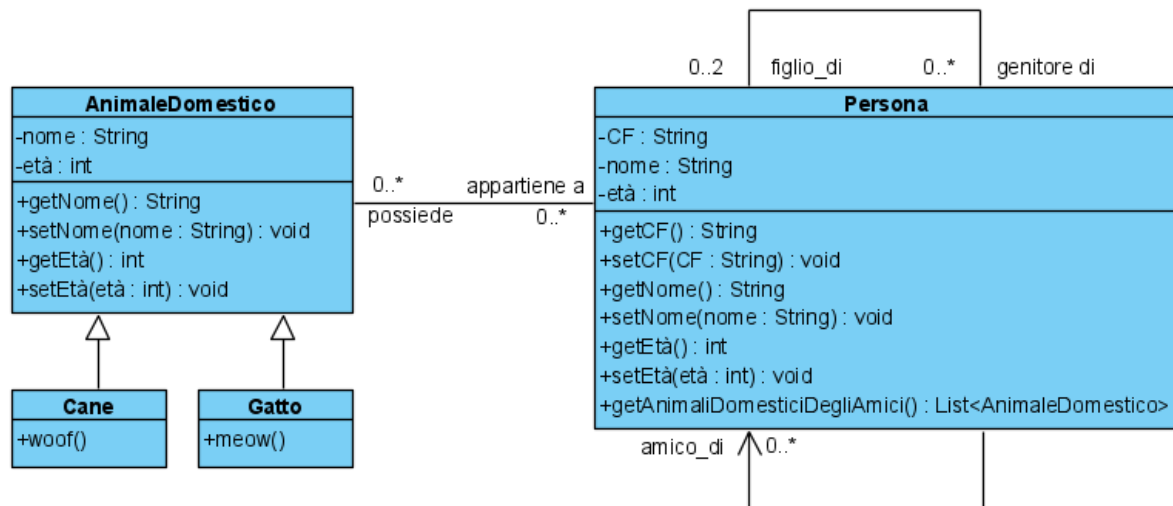


# Esercizi di Object Orientation

## Esercizio 1

Si scriva tutto il codice Java che è possibile desumere dal seguente class diagram.



## Esercizio 2

Si realizzi un class diagram per il seguente frammento di codice.

```
import java.util.Date;
import java.util.List;

public class Post {
    String titolo;
    String contenuto;
    Date data;

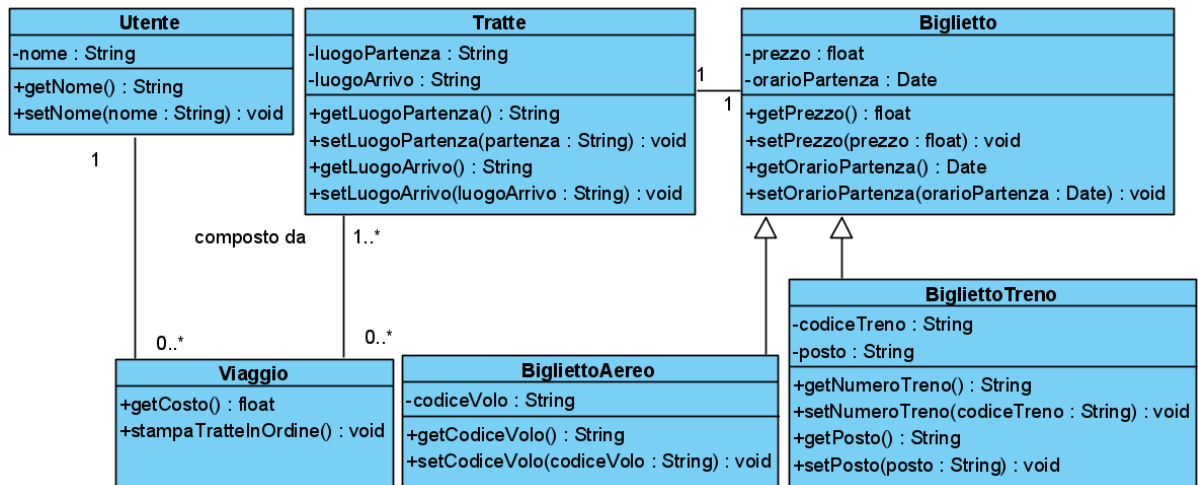
    Utente autore;
    List<Commento> commenti;
    List<Reazione> reazioni;

    /* Costruttori, getter e setter omissi per brevità */

    void stampaCommentatori() {
        for(Commento c : commenti) {
            System.out.println(c.getAutore().toString());
        }
    }
}
```

## Esercizio 3

Si scriva tutto il codice Java che è possibile desumere dal seguente class diagram.



## Esercizio 4

Si realizzi un class diagram per il seguente frammento di codice.

```
import java.util.List;

public class Automobile {
    String targa;
    String modello;
    Persona conducente;
    Persona[] passeggeri = new Persona[4];
    List<Revisione> revisioniEffettuate;

    /* Costruttore e getter e setter omessi */

    void stampaPasseggeri() {
        for(int i=0;i<passeggeri.length;i++) {
            if(passeggeri[i]!=null) {
                passeggeri[i].printName();
            }
        }
    }
}
```

## Esercizio 5

---

```
class EccezioneA extends Exception {}
class EccezioneB extends EccezioneA {}
class EccezioneC extends EccezioneB {}

public class Test {
    public void foo() {
        try{
            throw new EccezioneB();
        }
        catch(EccezioneC e) {
            System.out.println("Catch C");
        }
        catch(EccezioneA e) {
            System.out.println("Catch A");
        }
        catch(Exception e) {
            System.out.println("Catch Exception");
        }
    }
}
```

Si consideri il codice Java in figura.

- Il codice compila correttamente? Se no, indicare come sarebbe possibile modificare il codice affinché la compilazione vada a buon fine.
- Qual è l'output su stdout di un'invocazione del metodo `foo()` nella classe `Test`?

## Esercizio 6

---

```
class MiaEccezione extends Exception {}

public class Main {

    public void test() {
        try {
            System.out.println("Hello");
            randomMethod();
            System.out.println("World");
        }
        catch(MiaEccezione e) {
            System.out.println("MiaEccezione");
        }
    }

    public void randomMethod() {
        //Math.random() ritorna un double pseudo-casuale tra 0.0 e 1.0
        if(Math.random() > 0.5) {
            throw new MiaEccezione();
        } else {
            throw new Exception();
        }
    }
}
```

Si consideri il codice riportato in figura.

- Il codice compila correttamente? Se no, indicare come sarebbe possibile modificare il codice affinché la compilazione vada a buon fine.
- Qual è l'output su stdout di un'invocazione del metodo test() nella classe Main? Nel caso siano possibili più output, indicarli tutti.

## Esercizio 7

---

```
class EccezioneA extends Exception {
    String testo;
    public EccezioneA(String testo) {
        this.testo = testo;
    }
}

class EccezioneB extends EccezioneA {
    int n;
    public EccezioneB(int n) {
        this.n=n;
    }
}

public class Test {
    EccezioneA a;
    EccezioneB b;
    public void test() throws Exception {
        try {
            b = new EccezioneB(123);
            a = new EccezioneA("Ooops");
            throw a;
        } catch (EccezioneB b) {
            throw a;
        }
        catch (EccezioneA a) {
            throw b;
        }
    }
}
```

Si consideri il codice riportato in figura.

- Il codice compila correttamente? Se no, indicare come sarebbe possibile modificare il codice affinché la compilazione vada a buon fine.
- Qual è l'output su stdout di un'invocazione del metodo test() nella classe Test? Nel caso siano possibili più output, indicarli tutti.