

## Enunț laborator – *DuckSocialNetwork*

### Tema: Rețea de socializare cu utilizatori umani și rațe

Implementați o **rețea de socializare** numită DuckSocialNetwork, care permite interacțiunea dintre **persoane** și **rațe**.

Aplicația trebuie să ofere funcționalități de bază specifice unei rețele sociale, dar și extensii tematice inspirate din „Problema Natație” .

---

#### Ierarhia de clase

##### User (*abstractă*)

Reprezintă entitatea de bază din rețea.

##### Atribute:

- id: Long
- username: String
- email: String
- password: String

##### Comportament:

- metode generale: login(), logout(), sendMessage(), receiveMessage()
  - lista de prieteni, istoricul evenimentelor etc.
- 

##### Persoana extends User

Reprezintă un utilizator uman.

##### Atribute:

- nume, prenume, dataNasterii, ocupatie
- nivelEmpatie – influențează interacțiunea cu rațele 😊

##### Comportament:

- poate crea și administra evenimente (ex: „Duck Race”)
  - poate trimite mesaje altor utilizatori (oameni sau rațe)
- 

##### Duck extends User

Reprezintă un utilizator de tip rață. Rațele pot fi de mai multe tipuri și se pot organiza în carduri.

##### Atribute specifice:

- tip: TipRata ∈ { FLYING, SWIMMING, FLYING\_AND\_SWIMMING }

- viteză: double
- rezistență: double
- card: Card (grupul din care face parte rața)

#### Comportament:

- participă la evenimente (curse, zboruri etc.)
- poate trimite mesaje automate („Quack! Am terminat antrenamentul!”)
- poate fi observator sau participant într-un eveniment

### Alte entități importante

#### Cârd

- grup de rațe cu scop comun (ex: *SwimMasters*, *SkyFlyers*)
- id, numeCârd, membri: List<Duck>
- getPerformantaMedie() – media vitezelor și rezistențelor

#### Friendship

- relație bidirecțională între doi User (om–om, om–rață, rață–rață)

#### Message

- id, sender, receiver, content, timestamp

#### Event

- eveniment general din rețea (poate fi observat de utilizatori)
- subscribers: List<User>
- metode: subscribe(), unsubscribe(), notifySubscribers() (**Observer Pattern**)

#### RaceEvent extends Event

- eveniment special pentru rațe înotătoare
- selectează automat M rațe conform regulilor problemei „Natație”:
  - rața i are viteză  $v_i$ , rezistență  $r_i$ ;
  - balizele au distanțe  $d_j$ ;
  - o cursă este validă dacă ordinea rezistențelor respectă  $r_1 \geq r_2 \geq \dots \geq r_m$ ;
  - timpul cursei =  $\max(2 \cdot d_j / v_i)$  pentru toți participanții.

### Funcționalități de bază

#### 1. Gestiunea utilizatorilor (CRUD)

- creare / modificare / ștergere / listare utilizatori
- încărcare inițială din fișiere .csv

## 2. Prietenii

- adăugare / ștergere prieten
- căutare utilizatori după nume

## 3. Mesagerie

- trimitere / ștergere mesaje
- istoric mesaje per utilizator

## 4. Evenimente (Observer Pattern)

- crearea evenimentelor (curse, zboruri, întâlniri)
- abonare / dezabonare utilizatori
- notificări automate

## 5. Carduri de rațe

- creare / listare carduri
- adăugare / eliminare rațe dintr-un card
- afișare performanță medie

## 6. Simulare de cursă („Natație”)

- se definesc N rațe și M balize
- se aleg M rațe potrivite conform criteriilor
- se calculează durata minimă totală a cursei
- se afișează rezultatul:
- Duck 3 on lane 3:  $t = 4.000$  s
- Duck 2 on lane 2:  $t = 2.400$  s
- Duck 1 on lane 1:  $t = 1.500$  s

## 7. Autentificare și jurnalizare

- logare / delogare utilizatori
- vizualizare istoric evenimente

---

## Cerințe non-funcționale

- Proiectare UML în **StarUML**
- Implementare în **Java**, respectând principiile **SOLID**

- Utilizarea șabloanelor de proiectare
    - **Factory** (creare utilizatori sau evenimente)
    - **Singleton** (gestiunea globală a rețelei)
    - **Observer** (abonare la evenimente)
    - **Decorator** (extinderea comportamentului runner-ului de taskuri)
    - **CE VOM MAI ÎNVĂȚA PE PARCURS**
- 

### Extensii posibile

- Rațe care pot zbura în formație (strategie colectivă – *Strategy Pattern*).
- Sistem de ranking între rațe în funcție de numărul de curse câștigate.
- Integrare cu sistemul TaskRunner (de la seminar):
  - RaceTask pentru simularea unei curse;
  - PrinterTaskRunner pentru raportarea timpului cursei.