Lab 3: Behavioural Patterns

1 Introduction

This week kicks off our introduction to design patterns. The Model-View-Contoller pattern plays a role in your assignment, and we've discussed the Observer pattern in class. In this lab we'll be diving a little more deeply into the Observer pattern while we revisit our work with city trees.

To begin, log into MarkUs and click on the assignment called "Lab 3" to add starter files on your local machine. In the repository you should find five java files: *Main.java, Forester.java, Guildelines.java, Observer.java, and Observable.java.*

2 Programming Task

In the past, we visualized the city's tree inventory; this week we'll consider adding some new trees to this inventory. Different municipalities have guidelines that describe the locations where different species of trees can be planted. This week, we'll be exploring 2016 guidelines for tree planting from municipality of Richmond Hill, and which you can find at this URL:

https://www.richmondhill.ca/en/shared-content/resources/documents/urban-forest-planting-guidelines.pdf.

Richmond Hill's guidelines tell us that Linear Parks are too small for the roots of some kinds of trees; for such trees, Neighbourhood Parks are more appropriate. However, like all guidelines, the Richmond Hill guidelines are subject to change. The municipality may provide different recommendations in 2023; these may change again in 2024 and 2025.

Your task is to use the Observer pattern to allow urban foresters to **subscribe** to the Richmond Hill guidelines. Subscribers should be able to register to receive updates if and when guidelines change, and to unsubscribe if and when they would like to stop receiving such updates.

Your starter code contains classes that will help you get started. You have code for an **Observer** interface and an **Observable** base class. You also have code for a **Forester** class; this will "observe" instances of the **Guidelines** class and receive "updates" when guidelines change. You will be altering the file called *Guidelines.java* to implement:

- The method *register*. This is inherited from the Observable class, and should be used to register a new observer to the list of Guideline observers.
- The method *unregister*. This is also inherited from the Observable class, and should be used to unregister a new observer from the list of Guideline observers.
- The method $notifyObservers$. This should broadcast a message to all observers so that they can "update" their guidelines to the newest version.
- *addGuideline*. This will add a new location to the list of locations that guidelines indicate to be appropriate for a given tree type.
- *removeGuideline*. This will remove a location from the list of locations that the guidelines indicate are appropriate for a given tree type.

In addition you will implement the following in *Forester.java*:

- The method *update*. This should update the treeGuidelines attribute of the Forester Class in order to reference new guidelines.

3 What to Submit

1. Guidelines.java
2. Forester.java

HAVE FUN AND GOOD LUCK!