COMP 1020
Lab 7

MATERIAL COVERED

- Inheritance, polymorphism, instanceof, casting

Notes:

- The three exercises are cumulative – each builds on the previous one.
- Only one of the three exercises is required.
- Try to complete the Bronze and Silver exercises. The Gold exercise is trickier, as usual.

Creating a class hierarchy

1. Create three classes **`Data`**, **`Single`**, and **`List`**, as follows:

    a. Create an ***abstract*** class **`Data`**, which will contain no instance variables, no constructor, and only one method: **`double valueOf()`** which returns **`0.0`**.

    b. Create a class **`Single`** which is a subclass of **`Data`**, and which will store one **`double`** value. Provide a constructor to initialize the value. Override the **`valueOf()`** method so that it returns this value.

    c. Create a class **`List`** which is a subclass of **`Data`**, and which will store a **`double[]`** array. Provide a constructor **`List(double[] a)`** which will initialize this array. Override the **`valueOf()`** method so that it returns the ***sum*** of all the **`double`**s in the array. Note that this will always be a *full* array, not a *partially full* array. There will be *no* separate length variable.

2. Start with the **`TemplateLab7.java`** file. It creates a list of **`Data`** objects (a mixture of **`Single`** and **`List`**) using a **`Data[] myData`** array. Take a look at it. Add a loop at the indicated position which will find and print the sum of every number that appears in **`myData`**, whether it appears in a **`Single`** or in a **`List`**, using **`valueOf()`**. It should print the line

    `The sum of everything is 35.8`

Creating a bigger class hierarchy

1. Add a **`length()`** method to the **`List`** class which returns the size of the list stored in the object. You are ***not*** allowed to add a **`length()`** method to the **`Data`** or **`Single`** classes. (This would be a logical thing to do, but it would destroy the purpose of the question.)

2. Add more code to the **`TemplateLab7.java`** file at the indicated position which will find and print the total number of values that appear anywhere in **`myData`**. It should print the line **`There are 7 values in total.`** as well as the output line from the Bronze exercise.

Handling an array of Objects

1. Look at the file **`TemplateLab7Gold.java`**. Note that the **`Data[]`** array has been replaced by an **`Object[]`** array instead. This **`Object`** array contains **`Single`** and **`List`** objects, but it also contains **`String`**s and other types as well.
2. Paste your code from the Bronze and Silver exercises into this file at the places shown – they remain unchanged.
3. Complete the **`public static Data[] convert(Object[] objects)`** method, which will convert the **`Object[]`** array **`objects`** into a **`Data[]`** array as follows:
   a. The two arrays should have the same length.
   b. Any **`Single`** or **`List`** objects should be placed unchanged into the new array. (Use only a shallow copy.)
   c. Any **`String`** should become a **`List`** object containing all of the numbers that can be found (as separate tokens) in the **`String`**. Use a **`Scanner`** to scan the **`String`** for numbers. Non-numbers should be ignored. [Note: any number will give true for **`hasNextDouble()`**.] Assume that the resulting **`List`** will always have from 0 to **`MAX_LIST_SIZE`** values in it. Remember that a **`List`** object must always be a full list, not a partially-full list.
   d. Any other kind of object should be changed into a **`List`** object containing a length-0 array.
4. The new result should be:

```
The sum of everything is 45.4
There are 11 values in total.
```