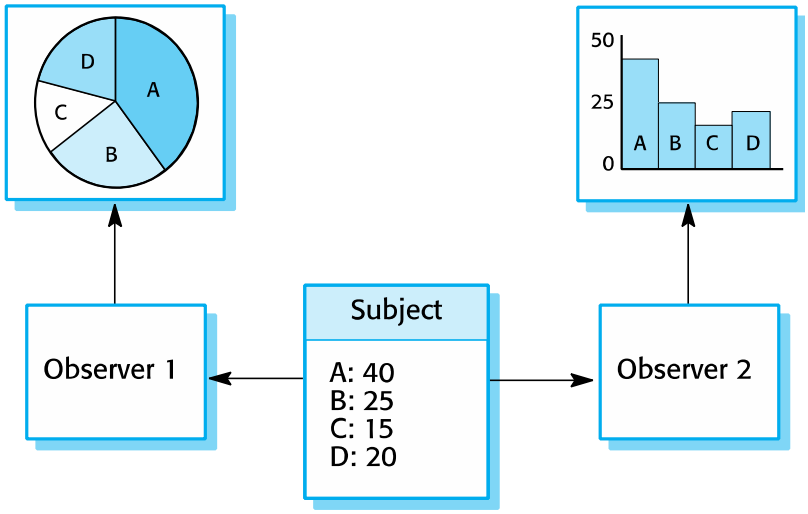
Week 12 - Quiz

Multiple displays using the Observer pattern



Problem 1:

- Given a subject, e.g., a map or matrix…
- In: state 1 → out: Display 1: PieChart of that subject
- In: state 2 → out: Display 2: BarChart of …

Sample answer:

```
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;

public class Subject {

  private List<Observer> observers = new
ArrayList<Observer>();
  private int state;
  private HashMap map;

  public Subject (HashMap m){
    this.map = m;
  }

  public HashMap getMap(){
    return this.map;
  }

  public int getState() {
    return state;
  }

  public void setState(int state) {
    this.state = state;
    notifyAllObservers();
  }

  public void attach(Observer observer){
    observers.add(observer);
  }

  public void notifyAllObservers(){
    for (Observer observer : observers) {
      observer.update();
    }
  }
}

////

public abstract class Observer {
  protected Subject subject;
  public abstract void update();
}

////

public class PieObserver extends Observer{
   public PieObserver(Subject subject){
    this.subject = subject;
    this.subject.attach(this);
  }
```

```
  @Override
  public void update() {
     if (subject.getState()==1)
         System.out.println( "Display 1: PieChart of {" +
subject.getMap() + "}");
  }
}

////

public class BarObserver extends Observer{
   public BarObserver(Subject subject){
     this.subject = subject;
     this.subject.attach(this);
  }

  @Override
  public void update() {
     if (subject.getState()==2)
         System.out.println( "Display 2: BarChart of {" +
subject.getMap() + "}");
  }
}

////

public class ObserverPatternDemo {

   public static void main(String[] args) {
     Subject subject = new Subject(null);

     new PieObserver(subject);
     new BarObserver(subject);

     subject.setState(1);
     subject.setState(2);
  }
}
```

==

Problem 2:

- Given a subject, a state presents content or data of the subject at a time
- In: {40, 25, 35, 15} → out: Display 1: PieChart of {40, 25, 35, 15}
- In: {40, 0, 35, 0} → out: Display 2: BarChart of {40, 0, 35, 0}

Sample answer:

```
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;

public class Subject {

  private List<Observer> observers = new
ArrayList<Observer>();
  private int[] state;


  public int[] getState() {
    return state;
  }

  public void setState(int[] state) {
    this.state = state;
    notifyAllObservers();
  }

  public void attach(Observer observer){
    observers.add(observer);
  }

  public void notifyAllObservers(){
    for (Observer observer : observers) {
      observer.update();
    }
  }
}

///

public abstract class Observer {
  protected Subject subject;
  public abstract void update();
}

///

public class PieObserver extends Observer{
   public PieObserver(Subject subject){
     this.subject = subject;
     this.subject.attach(this);
  }

  @Override
  public void update() {
     System.out.println( "Display 1: PieChart of {" +
subject.getState() + "}");
  }
}

///

public class BarObserver extends Observer{
```

```
   public BarObserver(Subject subject){
     this.subject = subject;
     this.subject.attach(this);
   }

   @Override
   public void update() {
      System.out.println( "Display 2: BarChart of {" +
subject.getState() + "}");
   }
}

///
```

```
public class ObserverPatternDemo {

   public static void main(String[] args) {
     Subject subject = new Subject();

     new PieObserver(subject);
     new BarObserver(subject);
     int[] s1 = {40, 25, 35, 15};
     subject.setState(s1);
     int[] s2 = {40, 0, 35, 0};
     subject.setState(s2);
   }
}
```