Exercises 15

Java classes (continued)

Exercise 15.1. - Compte - Account (continued)

We want to improve and complete the Compte class as follows:

```
1  class Compte{
2      int solde =
0;
3      void deposer(int montant){
4          solde = solde + montant;
5      }
6      void retirer(int montant){
7          solde = solde - montant;
8      }
9      void virerVers(int montant, Compte destination){
10         destination.retirer(montant);
11         this.deposer(montant);
12     }
13     void afficher(){
14         Terminal.ecrireString("solde: "+ solde);
15     }
16 }
```

1. change the method remove() - withdraw() to prevent withdrawal when the account is not sufficiently supplied.
2. modify the class to add an account number to each account.

Exercise 15.2 library

We want to write a program that represents a library with books.

Question 15.2.1

Write a class to represent books with the title, author and publisher. The operation allows to get a book, which creates a new object and display the information about a book object. Take care care of each transaction and determine whether it should be a static method or not.

Question 15.2.2

A library is a **collection** of several books. The books will be represented by the class defined in question 1. Write a **class library** that can store several books, with two operations: **add a book, view the contents of the library**. Write a `main()` method that creates one or two books, a library and use the calls to the available methods.

Exercise 15.3. - Sorted array

We'll see how this exercise can have a sorted array without ever making a sort operation. We will use products with a name, a reference, and a duty-free price. Reference is an integer. We want to write a class `Stock` that contains a list of products represented by an array of products sorted by ascending order of reference.
The Stock is initially empty and then filled slowly through an `add` operation. This `add` operation inserts the new object in the right place in the array accordingly to the order of reference.
In addition to this `add` operation, write an operation that displays the contents of a stock and an operation that searches for the price of a product at a given reference. Organize two classes Product and Stock for each of the methods on their respective data. Put the `main`() method in a third class.

Exercise 15.4 - Polynomials

A polynomial is a function that associates a numeric variable x to the sum of products obtained by multiplying x by a power of some integer coefficient.
For example:

$$f(x) = 3 * x^3 - 5 * x^2 + 2.$$

Note that the coefficient 2 implicitly applies the power x0, which is 1.
The addition of polynomials is done by adding the coefficients of power by power.
Example $(3 * x^3 - x^2 + 2 * 5) + (2 * x^3 - 4 * x - 3) = 5 * x^3 - x^2 * 5 - 4 * x - 1$
The polynomial multiplication is done using the distributive property of multiplication over addition.

For example:

$$(3 * x^3 - 5 * x^2 + 2) * (2 * x^3 - 4 * x - 3) =$$
$$(3 * x^3) * (2 * x^3 - 4 * x - 3) + (-5 * x^2) * (2 * x^3 - 4 * x - 3) + 2 * (2 * x^3 - 4 * x - 3)$$
$$= (6 * x^6 - 12 * x^4 - 9 * x^3) + (-10 * x^5 + 20 * x^3 + 15 * x^2) + (4 * x^3 - 8 * x - 6)$$
$$= 6 * x^6 - 10 * x^5 - 12 * x^4 + 15 * x^3 + 15 * x^2 - 8 * x - 6$$

Write a class of polynomials with the operations of addition, multiplication and calculate the value of the polynomial for a given x.