Roulette
French Roulette is a casino game that uses a wheel with 37 pockets numbered 0 to 36. To play a “straight-up” bet, you lay your money beside any single number. The croupier spins the wheel in one direction and a ball in the opposite direction. When everything slows to a stop, if the ball lands in the pocket with your chosen number then you win $35 for every $1 you bet.

Let’s write a simple Java application that uses the `Spinner` class to build an object that can simulate a French Roulette wheel. The simple application will spin the wheel and print the resulting number.

The algorithm is:

```java
build a spinner object that can spin numbers from 0 to 36
spin the spinner
print the spinner's stoppedAt value
```

Here’s the application:

```java
public class FrenchRoulette {
    public static void main(String[] args) {
        // build the French Roulette wheel
        Spinner spinner = new Spinner(0, 36);
        // spin it
        spinner.spin();
        // print the final result
        System.out.println("Wheel rolls " + spinner.stoppedAt);
    }
}
```
**Foreign Currency Exchange**

People who travel abroad can’t normally buy stuff using the currency of their homeland. For example, a European traveling in the U.S. can’t use euros to pay for his breakfast at Ma’s Diner in Kit Carson, Colorado. Instead, he must go to a bank and use his euros to buy an equivalent amount of U.S. dollars. Since currencies differ in value, the bank applies an *exchange rate* to determine how many dollars the traveler can buy.

**Example**

Suppose 1 dollar = 0.9230 euro so that \( \frac{1}{0.9230} \) dollars = 1 euro.

Let’s pretend that our European visitor arrives with €250 and needs to buy dollars.

\[
€250 = \frac{250}{0.9230} = $270.86
\]

He returns to his home country with $55 left in his pocket. Since he no longer needs the left-over dollars, he sell them to the bank for:

\[
$55 = 55 \times 0.9230 = €50.77
\]

In the world of finance, it is convention to express the X-to-Y exchange rate as \( \frac{X}{Y} \) where \( X \) and \( Y \) are standard codes taken from the ISO 4217 code list published by the International Organization for Standardization.

**Example**

The ISO 4217 code for the U.S. dollar is USD and for the euro is EUR. The dollar-to-euro exchange rate is expressed as USD/EUR.

\[
\text{USD/EUR} = 0.9230 \text{ means that 1 USD} = 0.9230 \text{ EUR.}
\]

The \( \frac{X}{Y} \) exchange rate is \( \frac{1}{Y/X} \) exchange rate.

**Example**

If USD/EUR = 0.9230 then EUR/USD = \( \frac{1}{0.9230} \).

\[
x \text{ USD} = x \times 0.9230 \text{ EUR and } x \text{ EUR} = \frac{x}{0.9230} \text{ USD}
\]
I want to build a class from which programmers can build objects that do USD/X currency conversions, where X is some foreign currency. The details of the class are not important so long as its methods are well enough specified so that my fellow programmers can effectively use it.

The name of the class is **USD_X_Exchange** and it provides this explicit constructor:

```java
public USD_X_Exchange( double rate )
// Construct object to convert U.S. dollars and currency X.
// Parameter rate is the USD/X exchange rate.
```

Using the information from this specification and the Java idiom for building objects, a programmer can write the following Java statement to build an object that does conversions between U.S. dollars and euros. I’m assuming that USD/EUR = 0.9230;

```java
USD_X_Exchange usd_eur = new USD_X_Exchange( 0.9230 );
```

The class provides this method for buying dollars with euros:

```java
public double buyDollars( double x )
// Use x of currency X to buy and return equivalent amount of dollars.
```

The following Java statement uses the `usd_eur` object to determine the amount of U.S. dollars that can be bought with €250:

```java
double dollars = usd_eur.buyDollars( 250 );
```

The class has this method for buying euros with dollars:

```java
public double sellDollars( double d )
// Sell d dollars and return equivalent amount in X.
```

The following Java statement uses the `usd_eur` object to determine how many euros $55 will bring:

```java
double euros = usd_eur.sellDollars( 55 );
```

The complete Java class is found on the next page. It is best, however, if you don’t look at it. Your goal in this topic is to learn how to read the specification of the constructors and methods.
// A USD_X_Exchange exchanges U.S. and foreign currency.

public class USD_X_Exchange
{
    private double er; // USD/X exchange rate, 1 USD = er X

    public USD_X_Exchange( double rate )
    {// Construct object to convert U.S. dollars and currency X.
     // Parameter rate is the USD/X exchange rate.
    {
        er = rate;
    }

    public double sellDollars( double d )
    {// Sell d dollars and return equivalent amount in X.
    {
        return d * er; // 1 USD = er X
    }

    public double buyDollars( double x )
    {// Use x of currency X to buy and return equivalent
     // amount of dollars.
    {
        return x / er; // 1 X = 1/er USD
    }
}
**Exercises**

Enter class `USD_X_Exchange` into jGRASP and save it to a file. Compile it. Perform the following experiments using jGRASP’s Interactions pane. To answer the questions, observe jGRASP’s Workbench pane.

1. In jGRASP’s Interactions pane, enter and execute the statement:

   ```java
   USD_X_Exchange usd_eur = new USD_X_Exchange( 0.9230 );
   ```

   In jGRASP’s Workbench pane, what is the value of reference variable `usd_eur`? Find the entry for `usd_eur` and click the ![question mark](?) so that it becomes ![question mark](?)? What is the value of the object’s instance variable?

2. In jGRASP’s Interactions pane, enter and execute the statement:

   ```java
   double dollars = usd_eur.buyDollars( 250 );
   ```

   In jGRASP’s Workbench pane, what is the value of the variable `dollars`? Is this result consistent with the first example on page 2?

3. In jGRASP’s Interactions pane, enter and execute the statement:

   ```java
   double euros = usd_eur.sellDollars( 55 );
   ```

   In jGRASP’s Workbench pane, what is the value of the variable `euros`? Is this result consistent with the first example on page 2?