Méndez Ávila Luis Geovanni. **317143980**
Cuestionario de la práctica 9

¿Cuál fue el proceso de programación del ajedrez?
R: Primero se hizo la clase Position que servirá para usar como atributo en la clase Piece y se hicieron un par de numeraciones, una de color que contenía BLACK, WHITE, NONE y otra que contenía los tipos de piezas ROOK, BISHOP, QUEEN, KING, KNIGHT, PAWN, EMPTY.

Después se comenzó por hacer una clase abstracta Piece , la cual iba a contener los métodos de todas las demás piezas, sin embargo tiene un método no implementado, el cual debe ser implementado dependiendo de los movimientos de las demás piezas.

Una vez hecha la clase pieza se hizo la clase Rook, a la cual se le implementó el método abstracto de Piece, la implementación consiste en que deja fijo uno de los valores de posición de la pieza y el otro varía dependiendo de un ciclo for.

En la práctica 7 donde se aplicó por primera vez lo que vimos de herencia, pasamos los movimientos de las piezas tipo Rook a una clase Queen. En mi caso yo le puse a esa clase FakeQueen porque no estaban todos los movimientos posibles implementados.

Después de eso, se hizo la clase Board que construirá una matriz de piezas de 8x8 . En Board se hizo un método especial que hace que únicamente se puede instanciar una vez el tablero.

Después se prosiguió con la creación de las clases hijas de Piece, cada una debía implementar de forma distinta el método abstracto de la clase Piece.

Una vez hechas las piezas, se hizo la clase Game y la clase ChessGUI, la primera aún no entiendo bien su finalidad. Mientras que ChessGUI es la interfaz gráfica de processing, donde se definen todos los métodos que se pueden usar para dibujar los elementos programados previamente. Para ello previamente se debía incorporar al directorio toda la librería de processing.

¿Cuál es la complejidad del programa completo?(Qué tanto poder de cómputo requiere)
R: Supongo que es cuadrática, hay ciclos anidados en los algoritmos de las piezas y se ejecuta relativamente rápido … creo.

¿Cúal es el algoritmo o la función que es más compleja de ejecutar?

R: getLegalMoves … creo.

¿Qué conceptos vistos en clase aplicaste y en donde?
R: Matrices en la clase Board ; Herencia al heredar Piece a las demás clases hijas; Ciclos for, if en los algoritmos de getLegalMoves; Clases abstractas en la clase Piece; Creación de clases en prácticamente todo; Sobreescritura de métodos en todos los toString, equals y getLegalMoves.

¿Es un proyecto difícil?
R: Algo si no tienes mucha práctica y conocimiento en java.

Después de haberlo hecho entre todos ¿Crees que podrías ahora implementarlo completo tú solo?
R: Tal vez, como dije anteriormente, es cuestión de tener un poco más de práctica y a lo mejor un poco más de conocimiento sobre el lenguaje. También es importante tener una buena idea de cómo diseñar algoritmos, la parte difícil de la práctica es justamente el diseño de los mismos, en lo personal tuve problemas para hacer la clase King y Knight. De nada te sirve conocer bien el lenguaje java si no sabes diseñar algoritmos.

Aunque podría hacerlo con más práctica, conocimiento, imaginación y paciencia … a lo mejor sin tantas presiones encima.

Describe con tus palabras cómo implementarías la regla Peón al paso
R: No se me ocurre algo.

Describe con tus palabras cómo implementarías detectar que hay un jaque
R: Un método que devuelva un valor booleano y que reciba una pieza con .getLegalMoves y con un if si la posición del rey coincide con uno de los movimientos de getLegalMoves de la pieza en cuestión entonces devuelva true y false si no está.

Describe con tus palabras cómo implementarías enroque
R: Un método en la clase Rook que revise si entre el rey y la torre no hay alguna pieza de por medio. Si es así entonces que se pueda mover la torre hasta la posición del rey y que le cambie la posición al rey por su posición … no se me ocurre con qué tipo de ciclo pueda hacerlo bien.