto Netbeans** se comprimirá en un único fichero en formato .ZIP que se subirá al buzón de la tarea en la plataforma Moodle.

IMPORTANTE: Lo que se debe entregar es la carpeta del proyecto Netbeans completa, no ficheros .java individuales. No se corregirán proyectos que no cumplan este requisito de entrega.

El archivo se nombrará siguiendo las siguientes pautas:

Apellido1_Apellido2_Nombre_PROG_ProyectoPrimerTrimestre.zip

Resultados de aprendizaje y criterios de evaluación relacionados

En esta actividad se evalúan los siguientes resultados de aprendizaje con los criterios de evaluación que se relacionan para cada uno de ellos:

- RA 1. Reconoce la estructura de un programa informático, identificando y relacionando los elementos propios del lenguaje de programación utilizado.
 - i) Se han introducido comentarios en el código.
- RA 2. Escribe y prueba programas sencillos, reconociendo y aplicando los fundamentos de la programación orientada a objetos.
 - c) Se han instanciado objetos a partir de clases predefinidas.
 - d) Se han utilizado métodos y propiedades de los objetos.
 - e) Se han escrito llamadas a métodos estáticos.
 - f) Se han utilizado parámetros en la llamada a métodos.
 - g) Se han incorporado y utilizado librerías de objetos.
- RA 3. Escribe y depura código, analizando y utilizando las estructuras de control del lenguaje.
 - a) Se ha escrito y probado código que haga uso de estructuras de selección.
 - b) Se han utilizado estructuras de repetición.
 - e) Se han creado programas ejecutables utilizando diferentes estructuras de control.
 - f) Se han probado y depurado los programas.
 - g) Se ha comentado y documentado el código.
- RA 5. Realiza operaciones de entrada y salida de información, utilizando procedimientos específicos del lenguaje y librerías de clases.
 - a) Se ha utilizado la consola para realizar operaciones de entrada y salida de información.
- RA 6. Escribe programas que manipulen información seleccionando y utilizando tipos avanzados de datos.
 - a) Se han escrito programas que utilicen arrays.

Rúbrica de evaluación

Cada apartado puntuable del proyecto se valorará con la siguiente rúbrica.

#	Criterio	Porcentaje
1	El programa/función implementado no cumple los requisitos, no soluciona de forma algorítmica el ejercicio o las soluciones obtenidas por el programa no son las esperadas, el código no compila o contiene errores.	0%
2	El programa/función implementado soluciona de forma algorítmica el ejercicio pero falla con datos de entrada no permitidos.	25%
3	El programa/función implementado tiene fallos inesperados en situaciones específicas o concretas, es decir, falla para un determinado caso o valor de entrada, pero en general el resultado obtenido es válido.	50%
4	El programa/función implementado cumple los requerimientos pero: <ul style="list-style-type: none">• El código no es legible o no está bien estructurado.	75%
5	El programa/función implementado se ajusta perfectamente a la especificación: <ul style="list-style-type: none">• Se validan los datos de entrada.• El resultado obtenido es válido para cualquier dato de entrada.• El código es modular y se emplean funciones/métodos adecuadamente.• El código es legible y usa comentarios relevantes y/o Javadoc.	100%