**Raditya Surya Pratama**

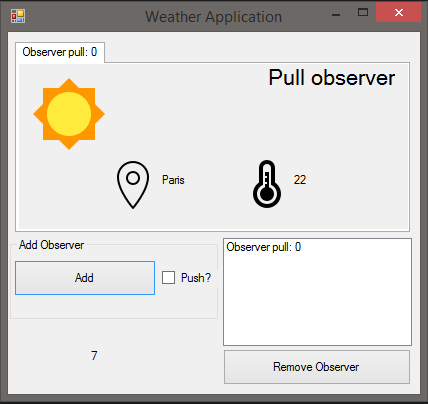
**Radu Alexandru Stoica**

Weather station

Observer Pattern

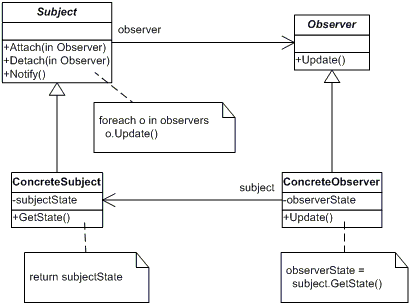
INTRODUCTION

The second assignment for DPR is to implement the observer pattern into a disk weather station application. This application is working to simulate how the weather station as subject gives information to other program as observer. The observer pattern aim to minimize the dependencies.



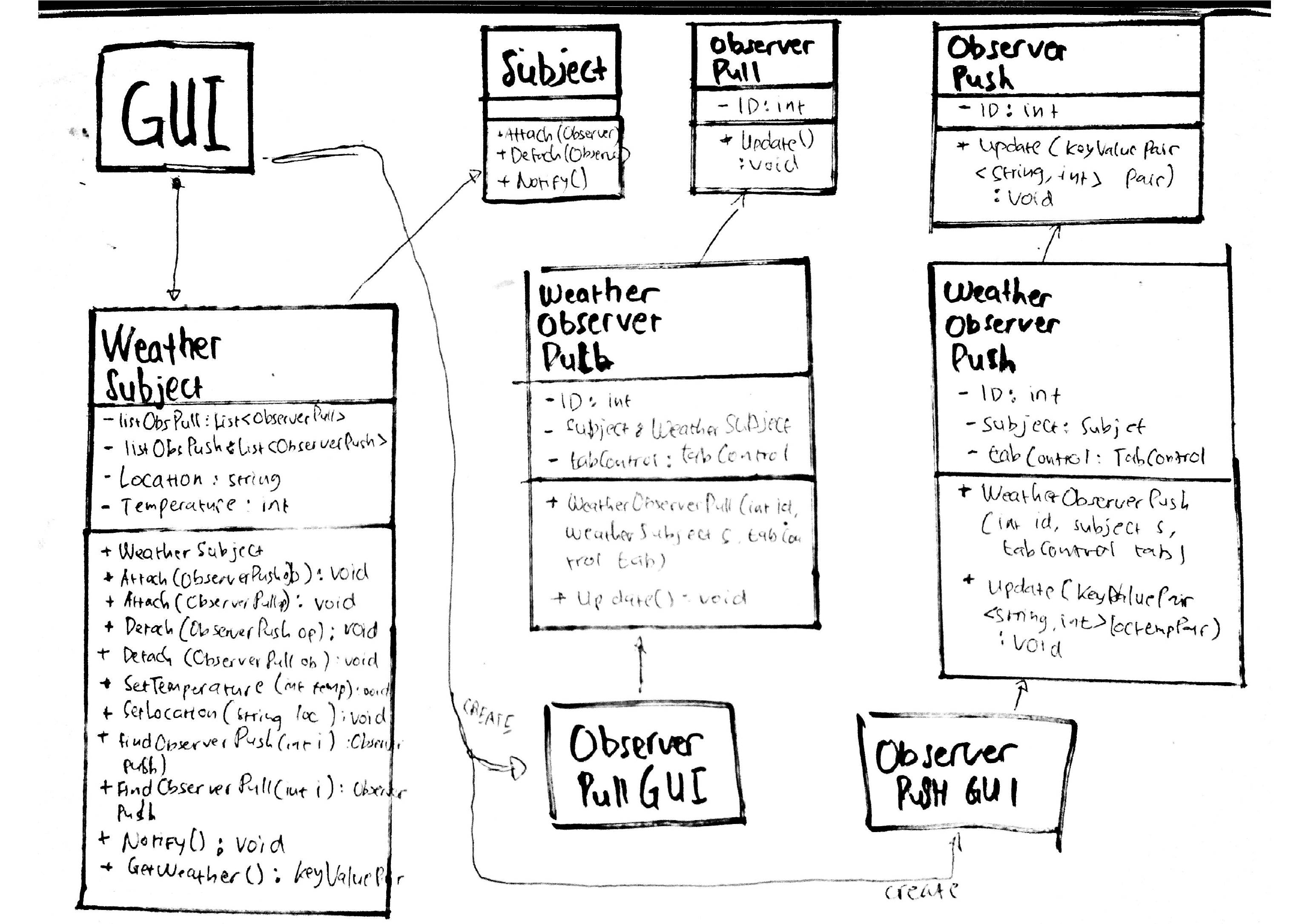
STRATEGY PATTERN

The strategy pattern is a one-to-many dependency between objects so that when one object changes state, all its dependents are notified and updated automatically.



Source: http://www.dofactory.com/net/observer-design-pattern.

UML DIAGRAM



REUSABILITY

The reusability of our code is represented by the observer class that can be used as dependencies and the subject class does not need to know who its observers are.

EXTENSIBILITY

The extensibility is represented by the abstract interface used to define the behavior for each observer strategy.

In this case, when there is new update or changes, we do not need to edit the observer class, we could just update the subject class that implement the notify method.

PATTERN DOWNSIDE

There are a couple of disadvantages of using the Observer pattern:

* You cannot create an instance of observable class and compose it with your own objects, you have to create subclass
* If not used carefully the observer pattern can add unnecessary complexity
* The order of observer notifications is undependable

Source: https://neillmorgan.wordpress.com/2010/02/07/observer-pattern-pros-cons/.

UNIT TEST

The unit test is included in the project solution.