

Unina Maps è un servizio che consente la ricerca e la visualizzazione di mappe geografiche di buona parte della Terra. Il servizio offre molteplici funzionalità. Un utente generico può inserire in un apposito form un indirizzo, per vederlo visualizzato su mappa (georeferenziazione). Tale utente può quindi modificare il livello di zoom, così come scorrere la mappa sull'asse X e Y. Inoltre è possibile ricercare Punti d'Interesse (PDI), tra cui ristoranti, monumenti, negozi, hotel, etc. Un utente registrato (cioè che ha effettuato un login) può inoltre, dopo aver georeferenziato un indirizzo, anche salvarlo in un suo elenco di PDI preferiti. Infine, un amministratore del sito ha la possibilità di aggiornare periodicamente le fotografie satellitari presenti nel sistema, specificando per ognuna di esse la latitudine, longitudine ed estensione.

Definire i Mock-up e un Class Diagram per modellare il salvataggio di un PDI. Il Class Diagram, inteso come modello di dominio, si deve rifare all'Entity-Boundary-Control (EBC).

Unina Maps è un servizio che consente la ricerca e la visualizzazione di mappe geografiche di buona parte della Terra. Il servizio offre molteplici funzionalità. Un utente generico può inserire in un apposito form un indirizzo, per vederlo visualizzato su mappa (georeferenziazione). Tale utente può quindi modificare il livello di zoom, così come scorrere la mappa sull'asse X e Y. Inoltre è possibile ricercare Punti d'Interesse (PDI), tra cui ristoranti, monumenti, negozi, hotel, etc. Un utente registrato (cioè che ha effettuato un login) può inoltre, dopo aver georeferenziato un indirizzo, anche salvarlo in un suo elenco di PDI preferiti. Infine, un amministratore del sito ha la possibilità di aggiornare periodicamente le fotografie satellitari presenti nel sistema, specificando per ognuna di esse la latitudine, longitudine ed estensione.

Fornire un Sequence Diagram di analisi dello scenario relativo salvataggio di un PDI. Il Sequence Diagram deve sfruttare le classi definite nel Class Diagram del punto precedente (che può essere parzialmente esteso in questo esercizio). **Si richiede di caricare, come svolgimento di questo esercizio, anche una fotografia dei Mock-up e del Class Diagram realizzati al punto precedente.**

```

public class Main {
    public static void main(String[] args) {
        GuessNumber g = new GuessNumber();
        try {
            System.out.println(g.guess(123));
            System.out.println(g.guess(-99));
            System.out.println(g.guess(42));
        } catch (InvalidNumberException e) {
            System.out.println(e.msg);
        }
    }
}

class GuessNumber {
    private int number;
    public GuessNumber() {
        this.number = 42; //secret, don't tell!
    }
    public boolean guess(int n) {
        if(n<0) {
            throw new InvalidNumberException("!!!");
        }
        return n == this.number;
    }
}

class InvalidNumberException extends Exception {
    private String msg;
    public InvalidNumberException(String msg){
        this.msg = msg;
    }
}

```

Si consideri il codice Java sopra riportato.

1. Il codice compila correttamente? Se la risposta è no, indicare quali sono gli errori riscontrati e come è possibile correggerli.
2. Qual è l'output su stdout (tenendo conto delle eventuali correzioni apportate nel punto precedente) di un'invocazione del metodo *main*?
3. Come modificherebbe il codice sopra riportato per gestire anche indovinelli con String?

Dato il seguente codice Java, fornire un Sequence Diagram rappresentante un'invocazione del metodo *foo()* della classe A.

```
class B {  
    public void function() {  
        this.compute();  
    }  
    private void compute() {  
        /*...*/  
    }  
    /*...*/  
}
```

```
class A {  
    private B myB = new B();  
    public void setB(B b) {  
        this.myB = b;  
    }  
    public boolean foo(){  
        for (int i =0; i<10; i++)  
            myB.function();  
        return true;  
    }  
    /*...*/  
}
```