

Esame di Object Orientation (Gr. 1) – Gennaio 2023

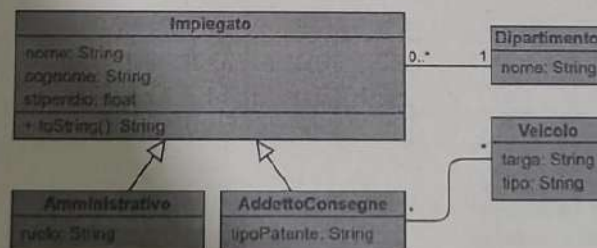
- Scrivere immediatamente su ogni foglio che vi è stato consegnato Cognome, Nome, N° Matricola.
- Non è consentito consultare appunti, libri, o colleghi, né qualunque dispositivo elettronico, PENA IMMEDIATO ANNULLAMENTO DELLA PROVA
- Tempo a disposizione: 3 ore

Esercizio 1

Vi è commissionata la realizzazione di un sistema per gestire i tirocini curriculari (sia interni che esterni) di studenti del **Corso di Laurea** in Informatica. Il sistema permette agli **studenti** di inserire una richiesta di **tirocinio**, indicando la **tipologia di tirocinio** (interno o esterno), una breve descrizione testuale della **tematica** trattata, una **data prevista di inizio e di fine**. Per **tirocini interni**, inoltre, lo studente deve indicare un **docente di riferimento** tra i **Professori del Corso di Laurea** in Informatica. Facoltativamente, lo studente **può indicare** anche al più due altri co-tutor, da selezionare tra i **Professori** e/o i **Ricercatori** del Corso di Laurea in Informatica. Per i tirocini esterni, invece, lo **studente** deve selezionare un'azienda tra quelle convenzionate con l'Università, e inserire **nome, cognome** ed **e-mail** del **referente aziendale** che seguirà il tirocinio. I **membri della commissione tirocini** del **Corso di Laurea** possono visualizzare le domande inviate dagli studenti e decidere se approvarle o rigettarle.

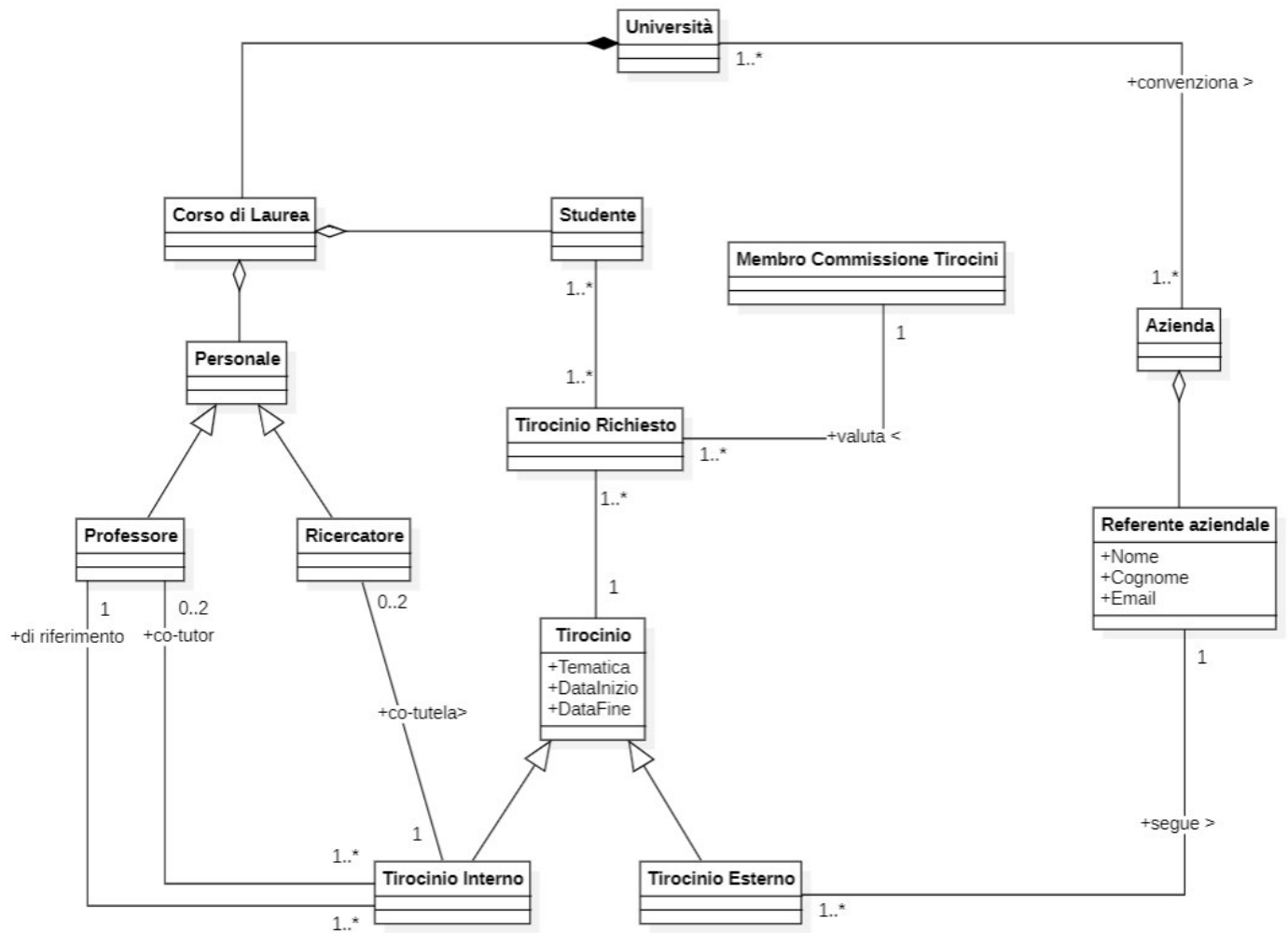
Si richiede di definire (a) Class Diagram e (b) Sequence Diagram per modellare la funzionalità di inserimento di una **richiesta di tirocinio**, eventualmente aiutandosi con delle rappresentazioni grafiche dell'interfaccia utente (mock-up). Il Class Diagram e il Sequence Diagram, intesi come modello di dominio, devono rifarsi all'euristica Entity-Boundary-Control (EBC).

Esercizio 2



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

ESERCIZIO 1



Esercizio 3

```

public class Main {
    public static void main(String[] args){
        Studente[] studenti = new Studente[5];
        studenti[0] = new Studente("Mario Rossi", "N86/1234");
        studenti[1] = new StudenteLavoratore("Sam Porter Bridges", "N86/5678", "Bridges");
        Corso c = new Corso(studenti);
        try {
            for(int i=0; i < studenti.length; i++){
                c.stampaStudente(i);
            }
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("404: Studente non trovato");
        } catch (Exception e){
            System.out.println("Whoops");
        }
    }
}

public class Corso {
    Studente[] studenti;
    public Corso(Studente[] studenti){
        this.studenti = studenti;
    }

    public void stampaStudente(int i) {
        System.out.println(i+": "+this.studenti[i].nome);
    }
}

public class Studente {
    String nome; String matricola;
    public Studente(String nome, String matricola){ this.nome=nome; this.matricola=matricola;}
}

public class StudenteLavoratore extends Studente {
    String azienda;
    public StudenteLavoratore(String nome, String matricola, String azienda){
        this.nome=nome; this.matricola=matricola; this.azienda=azienda;
    }
}

```

→ Costruttore sbagliato

Si considerino le classi Java riportate sopra (si assuma che ogni classe sia definita in un file apposito).

- Il codice compila correttamente? Motivare la risposta e fornire le possibili correzioni in caso di risposta negativa.
- Dopo aver apportato le eventuali correzioni di cui al punto (a), qual è il risultato di un'invocazione del metodo main della classe Main?
- Nel codice è presente un esempio di overriding? Se sì, indicare dove. Se no, indicare come è possibile modificare il codice per introdurre un esempio.
- Nel codice è presente un esempio di overloading? Se sì, indicare dove. Se no, indicare come è possibile modificare il codice per introdurre un esempio.

Esercizio 4

Si spieghi il meccanismo che permette, in Java Swing, di specificare una o più porzioni di codice da eseguire ogni volta che un utente clicca su un pulsante in un'interfaccia grafica.

→ ActionListener?

ESERCIZIO 3

a) Il codice non compila in quanto il costruttore della classe `StudenteLavoratore` è sbagliato. La versione corretta è:

```
public StudenteLavoratore (String nome, String matricola, String azienda ) {  
    super (nome, matricola);  
    this.azienda = azienda;  
}
```

b) Il risultato sarà:

0: Mario Rossi

1: Sam Porter Bridges

404: Studente non trovato

c) Non è presente un overriding. Per averlo ci dovrebbe essere la presenza di un'altra funzione `StampaStudente` con lo stesso parametro ma con un'implementazione differente.

d) Non è presente un overloading. Per averlo ci dovrebbe essere un'altra funzione `StampaStudente` che o ha lo stesso parametro e un valore di ritorno differente, o stesso valore di ritorno e l'aggiunta di un parametro o un altro tipo di parametro.