

Traccia di Object Orientation – 12 Ottobre 2020

Esercizio 1

Si realizzi un sequence diagram corrispondente all'invocazione del metodo `crawl` riportato di seguito. Si assuma che tutti gli import necessari siano definiti correttamente.

```
class Crawler {
    Browser browser = new FakeBrowser();
    Database database = new Database();

    public void crawl(String url){
        while(url != null){
            WebPage page = browser.visit(url);
            String nextUrl = page.getNextUrl();
            database.save(page);
        }
    }
}
```

Esercizio 2

Un gruppo di giocatori di poker vi commissiona la realizzazione di un sistema informatico per tenere traccia dei punteggi delle loro sfide. In particolare, il sistema permette a un giocatore di creare una nuova partita, specificando un luogo tra quelli presenti nel sistema, una data e un orario, una quota di partecipazione (espressa in €), e un elenco di almeno due e al più dieci giocatori partecipanti. Successivamente, il giocatore che ha creato la partita può specificare quale dei partecipanti ha vinto. Infine, tutti i giocatori possono visualizzare una schermata di riepilogo che mostra, per ogni giocatore, numero di partite giocate e numero di vittorie.

Definire un Class Diagram per modellare la creazione di una partita, inteso come modello di dominio.

Esercizio 3

Un gruppo di giocatori di poker vi commissiona la realizzazione di un sistema informatico per tenere traccia dei punteggi delle loro sfide. In particolare, il sistema permette a un giocatore di creare una nuova partita, specificando un luogo tra quelli presenti nel sistema, una data e un orario, una quota di partecipazione (espressa in €), e un elenco di almeno due e al più dieci giocatori partecipanti. Successivamente, il giocatore che ha creato la partita può specificare quale dei partecipanti ha vinto. Infine, tutti i giocatori possono visualizzare una schermata di riepilogo che mostra, per ogni giocatore, numero di partite giocate e numero di vittorie.

In relazione all'esercizio precedente, fornire i Mock-up e un Sequence Diagram di analisi dello scenario relativo alla creazione di una partita.

Esercizio 4

```
class Foo {  
    void doSomething(Object o) {  
        try {  
            System.out.println("Before");  
            o.toString();  
            System.out.println("After");  
        }  
        catch(Exception e) {  
            System.out.println("Exception");  
        }  
        finally {  
            System.out.println("Done");  
        }  
    }  
}
```

Si consideri il codice Java sopra riportato. Assumendo che tutte le import siano definite correttamente, il codice compila? Se no, indicare come modificare il codice affinché la compilazione abbia successo. Qual è l'output su stdout di un'invocazione del metodo `doSomething` passando come argument `null`? Se il risultato dell'esecuzione del metodo è il lancio di un'eccezione, indicare di quale eccezione si tratta.

Esercizio 5

```
abstract class Car {
    public int maxSpeed;
    public int acceleration;
    public int range;
    abstract void revEngine();
}

class F1 extends Car {
    void revEngine() {System.out.println("Mweeeee");}
}

class ElectricCar extends Car {}

class RaceSimulator {
    //Ritorna l'automobile vincente tra "a" e "b"
    public F1 simulateRace(F1 a, F1 b){
        F1 winner=null;
        //codice di simulazione gara omissa
        return winner;
    }
}
```

Si consideri il codice Java sopra riportato.

- Il codice compila correttamente? Se no, indicare come modificare il codice affinché la compilazione abbia successo.
- Come modificherebbe la classe RaceSimulator per introdurre la possibilità di simulare corse tra veicoli elettrici? E per simulare anche corse “miste”, cioè tra veicoli generici, come ad esempio un veicolo elettrico e una monoposto di Formula 1?
- Si consideri il codice ottenuto dopo le modifiche del punto precedente. È presente un esempio di overriding? È presente un esempio di overloading? Per entrambe le precedenti domande, in caso di risposta positiva indicare in quale/quali parti di codice. In caso di risposta negativa, indicare – se possibile – come si potrebbe modificare ulteriormente il codice per introdurre un esempio.