

Metode Avansate de Programare – Laborator 3

Tema: DuckSocialNetwork

DEADLINE: Săptămâna 6

Cerințe funcționale

Continuați modelarea și proiectarea problemei duck_socialnetwork și implementați operația de adăugare Cârd și Event - **RaceEvent**

Cârd

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

Sugestii de implementare:

```
public interface Zburator { void zboara(); }
```

```
public interface Inotator { void inoata(); }
```

```
clasa Duck abstracta
```

```
public class SwimmingDuck extends Duck implements Inotator
```

```
public class FlyingDuck extends Duck implements Zburator
```

```
public abstract class Cârd<T extends Duck> ...
```

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”:

Cerințe non-funcționale

Aceleași cu cele de la lab precedent