CS 110 Lab Sheet 8

Lab cycle starting 20th November 2019, normal deadline two weeks after your lab in this cycle.

General Comment

All these tasks are about writing classes. But as well as the objects, you will have to write some code to test them. So you will also need a main method that creates your objects.

Stage 1: Fruit

Write a class called Fruit that represents the properties of fruit. Fruit should have the following properties: name and colour. You will need to think about what methods you need and any constructor. For example, does it make sense to be able to change the a fruit's properties after you have created it? (There is no right answer to that question here - it's up to you.) Next write a main method in the same class to create and test your Fruit class. The main method should create a number of Fruit objects and call the various methods, and output the results to the screen.

Stage 2: Regular Polygons

A regular polygon is a polygon in which all the sides are the same length. Simple examples are equilateral triangles and squares. The area of a regular polygon is given by the formula:

```
area = (S*S*N)/(4*tan(Pi/N))
```

where `S` is the length of one of the sides (remember they are all the same length), and `N` is the number of sides. The function tan, or tangent, is a trigonometric function – it doesn't matter if you are not familiar with trigonometry – and in Java you can access it as `Math.tan(e)` where `e` is a double. You can also access the constant Pi as `Math.PI`.

Write a RegularPolygon class with a constructor that takes a number of sides and a side length. Your class should have methods to return and set the number of sides and the length of a side, as well as compute the area (using the formula above).

This time write a separate class to test RegularPolygon - that is, your program should have TWO classes and hence, sensibly, TWO files.

Assessed Task Stage 3: Chessboard

This is much simpler than it seems at first sight.

YOU ARE NOT BEING ASKED TO WRITE A CLASS THAT CAN PLAY CHESS, or know anything about the rules of chess - just a board on which you can place strings identifying pieces. Your code does not have to worry about what is placed on the board and if they are “legal” pieces - any string value is fine.

A chessboard is an 8x8 array of squares each of which can hold a piece. The squares are identified by numbers from 1 to 8 (for the rows) and letters from a to g (for the columns). Write a Chessboard class with two methods one to set the piece on each square; and one to get the piece on each square.

1. The pieces should be identified by String (there is actually a specified format for chess pieces, but you need not worry about it - any String is fine; it does not have to represent an actual piece - it can be “Queen”, “Knight”, or “Lemon” - there is no need to check).

2. The positions should be identified by a number 1-8 for rows and a lower-case character a-h for columns.

So, for example:

```
Chessboard board = new Chessboard();

board.setPiece(“White King”, ‘a’, 4);

System.out.println(board.getPiece(‘a’, 4);
```

would put a White King on column a, row 4, and then get the value of that piece (printing ‘White King’). (Because you don’t have to worry about legal chess pieces, `board.set(“Lemon”, ‘a’, 4);` would also be fine.)

Hints: the obvious way to define the board is as a two-dimensional array of Strings. The numbers that identify rows are obviously already close the to numbers you need to index the array rows - they just start from 1 instead of 0. The trickier bit is turning the characters into numbers. One way is to use switch statements. Another way is to run the program on the next page and think about it:-)

```
class CharTest {
    public static void main(String[] args) {
        char c = 'a';
        int i = (int)c;
        System.out.println(i);
    }
}
```

Don’t spend too much time on this if you don’t get it– using a switch statement is fine.

If you don’t recall this from earlier, then a `char` is a single character as opposed to a String which is a sequence of zero or more characters.

As before, write another class to test your Chessboard class - it should just set a few pieces on the board, and then use the get method to retrieve and print them, nothing complicated.

Challenge

Write a `Chesspiece` class to represent a chess piece and change your Chessboard class to store `Chesspiece` objects instead of Strings. It’s up to you how you represent chess pieces (remember they have a name and a colour and they should be separately stored) and what methods, constructors etc. you choose.

You can also write a ChessGame class that uses your Chessboard and Chesspiece classes to represent a complete chess game. Don’t worry - you don’t have to program the actual game play. Your class should be able to record the moves made by the players - player 1 and player 2 - with a method takes the starting and ending square of a move (you can assume the players know the rules, you don’t have to check if moves are legal). It should then store that move in some way (hint: you don’t know how many moves there will be) and it should be able to work out which piece was actually moved (remember that your Chessboard class has information on which piece is on which square). If you ‘get’ Object Oriented programming, this is nowhere near as hard as it looks at first.