

## Programare avansata pe obiecte – laborator 4

*Alina Puscasu [alina.puscasu@endava.com](mailto:alina.puscasu@endava.com)*

### Imutabilitate

- Odata ce un obiect este creat, nu ii mai putem schimba continutul - Toate clasele wrapper sunt imutabile, String este imutabil!
- Reguli de urmat pentru a obtine asta:
  1. Clasa trebuie sa fie final, pentru a nu mai fi extinsa
  2. Membrii clasei sunt final (initializati doar in constructor)
  3. Constructor cu parametrii folosit pentru initializarea membrilor clasei
  4. Getteri pentru toate field-urile
  5. Fara setter (ca sa nu oferim posibilitatea schimbarii field-urilor)
  6. In orice metoda a clasei care modifica starea obiectelor, trebuie returnata o noua instanta a acestora (inclusiv in constructor)

### String

- Immutable
- De fiecare data cand facem o schimbare, un alt String este creat
- Are implementat toString (afiseaza valoarea stringului) si equals (verifica egalitatea logica dpdv al continutului celor 2 Stringuri comparate)

### StringBuilder si StringBuffer

- Ambele sunt mutabile, operatiile pe StringBuilder si StringBuffer nu creeaza obiecte noi, modifica obiectul initial
- Alternativa thread safe este StringBuffer dar nu prea este folosit pentru ca este mai slow
- .toString() aplicat pe StringBuilder intoarce un alt obiect de tip String!
- Nu au implementat equals(), verifica la fel precum o face == , adica zona de memorie

### Metode String vs StringBuilder

- String: <https://docs.oracle.com/javase/7/docs/api/java/lang/String.html>
- StringBuilder: <https://docs.oracle.com/javase/7/docs/api/java/lang/StringBuilder.html>

## Exercitii

1. Se citeste un sir de caractere de la tastatura, verificati daca este un palindrom.
2. Scrieti un program care verifica daca doua siruri de caractere sunt anagrame. (ex: ramo, mora si roma sunt anagrame)
3. Sa se implementeze o clasa PasswordMaker ce genereaza o parola pornind de la datele unei persoane. Aceasta clasa o sa contina urmatoarele:
  - a. o constanta MAGIC\_NUMBER, care ia orice valoare doriti
  - b. un String constant MAGIC\_STRING, lung de minim 20 caractere, generat random (puteti crea o metoda pentru asta – Stringul este alcatuit din caractere 😊)
  - c. un constructor care primeste un String, numit name
  - d. o metoda getPassword() care va returna parola  
Parola se construiește concatenand urmatoarele:
    - un sir random de lungime MAGIC\_NUMBER
    - 10 caractere din MAGIC\_STRING
    - lungimea atributului name ca si String
    - un numar intreg generat random din intervalul [0,100] folosind clasa Random ( <https://docs.oracle.com/javase/8/docs/api/java/util/Random.html> )

Modificati clasa PasswordMaker astfel incat sa respecte conceptul de Singleton.