```
public class StringObservable {
    private List<Observer<String>> consoleWriterObservers;
    public StringObservable() {
        this.consoleWriterObservers = new ArrayList<>();
    }
    public void addObserver(ConsoleWriterObserver writer) {
        this.consoleWriterObservers.add(writer);
    }
    protected void notifyObservers(String string) {
        for (Observer<String> writer : consoleWriterObserve
            writer.update(string);
    }
}
```

```
public class Observable<T>{
    private List<Observer<T>> observers;
    public Observable() {
        this.observers = new ArrayList<>();
    }
    public void addObserver(Observer<T> writer) {
        this.observers.add(writer);
    }
    protected void notifyObservers(T t) {
        for (Observer<T> writer : observers)
            writer.update(t);
    }
}
```

```java
public class Subject <T> extends Observable<T>{
    public void set(T t){notifyObservers(t);}
}
```

```java
public class Observable<T>{
    private List<Observer<? super T>> observers;
    public Observable() {
        this.observers = new ArrayList<>();
    }
    public void addObserver(Observer<? super T> writer) {
        this.observers.add(writer);
    }
    protected void notifyObservers(T t) {
        for (Observer<? super T> writer : observers)
            writer.update(t);
    }
}
```

```
public class LetterCount  extends Observable<Integer> imple
    @Override
    public void update(String s) {
        notifyObservers(s.length());
    }
}
```