

Metode Avansate de Programare – Laborator 3

Tema: DuckSocialNetwork

DEADLINE: Săptămâna 5

Link GIT: <https://classroom.github.com/a/esVzxmy>

Cerințe funcționale

(– vezi fișierul ENUNT_Lab_DuckSocialNetwork.docx)

- Add/Remove Utilizator (2p)

Implementați operațiile de adăugare și ștergere de utilizatori din rețea. Un utilizator poate fi de două tipuri: Persoană – utilizator uman (cu nume, prenume, ocupație etc.) sau Duck – utilizator de tip rață (cu tip: FLYING, SWIMMING sau FLYING_AND_SWIMMING, viteză și rezistență). La adăugarea unui utilizator, se va specifica tipul și atributele corespunzătoare.

- Add/Remove Prieten (2p)

Se pot adăuga sau elimina relații de prietenie între doi utilizatori (indiferent dacă sunt persoane sau rațe). Relația de prietenie este bidirecțională.

- Afișare număr de comunități (0.5p)

Determinați numărul de comunități (componente conexe) din graful rețelei sociale.

- Afișare cea mai sociabilă comunitate (0.5p)

Afișați comunitatea (componenta conexă) care are cel mai lung drum (diametru maxim) între utilizatori.

Cerințe non-funcționale

(– vezi seminarul 4)

Cerință	Punctaj
Arhitectură stratificată	1.5p
DDD (Domain Driven Design)	0.5p
Documentația – JavaDoc (Tools → Generate JavaDoc în IntelliJ IDEA)	0.5p
Persistența datelor – în memorie sau în fișiere	1p

Validarea datelor – folosind Strategy Pattern	0.5p
Definirea propriilor clase de excepții	0.25p
Interfață cu utilizatorul de tip consolă (minimalistă)	0.5p
Total: 8p	