

# JavaFX

Modeli podataka, Observer obrazac

# ListView komponenta

- Generička komponenta **ListView<T>**
- Inicijalno prazna
- Postavljanje internog modela podataka korišćenjem metode **setItems** ili kroz konstruktor
- Mogućnost renderovanja svake ćelije na specifičan način metodom **setCellFactory**

[http://docs.oracle.com/javafx/2/ui\\_controls/list-view.htm](http://docs.oracle.com/javafx/2/ui_controls/list-view.htm)

# ListView komponenta

```
ListView<String> list = new ListView<String>();
```

```
ObservableList<String> names =  
FXCollections.observableArrayList();
```

```
names.addAll(Arrays.asList("Petar", "Ana",  
"Mira", "Marko", "Jovana"));
```

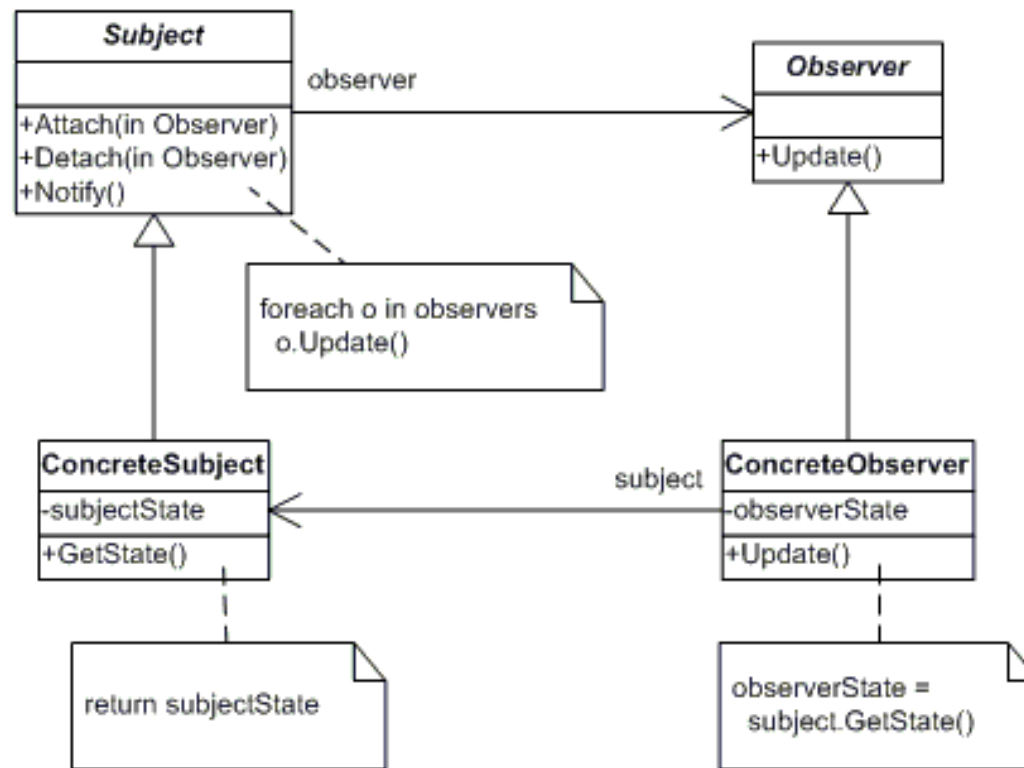
```
list.setItems(names);
```

# Pretplaćivanje na događaje

## Observer(s) - Observable

- Newsletter, grupe
- Pretplata na časopis
- Obaveštenja javnih službi
- Sudski pozivi itd...

# UML model Observera



# Primene u programiranju

1. MVC je baziran na Observeru – Smalltalk
2. Document View Architecture – skoro ekvivalentno MVC-u (Ažuriranje promena u različitim pogledima istog dokumenta)

# Observer u javi

- Observer interfejs
  - update
- Observable klasa
  - addObserver
  - notifyObservers

# Zadatak 1

- Ekran treba da sadrži dve liste u kojima su zapisana imena i godišta nekoliko osoba
- Između listi treba da stoje dva dugmeta označena sa “>>” i “<<”
- Svaki pojedinačni unos u listi je objekat klase Person koja je opisana imenom i godinom rođenja
- Inicijalno kreirati po tri elementa u levoj i desnoj listi
- Odabirom elementa iz leve liste i klikom na “>>”, taj element je potrebno prebaciti na kraj desne liste (analogno i za dugme “<<”)



# Dodatno (za domaći)

- Omogućiti da svaka ćelija bude obojena nekom slučajno odabranom bojom
- Za ovo koristiti metodu `setCellFactory`