**Academia Tehnică Militară "Ferdinand I“**
Facultatea de Sisteme Informatice şi Securitate Cibernetică

# Paradigme de programare (în Java)

## Lab 5/Curs 5- Programare funcțională

Col(r) Traian Nicula

Se creează un proiect Maven cu numele ***lab5***, 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.

### Se rezolvă următoarele exerciții:

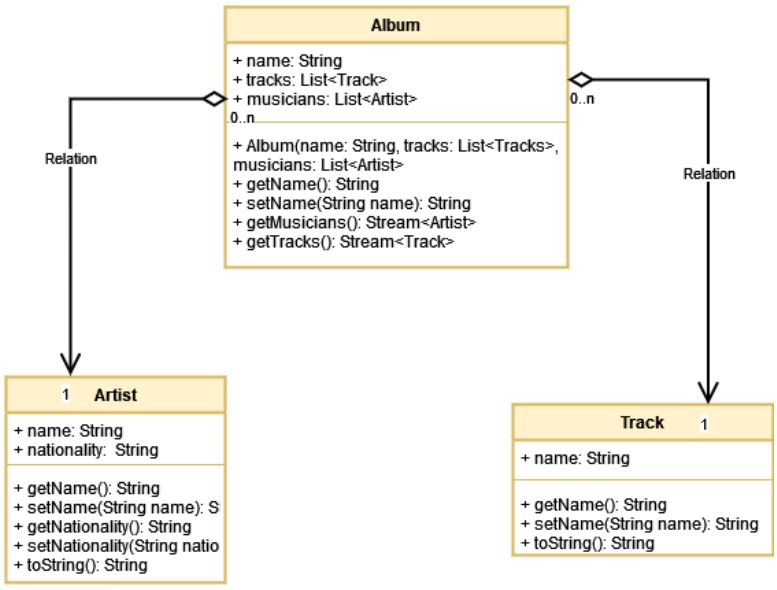
1. Scrieți un program Java care: (**1p**)
   - creează o interfață funcțională cu denumirea ***MathOperation*** care conține metoda *int operation(int a, int b)*
   - creează variabile de tip ***MathOperation*** cărora li se atribuie expresiile lambda corespunzătoare operațiilor: *adunare*, *scădere*, *înmulțire*, *împărțire* și *restul împărțirii*
   - creează metoda statică *int calculate(int a, int b, MathOperation m)* care calculează răspunsul în funcție de operație
   - testează toate operațiile matematice definite și tipărește numerele, operația și rezultatul.
2. Scrieți un program Java care: (**1p**)
   - Creează o interfață funcțională cu denumirea ***Greetings*** care conține metoda *void say(String message)*
   - creează variabile de tip ***Greetings*** cărora li se atribuie expresiile lambda pentru tipărirea textului "Java programming rules":
     a. așa cum este, folosind referință de metodă la *System.out.println*
     b. prin concatenarea textului MESSAGE: cu mesajul convertit la litere mari
     c. prin concatenarea textului Message: cu mesajul convertit la litere mici
   - folosește variabilele create pentru tipărirea mesajului corespunzător.
3. Scrieți un program Java care: (**1.5p**)
   - creează o listă ce conține String-urile: "Java", "Scala", "C++", "Haskell", "Lisp"
   - creează o metodă statică *void filter(List<String> names, Predicate<String> condition)* care parcurge lista, testează condiția și tipărește elementele din listă care o satisfac
   - folosește metoda *filter* cu diferite expresii lambda pentru a afișa:
     a. cuvintele care încep cu J
     b. cuvintele care se termină cu litera a
     c. toate cuvintele
     d. nici un cuvânt.
4. Scrieți un program Java care: (**1p**)
   - creează o listă de String-uri dintre care unele sunt goale (“”)
   - folosește Stream API pentru a număra String-urile nenule
   - tipărește rezultatul.
5. Scrieți un program Java care: (**1p**)

- creează o listă cu 8 String-uri având 1, 2, 3 și mai multe caractere
- folosește Stream API pentru a colecta cuvintele cu mai mult de 2 caractere
- tipărește rezultatul.

6. Scrieți un program Java care: (**1p**)
   - creează o listă de String-uri nenule conținând litere mari și mici
   - folosește Stream API pentru a transforma toate cuvintele în litere mari și a le colecta
   - tipărește rezultatul.
7. Scrieți un program Java care: (**1p**)
   - creează o listă de numere întregi, dintre care unele duplicate
   - folosește Stream API pentru a calcula suma pătratelor numerelor distincte (se folosește stream specializat pe primitive)
   - tipărește rezultatul.
8. Scrieți un program Java care: (**1.5p**)
   - creează un stream cu metoda *Stream.of()* care primește ca parametrii 3 liste de numere întregi: (1, 2), (4, 6), (8, 9)
   - folosește Stream API pentru a transforma stream-ul de liste în stream de întregi (folosește metoda *stream()* a listei)
   - convertește stream-ul într-o listă care conține doar 4 elemente și o afișează.
9. Scrieți un program Java care: (**1p**)
   - creează un stream de numere pare folosind *Stream.iterate()*
   - limitează stream-ul la 20 de elemente
   - calculează suma folosind operația *reduce*
   - tipărește rezultatul.

## Temă pentru acasă:

10. Scrieți un program Java care: (**0.5p**)
    - creează un stream de 8 cuvinte
    - transformă cuvintele la litere mari
    - filtrează cuvintele care încep ***A*** sau cu ***B***
    - afișează pe primul găsit sau cuvântul ***none*** dacă nu există.
11. Scrieți un program Java care: (**0.5p**)
    - creează un stream de 8 cuvinte
    - elimină cuvintele duplicate
    - afișează un mesaj dacă există cel puțin un cuvânt care conține mai mult de 4 caractere.
12. Scrieți un program Java care: (**1p**)
    - creează clasele din diagrama UML de mai jos




- creează o instanță a clasei Album care conține o listă de instanțe ale clasei ***Track*** și o listă de instanțe ale clasei ***Artist***
- obține și afișează în majuscule lista cântecelor (tracks) și numărul de cântece
- obține și afișează lista artiștilor sortați după nume.