First Guesses

As a starting point, we'll implement the basic components of a simple card game in Java.

The Game: "Sleuth"

Sleuth is a two-player game of deduction where a player is dealt a hand of 3 cards, and it is the job of the other player to guess what cards the first player has.

The cards in the game of Sleuth have three properties: a *shape*, a *colour* and a *number*. The four possible shapes are diamond, hexagon, rhombus and circle. The four possible colours are red, green, blue and yellow, and the four possible numbers are (you guessed it), 1, 2, 3 and 4.

The second player can ask the first player a limited set of questions, how many card the play is holding with: a particular colour, a particular shape, a particular number, or a particular combination of two of the properties properties.

The second player may also guess if the first player has a particular card.

The Starting Point

The scaffold contains a Runner class, and nothing else. Runner is to facilitate you running any test/debugging code you need to complete the task, however it is not part of the tests.

You will have to create two public classes: Card and Player that implement the necessary methods to support the functionality described above/ Additionally, you will need to make two enums: Shape and Colour to help model the card properties.

The enums

This should be fairly straightforward, we need two public enums, each with the values specified below.

* Shape has values: RHOMBUS, CIRCLE, DIAMOND, HEXAGON.
* Colour has values: RED, GREEN, YELLOW, BLUE.

*(Note, this data should be in ALL CAPS, because enum values are constants)*

The Card

Each card has three components:

* Number
* Colour
* Shape

You can track these however you choose, but there should be a reasonably obvious way to do this.

The Card class should have the following methods:

* a single public constructor that takes three parameters: a Colour, Shape and an int
* it shouldn't have any public data
* three public methods, each with no parameters and returning the requested data
  + getColour
  + getShape
  + getNumber
* A public toString method that overrides the standard toString method, returning the card data in the format "number, colour, shape"

The Player

A Player should have *only* these public methods:

* a public constructor that takes a List of Cards.
* howManyColour which takes a Colour as a parameter and returns an int with how many of the specified colour cards the player is holding.
* howManyShape, taking a Shape as a parameter and returns an int with the same logic as above, but for a shape.
* howManyNumber, as above, but takes an int parameter and functions as above, but for the number property.
* howManyColourNumber which takes a Colour and an int and returns how many cards with both that colour and number the player is holding.
* howManyColourShape, as before, with taking a Colour and a Shape and checking the players cards for those properties.
* howManyShapeNumber, again, as before, but checking the Shape and Number.
* hasCard which takes a Card as a parameter and returns a boolean if the player is holding that card.

Player will also need variables to keep track of relevant data, these should all be private.