



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE  
ESCUELA DE INGENIERÍA  
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN  
SEGUNDO SEMESTRE DE 2014

## IIC 1103 - Introducción a la Programación

### Tarea 4

## Objetivo General

Esta tarea tiene como objetivo que aprendas a:

- Utilizar orientación a objetos para resolver problemas de programación. Esto significa modelar e implementar clases, incluyendo sus acciones e interacciones.
- Incluir la lectura y escritura de archivos en el funcionamiento de programas.
- Modelar y resolver problemas de programación utilizando un enfoque recursivo.

## Introducción

En esta tarea deberás modelar el movimiento de fichas sobre tableros, y resolver el problema de decidir si un conjunto de fichas se puede mover sobre un tablero de acuerdo a ciertas restricciones.

Conceptualmente, consideraremos que un tablero es una grilla rectangular que tiene algunas posiciones *libres* y otras *prohibidas*. Para representar un tablero utilizaremos archivos de texto con el siguiente formato: En la primera línea se encuentran las dimensiones de la grilla, separadas por una coma y cada una de las líneas sucesivas contiene las coordenadas de una de las posiciones prohibidas, también separadas por coma. Todas las posiciones que no sean mencionadas en el archivo son posiciones libres. Por ejemplo, el archivo

tablero.txt
5,6
0,1
2,2
4,5

representa un tablero de  $5 \times 6$  en el que las únicas posiciones prohibidas son  $(0,1)$ ,  $(2,2)$  y  $(4,5)$ . Consideraremos que los índices de las posiciones del tablero comienzan desde cero.

Modelaremos también fichas sobre los tableros. Un tablero puede contener varias fichas, pero no puede haber dos fichas en una misma posición ni fichas en posiciones *prohibidas*. Cada ficha tendrá una lista de movimientos. Cada movimiento es representado por un par ordenado que indica cuánto avanzar en cada coordenada. Por ejemplo, una ficha que se encuentra en la posición  $(2,3)$  cuya lista de movimientos es  $[(1,2), (-2,-1)]$  podría moverse en el siguiente paso a la posición  $(3,5)$  o a la posición  $(0,2)$ . Una ficha también será representada por un archivo, el cual indicará los movimientos de la ficha. Estos movimientos son entregados uno por línea como pares ordenados separados por coma. Por ejemplo, el archivo

ficha.txt
1,2
-2,-1

representa una ficha cuyos movimientos están dados por la lista  $[(1,2), (-2,-1)]$ .

## Enunciado

En esta tarea deberás primero escribir una clase **Tablero** y una clase **Ficha**. A continuación se muestran algunos requerimientos para estas clases:

- **Tablero**: Su constructor recibe como parámetro el nombre del archivo en el cuál se indican las dimensiones y posiciones *prohibidas*. Debe tener dentro de sus atributos una lista con cada una de las fichas que se encuentran sobre el tablero, y debe tener un método para agregar una ficha en una posición del tablero. Al inicializar el tablero, la lista de fichas debe ser vacía.
- **Ficha**: Su constructor recibe como parámetro el nombre del archivo en el cuál se indican los movimientos de la ficha. Debe tener como atributo la posición en la que se encuentra cuando está en un tablero. Antes de estar en un tablero, la posición debe ser un objeto de tipo **None**. También debe tener como atributo el nombre del archivo correspondiente a la ficha.

Debes tener claro que lo anterior no será suficiente para realizar tu tarea, por lo que deberás agregar lo que consideres pertinente para poder llevar a cabo todo lo que se pide a continuación. Una vez que tengas modelado e implementado lo anterior, debes escribir un nuevo método `se_puede_recorrer(self, archivo_salida)` en la clase **Tablero**. Este método debe retornar **True** si existe una forma de mover sucesivamente las piezas del tablero de tal forma que por cada posición permitida del tablero pase exactamente una ficha una sola vez. Por ejemplo, supongamos que tenemos inicialmente los siguientes tres archivos:

tablero.txt	caballo.txt	rey.txt
3,3	1,2	1,0
1,0	1,-2	1,1
	-1,-2	0,1
	-1,-2	-1,1
	...	...

Si generamos el tablero correspondiente a `tablero.txt` y le agregamos las fichas correspondientes a `caballo.txt` y `rey.txt` en las posiciones (0,0) y (2,2) respectivamente, obtenemos la configuración que se muestra en la Figura 1. Nuestro método debe entonces responder a la pregunta de si se puede mover las fichas sobre el tablero de tal forma que por cada posición libre del tablero pase exactamente una ficha una sola vez. En este caso el método debiese retornar **True**, pues tal como se muestra en la Figura 2, es posible realizar lo que se pide. Además, en caso de que el método retorne **True**, debe escribir en un archivo cuáles fueron los movimientos para llevar a cabo lo que se pide. El nombre de este archivo viene dado por el parámetro `archivo_salida`, y debe contener una pequeña descripción seguida de las posiciones por las que pasó cada pieza. Por ejemplo, para los movimientos que se muestran en la Figura 2 el archivo de output sería:

output.txt
Recorriendo tablero.txt con caballo.txt y rey.txt
caballo.txt: (0,0)->(1,2)->(2,0)
rey.txt: (2,2)->(2,1)->(1,1)->(0,1)->(0,2)

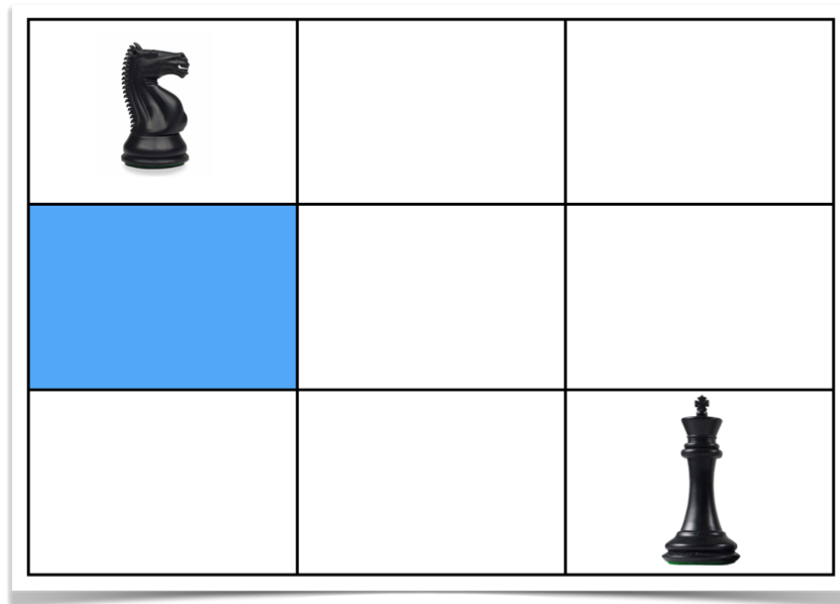


Figura 1: Ejemplo de configuración inicial. La posición prohibida es representada con el color celeste y, tal como en las imágenes de la Tarea 3, consideraremos que la posición  $(0,0)$  corresponde a la esquina superior izquierda.

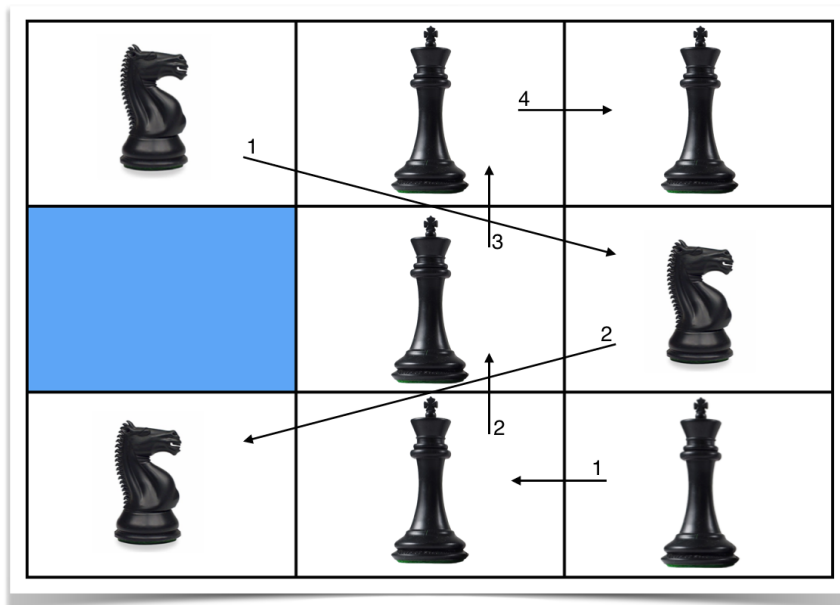


Figura 2: Movimientos para resolver el problema asociado a la función `se_puede_recorrer` comenzando desde la configuración que se muestra en la Figura 1.

# Programa

Tu programa deberá ser capaz de realizar lo siguiente:

- Crear un archivo correspondiente a una ficha con sus movimientos y guardarlo en la misma carpeta en que se encuentra tu tarea.
- Crear un archivo correspondiente a un tablero con sus posiciones prohibidas y guardarlo en la misma carpeta en que se encuentra tu tarea.
- Cargar un tablero y un conjunto de fichas en posiciones arbitrarias sobre él a partir de archivos. Una vez cargado lo anterior, tu programa debe ofrecer al usuario las siguientes dos opciones:
  - Mostrar el estado actual del tablero de alguna forma comprensible.
  - Preguntar cuáles son las casillas a las que se puede mover una pieza.
  - Preguntar si es posible mover las fichas sobre el tablero de acuerdo a las restricciones explicadas en la descripción del método `se_puede_recorrer`.

Para llevar a cabo lo anterior, tu programa debe ofrecer un diálogo que permita interactuar usando la consola. Aunque la forma particular del diálogo no esté especificado en el enunciado, se evaluará que esta sea razonable. Un ejemplo sería:

```
>> (1) Crear una nueva ficha
>> (2) Crear un nuevo tablero
>> (3) Cargar un tablero con fichas
>> (0) Salir
>> Seleccione una opción: 1
>>
>> Ingrese el nombre del archivo de la nueva ficha: caballo.txt
>> Ingrese el siguiente movimiento de la ficha o un 0 para terminar: 1,2
>> Ingrese el siguiente movimiento de la ficha o un 0 para terminar: 1,-2
>> Ingrese el siguiente movimiento de la ficha o un 0 para terminar: -1,-2
>> Ingrese el siguiente movimiento de la ficha o un 0 para terminar: 0
>> ----- ficha caballo.txt creada correctamente -----
>>
>> (1) Crear una nueva ficha
>> (2) Crear un nuevo tablero
>> (3) Cargar un tablero con fichas
>> (0) Salir
>> Seleccione una opción: 1
>>
>> Ingrese el nombre del archivo de la nueva ficha: rey.txt
>> Ingrese el siguiente movimiento de la ficha o un 0 para terminar: 1,0
>> Ingrese el siguiente movimiento de la ficha o un 0 para terminar: 0,-1
>> Ingrese el siguiente movimiento de la ficha o un 0 para terminar: 0,1
>> Ingrese el siguiente movimiento de la ficha o un 0 para terminar: 0
>> ----- ficha rey.txt creada correctamente -----
>>
>> (1) Crear una nueva ficha
>> (2) Crear un nuevo tablero
>> (3) Cargar un tablero con fichas
>> (0) Salir
>> Seleccione una opción: 2
```

```

>>
>> Ingrese el nombre del archivo del nuevo tablero: tablero.txt
>> Ingrese las dimensiones del tablero: 3,3
>> Ingrese la siguiente posición prohibida o un 0 para terminar: 1,0
>> Ingrese la siguiente posición prohibida o un 0 para terminar: 0
>> ----- tablero tablero.txt creada correctamente -----
>>
>> (1) Crear una nueva ficha
>> (2) Crear un nuevo tablero
>> (3) Cargar un tablero con fichas
>> (0) Salir
>> Seleccione una opción: 3
>>
>> Ingrese el nombre del archivo correspondiente al tablero: tablero.txt
>> Ingrese el nombre del archivo de la próxima ficha o un 0 para terminar: caballo.txt
>> Ingrese la posición inicial para caballo.txt: 1,0
>> Posición prohibida, inténtelo nuevamente: 0,0
>> Ingrese el nombre del archivo de la próxima ficha o un 0 para terminar: rey.txt
>> Ingrese la posición inicial para rey.txt: 0,0
>> Posición ocupada por otra ficha, inténtelo nuevamente: 2,2
>> Ingrese el nombre del archivo de la próxima ficha o un 0 para terminar: 0
>>
>> (1) Imprimir el tablero con las fichas
>> (2) Preguntar por los movimientos de una pieza
>> (3) Verificar si se pueden mover las piezas sobre el tablero sin repetir posiciones.
>> (0) Volver al menú principal
>> Seleccione una opción: 1
>>
>> 1 0 0
>> X 0 0
>> 0 0 2
>>
>> 0: Posición libre | X: Posición prohibida | 1: caballo.txt | 2: rey.txt
>>
>> (1) Imprimir el tablero con las fichas
>> (2) Preguntar por los movimientos de una pieza
>> (3) Verificar si se pueden mover las piezas sobre el tablero sin repetir posiciones.
>> (0) Volver al menú principal
>> Seleccione una opción: 2
>>
>> (1) caballo.txt
>> (2) rey.txt
>> Seleccione una pieza: 1
>>
>> caballo.txt se puede mover a las posiciones (1,2) y (2,1)
>>
>> (1) Imprimir el tablero con las fichas
>> (2) Preguntar por los movimientos de una pieza
>> (3) Verificar si se pueden mover las piezas sobre el tablero sin repetir posiciones.
>> (0) Volver al menú principal
>> Seleccione una opción: 3

```

[illegible]

Puedes suponer que el usuario siempre ingresa nombres de archivos existentes para cargar las fichas y el tablero, y nombres de archivos que no existen para guardar el output. Sin embargo, no puedes hacer suposiciones respecto del resto de los elementos que ingresa el usuario. Por ejemplo podría ingresar ‘hola’ cuando estás esperando un par ordenado, o podría intentar situar una ficha en una posición prohibida o fuera de la grilla. Tu programa no puede generar errores frente a estas situaciones y debe volver a pedir al usuario lo que corresponde.

## Indicaciones Generales

- Recuerda que la tarea es individual, y que las copias serán sancionadas con nota 1,1 final en la asignatura.
- La entrega de esta tarea es a través de SIDING en la sección cuestionario. Podrás subir tu tarea las veces que creas necesario mientras el plazo de entrega sea válido.
- La fecha de entrega es el miércoles 19 de noviembre hasta las 23:59 PM. No esperes hasta el final, pues el buzón se cierra automáticamente y no se aceptarán tareas entregadas tarde por otros medios.