



# Paradigme de programare (în Java)

## Lab 8/Curs 8- Programare reactivă

Col(r) Traian Nicula

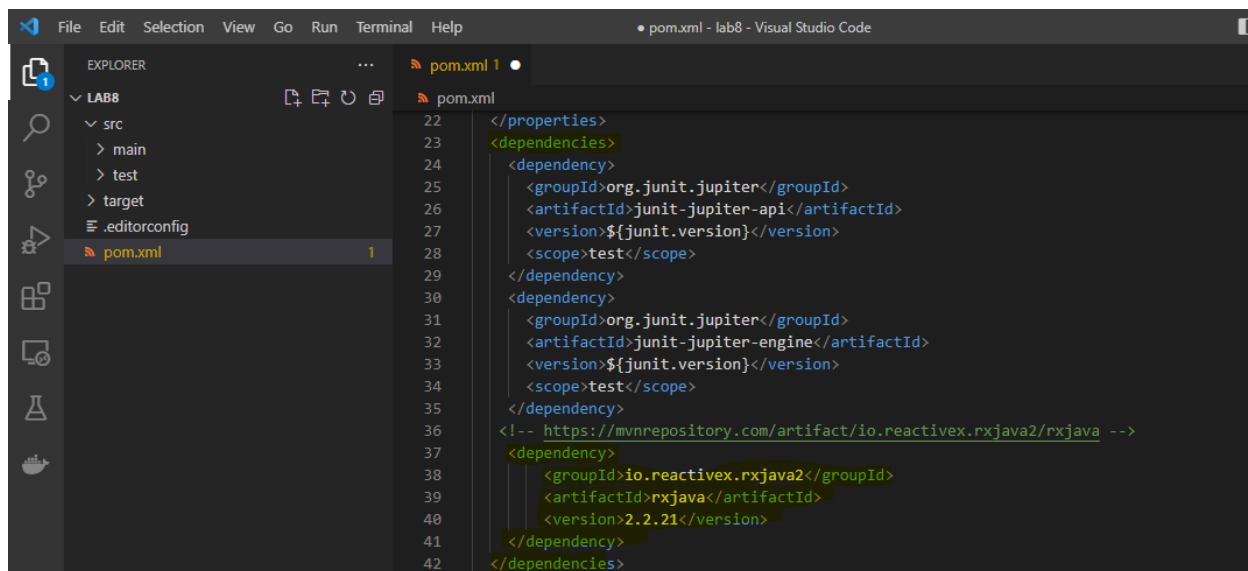
Se creează un proiect Maven cu numele **lab8**, folosind ca artefact **archetype-quickstart-jdk8** și se salvează în folder-ul cu numele studentului. Pentru fiecare exercițiu se va crea câte o clasă de test (Exercise1 etc.) cu metoda statică *main*, precum și alte clase cerute de exerciții.

Pentru a adăuga dependențele **RxJava 2**:

- se deschide fișierul **pom.xml** aparținând proiectului
- se copiază sub tag-ul `<dependencies>` fragmentul xml de mai jos și se salvează fișierul

```
<dependency>
  <groupId>io.reactivex.rxjava2</groupId>
  <artifactId>rxjava</artifactId>
  <version>2.2.21</version>
</dependency>
```

- fragmentul xml se găsește pe site-ul **MVN Repository** la link-ul <https://mvnrepository.com/artifact/io.reactivex.rxjava2/rxjava>





Se rezolvă următoarele exerciții:

1. Scrieți un program Java care: **(1p)**
  - creează un *Observable* de tip *String* care folosește *Observable.create* pentru a emite explicit 7 evenimente
  - subscrie la sursă cu o expresie lambda care tipărește rezultatul.
2. Scrieți un program Java care: **(1p)**
  - creează un *Observable* de tip *String* care folosește *Observable.create* pentru a emite explicit 7 evenimente și eventualele erori
  - folosește operatorii corespunzători pentru a transforma emisiile din *String* în întregi (lungime *String*) și a filtra cuvintele cu mai mult de 4 caractere
  - subscrie la sursă cu o expresie lambda care tipărește rezultatul.
3. Scrieți un program Java care: **(1p)**
  - creează un *Observable* de tip *String* cu metoda *Observable.just* care conține 8 cuvinte dintre care unele încep cu literele **a** și **A**
  - filtrează emisiile care încep cu **a** sau **A**
  - subscrie la sursă cu o expresie lambda care tipărește rezultatul.
4. Scrieți un program Java care: **(1p)**
  - creează un *Observable* de tip *String* cu metoda *Observable.just* care conține 8 cuvinte
  - filtrează cuvintele cu mai mult de 3 caractere
  - subscrie la sursă cu: o expresie lambda care tipărește rezultatul, afișează erorile (dacă sunt) și afișează un mesaj la terminarea stream-ului.
5. Scrieți un program Java care: **(1p)**
  - creează un *Observable* de tip *Long* cu metoda *Observable.interval*, care emite la o secundă
  - subscrie 2 *Observer*-i la sursă, care folosind expresii lambda, tipăresc emisia primită și un mesaj de identificare (ex. *Observer 1: 5*)
  - cei 2 *Observer*-i primesc simultan emisiile, primul pornește imediat, celălalt după 3 secunde.
6. Scrieți un program Java care: **(1p)**
  - creează o listă cu 10 numere întregi, inclusiv 0
  - creează un *Observable* pe baza listei
  - transformă numerele la inversul lor ( $1/x$ )
  - subscrie la sursă cu: o expresie lambda care tipărește rezultatul, afișează erorile (dacă sunt) și afișează un mesaj la terminarea stream-ului.
7. Scrieți un program Java care: **(1p)**
  - creează un *Observable* de tip *String* care emite o singură dată (*Single*) un cuvânt
  - convertește cuvântul la litere mari
  - subscrie la sursă cu o expresie lambda care tipărește rezultatul.
8. Scrieți un program Java care: **(1p)**
  - creează un *Observable* cu metoda *Observable.range* în intervalul 1 .. 100
  - lasă să treacă doar emisiile care respectă condiția  $i < 10$  (operatorul *takeWhile*)
  - subscrie la sursă cu o expresie lambda care tipărește rezultatul.
9. Scrieți un program Java care: **(1p)**
  - creează un *Observable* de tip *String* cu metoda *Observable.just* care conține 8 cuvinte (nici unul nu începe cu litera **Z**)



- filtrează doar cuvintele care încep cu litera **Z**
  - comută la alt *Observable* dacă cel inițial este nul (operatorul `switchIfEmpty`). Este creat tot cu *Observable.just* și conține 3 cuvinte
  - subscrie la sursă cu o expresie lambda care tipărește rezultatul.
10. Scrieți un program Java care: **(1p)**
- creează un *Observable* cu metoda *Observable.range* în intervalul 1 .. 100
  - sare peste primele 80 de emisii
  - subscrie la sursă cu o expresie lambda care tipărește rezultatul.

#### Temă pentru acasă:

11. Scrieți un program Java care: **(1p)**
- creează o listă cu 8 cuvinte
  - creează un *Observable* pe baza listei
  - folosește operatorul *zipWith* (similar *zip*) pentru a uni într-un String cuvintele din listă și valorile generate de *Observable.range(1, 100)*, folosind *String.format* cu formatul "%2d. %s"
  - subscrie la sursă cu o expresie lambda care tipărește rezultatul.
12. Scrieți un program Java care: **(1p)**
- creează o listă cu 8 cuvinte
  - creează un *Observable* pe baza listei
  - transformă cuvintele în caractere folosind: *flatMap*, *Observable.fromIterable*, metoda *split("")* a clasei String
  - alege doar caracterele distincte
  - sortează caracterele în ordine naturală
  - folosește operatorul *zipWith* (similar *zip*) pentru a uni caracterele distincte și valorile generate de *Observable.range(1, 100)*, folosind *String.format* cu formatul "%2d. %s"
  - subscrie la sursă cu o expresie lambda care tipărește rezultatul.