

Si definisca il class diagram relativo al seguente frammento di codice Java:

```
class Esame {
    private List<Domanda> domande;
    /*...*/
    public double getVoto(List<Risposta> r) {
        int corrette = 0;
        for(int i; i < r.size(); i++){
            if(r.get(i).equals(domande.get(i).getRispostaCorretta())){
                corrette += 1;
            }
        }
        return ( corrette/domande.size() ) * 100;
    }
}
```

Si vuole realizzare un sistema per l'acquisto on-line di biglietti per cinema. In particolare, un cliente potrà acquistare un biglietto per vedere un particolare film specificando lo spettacolo desiderato (se ci sono ancora posti disponibili).

Uno spettacolo è caratterizzato da una data, un orario, e una sala.

Realizzare un class diagram di dominio dello scenario descritto.

In relazione all'esercizio precedente, Scrivere il sequence diagram dell'acquisto di un biglietto.

```
Class Pet {
    public String name;
    public String getName(){return name;};
    public void faiVerso(){System.out.println("??");};
}
Class Cat extends Pet {}
Class Dog extends Pet {}
Class Bunny extends Pet {}
Class Person {
    public void playWithPet(Pet p){}
    public void playWithPet(Dog d){}
}
```

Si consideri il frammento di codice Java sopra riportato.

1. Rispetta il principio dell'incapsulamento? Motivare la risposta. Se no, come modificherebbe il codice per rispettarlo?

```

Class Pet {
    public String name;
    public String getName() {return name;};
    public void faiVerso() {System.out.println("???");};
}
Class Cat extends Pet {}
Class Dog extends Pet {}
Class Bunny extends Pet {}
Class Person {
    public void playWithPet(Pet p){}
    public void playWithPet(Dog d){}
}

```

Si consideri il frammento di codice Java sopra riportato.

1. È presente un esempio di overriding? Motivare la risposta. Se no, indicare come modificare il codice per includere un esempio di overriding.

```

Class Pet {
    public String name;
    public String getName() {return name;};
    public void faiVerso() {System.out.println("???");};
}
Class Cat extends Pet {}
Class Dog extends Pet {}
Class Bunny extends Pet {}
Class Person {
    public void playWithPet(Pet p){}
    public void playWithPet(Dog d){}
}

```

Si consideri il frammento di codice Java sopra riportato.

1. È presente un esempio di overloading? Motivare la risposta. In caso fosse possibile, come modificherebbe il codice per aggiungere un esempio di overloading?