Corso di Laurea in Informatica
Prova Scritta di Metodologie di Programmazione - Primo Canale

Sapienza Università di Roma

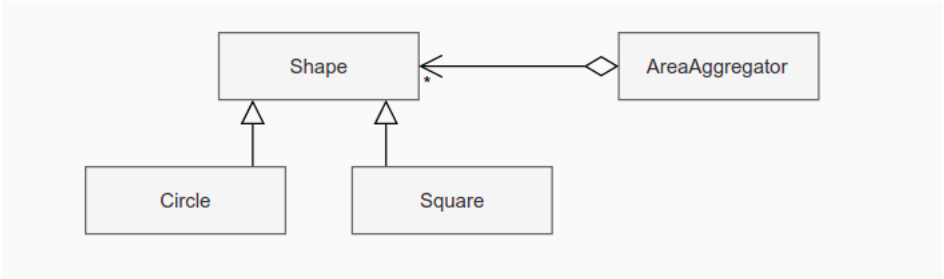
2 Luglio 2021

Durante l'esame non è consentito l'utilizzo di alcunché. Non è consentito inoltre l'utilizzo della matita o di penne il cui colore sia diverso dal nero o dal blu. Il ritiro dalla prova equivale al mancato superamento dell'esame. Le pagine che non sono considerate utili alla valutazione vanno **barrate**.

Nome e Cognome:

Matricola:

1. Descrivere il concetto di interfaccia, spiegando le motivazioni che stanno alla base del suo utilizzo e i vantaggi che apporta allo sviluppo. Produrre un esempio **minimale** scritto in Java che concretizzi quanto spiegato.
2. Sia data l'implementazione del seguente diagramma UML, composto da una classe *AreaAggregator* realizzata con lo scopo di calcolare l'area di una qualsiasi classe derivante da *Shape*:



La classe *AreaAggregator* è implementata nel seguente modo:

```
public class AreaAggregator {
        private ArrayList<Shape> shapes = new ArrayList<>();

        public void addShape(Shape shape) {
          shapes.add(shape);
        }
        public double sum() {
          double sum = 0;

          Circle c = new Circle(6);
          Circle c2 = new Circle(3.5);
          Square s = new Square(4);
```

```
        this.addShape(c);
        this.addShape(s);
        this.addShape(c2);

        for (Shape shape: shapes) {
          if(shape.getName() == "Circle") {
            int radius = ((Circle) shape).getRadius();
            sum += Math.PI * Math.pow(radius, 2);
          } else if (shape.getName() == "Square") {
            double side = ((Square) shape).getSide();
            sum += Math.pow(side, 2);
          }
          counter+= 1;
        }

        System.out.println("Sum:" + sum());
        return String.valueOf(sum); // Converto il risultato in stringa
      }
    }
```

Come si può notare, questa implementazione presenta alcuni errori di tipo sintattico, semantico ed un approccio *object oriented* non del tutto idoneo. Elencare quindi le anomalie e motivarle in modo opportuno. È consentito l'uso di codice Java.

NB: si assume che i metodi *get* siano *public* e che forniscano sempre l'attributo atteso.

3. Sia dato un file contenente le informazioni relative agli studenti iscritti ad un corso. Tali informazioni comprendono nell'ordine: la matricola, il nome, il cognome ed il voto verbalizzato, se presente. I campi sono separati dal carattere ":". Un esempio di contenuto del file è il seguente:

   1:Michele:Apicella:
   2:Salazar:Slytherin:24
   65:Neville:Longbottom:31
   74:Giulio:Ponte:25
   ...

   Si realizzi un programma che, leggendo da input questo file, provveda a:

   - Mostrare la lista degli studenti che hanno già verbalizzato;
   - Mostrare la lista degli studenti che devono ancora verbalizzare;
   - Mostrare le statistiche del corso (studenti iscritti, studenti che hanno superato l'esame, voto medio verbalizzato)

4. Una rubrica telefonica offre la possibilità di memorizzare uno o più contatti, ciascuno di essi rappresentato da un nome, un cognome ed un numero telefonico. Realizzare un programma che permetta di:

   - Restituire la rappresentazione testuale dell'intera rubrica;
   - Aggiungere un contatto alla rubrica;
   - Rimuovere dalla rubrica il contatto associato al nome e cognome forniti in input;
   - Cercare il numero di telefono associato ad un nome e cognome forniti in input;

   Successivamente, scrivere un programma di test che utilizzi tutte le funzionalità implementate.