Principles of Object Oriented Programming
# Spring 2019
Exercise 1

Published: 20.3.19   Due:28.3.19

1   General Description

The assignment goal is to practice the following concepts

- Class design

- UML

- The message-passing concept in OOP

You have to implemet a simulation of the florist example given in Lecture 1, in which Chris orders flowers for Robin. Specifically, your goal is to develop a system consisting of the following agents (objects):
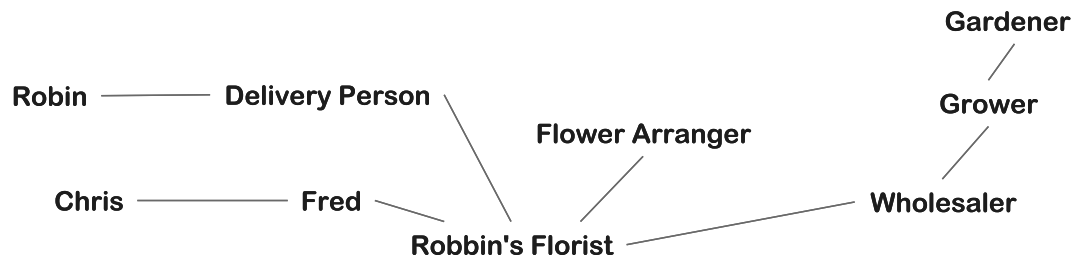
The links in the above figure represent connections between objects that should exchange messages (i.e., invoke methods) during the simulation. For example, the link between Chris and Fred means that one of these objects should invoke a method of the other one.

You should perform the following steps (the percentage of each step in the assignment grade appears in brackets):

1. (10 %) Describe the **responsibilities** of **each agent** that appears in the figure.

2. (10 %) For **each pair of agents** that are connected by a link, decide who is the **client** and who is the **server**. (It is possible that an agent is a client on some link, and a server on another link.)

3. (20 %) Design an appropriate class diagram. Think about the similarities and differences of the various objects you have to implement. The hierarchy structure in your class diagram should be reasonable, according to the principles learned in the lectures and sessions. The description should contain the methods provided by each class.

4. (60 %) Implement the simulation according the design in parts 1 - 3, using the guidelines that appear in the next section.

2 Implementation requierments

In addition to the above classes, you should implement a class representing a **Flowers-Bouquet**. The class should contain a string representing the names of the flowers in the bouquet, and a boolean field representing whether the bouquet is arranged or not.
The program should start from Chris's object that request a flowers order with a string specifying the desired flowers. Next, the message is propagated in the network, until it reaches the Gardener, who prepares an appropriated flowers bouquet (i.e., he creates a Flowers-Bouquet object, initializes it with the string that represents the flowers names, and sets the arranged field to false). Then he returns the bouquet, which eventually reaches Robin's Florist, who forwards it for delivery. To make the message-passing possible, each client object should hold a **reference** to its server object (or several server objects).
Your program should produce the following output:

```
Chris orders flowers from Fred:  Roses, Violets, Gladiolus
Fred forwards order to Robin's florist
Robin's florist forwards request to Wholesaler
Wholesaler forwards request to Grower
Grower forwards request to Gardener
Gardener prepares flowers
Gardener returns flowers to Grower
Grower returns flowers to Wholesaler
Wholesaler returns flowers to Robin's Florist
Robin's Florist request flowers arrangement from Flower Arranger
Flower Arranger arranges flowers
Flower Arranger returns arranged flowers to Robin's Florist
Robin's Florist forwards flowers to Delivery Person
Delivery Person delivers flowers to Robin
```

```
Robin accepts the flowers:  Roses, Violets, Gladiolus
```

Each object is allowed to produce only its output. In other words, a line of output that starts with a name X can be produced only by X. (For example, the line "Fred forwards order to Robin's Florist" can be produced only by Fred, the line "Flower Arranger arranges flowers" can be produced only be the Flower Arranger, etc'.) The object names should not be hard-coded in the message, but obtained using appropriate **getters**. The goal of the last requirement is making the program more general, and more flexible for modifications.

3   Submission instructions

Submit to the CS submission system a single archive file (.zip or .tar.gz) with the following contents:

1. hw1.jar - this file should contain the **source code** and **compiled class files**.

   (Check that you can run the file hw1.jar properly from the command line, as described in the submission instructions on the Assignments page.) Use the coding conventions given on the web site. Use packages for your classes: Do not use the default package.

2. hw1.pdf - a file with answers for tasks 1 - 3 of Section 1.