

PROVA D'ESAME 17 MAGGIO 2023

1. 10 punti Per ogni domanda, indicare con una X la risposta desiderata. Si ricorda che ogni domanda ha al più una risposta corretta. L'assegnazione dei punti alle risposte è la seguente. Verranno attribuiti 2 punti per ogni risposta esatta; -0.4 punti per ogni risposta errata; 0 punti per ogni risposta omessa. Al fine del superamento della soglia è necessario totalizzare un punteggio di almeno 5 punti.
- (a) Un ciclo for è un costrutto di iterazione ☐ Indeterminato ☒ Determinato ☐ Ciclico
 - (b) Una costante in Java è indicata dal modificatore ☐ static ☐ abstract ☒ final
 - (c) I metodi di un'interfaccia sono ☐ protected ☒ public ☐ private
 - (d) Il contenuto di due stringhe è confrontabile tramite il metodo ☐ compareTo ☒ equals ☐ corresponds
 - (e) Un metodo static può essere accessibile tramite ☒ la sua classe ☐ il suo oggetto ☐ l'uso di alcun modificatore di visibilità

2. 4 punti Qual è la differenza tra overloading e overriding? Fornire un esempio minimale, scritto in Java, che descriva quanto richiesto.

SONO ENTRAMBI CONCETTI LEGATI AL POLIMORFISMO IN JAVA, RISPETTIVAMENTE, SI RIFERISCONO A:
OVERLOADING: POSSIBILITA' DI DEFINIRE ALL'INTERNO DI UNA CLASSE, METODI DELLO STESSO TIPO E CON LO STESSO NOME, MA CON DIFFERENTI PARAMETRI IN INPUT.

ESEMPIO:

```
public class Converter {  
    public int convertToInteger(float value) {  
        return Integer.parseInt(value);  
    }  
    public int convertToInteger(double value) {  
        return Integer.parseInt(value);  
    }  
}
```

OVERRIDING: POSSIBILITA' DI DEFINIRE IN UNA CLASSE UN METODO GIA' DEFINITO NELLA SUA SUPERCLASSE, SOVRASCRIVENDOLO, E RENDENDOLO PIU' SPECIFICO ALLA SOTTOCLASSE.

ESEMPIO:

```
public class Color {  
    protected int r;  
    protected int g;  
    protected int b;  
    public Color(int r, int g, int b) {  
        this.r = r;  
        this.g = g;  
        this.b = b;  
    }  
    public void showInfo() {  
        System.out.println(String.valueOf(r) + String.valueOf(g) + String.valueOf(b));  
    }  
}  
  
public class AlphaColor extends Color {  
    protected int a;  
    public AlphaColor(int r, int g, int b, int a) {  
        this.r = r;  
        this.g = g;  
        this.b = b;  
        this.a = a;  
    }  
    @Override  
    public void showInfo() {  
        System.out.println(String.valueOf(r) + String.valueOf(g) + String.valueOf(b) + String.valueOf(a));  
    }  
}
```

3. [3 punti] Per ogni costrutto iterativo, indicare il numero di volte per il quale viene eseguito il suo corpo. Se non diversamente indicato, si assume che la variabile contatore non venga modificata all'interno del corpo di ciascun costrutto iterativo.

- (a) for(int i = 12; i <= 7; i++){...} **0**
- (b) for(int i = 0; i < 10; i++){...} **10**
- (c) for(int i = -10; i >= 0; i++){...} **0**
- (d) for(int i = -2; i >= 0; i++){...} **0**
- (e) for(int i = -8; i <= 3; i = i + 4){...} **3**
- (f) for (int k = 0; k < 20; k+=2) { **9**
 if (k + 3 == 1) System.out.println(k + " ");
}

4. [5 punti] Spiegare le principali differenze tra ArrayList e Set in Java.

SONO STRUTTURE DIVERSE CON DIVERSI SCOPI, ARRAYLIST E' UNA CLASSE IMPLEMENTA LIST, CHE A SUA VOLTA ESTENDE COLLECTION, PUO' COLLEZIONARE OGGETTI DI TIPO DIVERSO, AMMETTENDO DUPLICATI, HA DIMENSIONE DINAMICA.

SET E' UN INTERFACCIA CHE ESTENDE COLLECTION, ED E' IMPLEMENTATA DALLE DUE CLASSI HASHSET E TREESET, ESSI SONO INSIEMI DI DIMENSIONE DINAMICA, CHE POSSONO CONTENERE OGGETTI NON AMMETTENDO DUPLICATI, DIFFERISCONO TRA LORO PER IMPLEMENTAZIONE INTERNA: HASHSET E' IMPLEMENTATO CON UNA TABELLA HASH, TREESET CON UN ALBERO, GARANTENDO QUINDI UN ORDINAMENTO.

6. [8 punti] Un supermercato ha accesso ad un database contenente tutte le informazioni sui propri prodotti. Tali informazioni sono rappresentate in formato tabellare, in cui ciascuna riga contiene i seguenti campi separati da uno spazio:

```
public class Product {
    private int cod;
    private String name;
    private int quantity;
    private float priceForUnit;
    public Product(String line) {
        String data[] = line.split();
        cod = Integer.parseInt(data[0]);
        name = data[1];
        quantity = Integer.parseInt(data[2]);
        priceForUnit = Float.parseFloat(data[3]);
    }
    public int getCod() { return cod; }
    public String getName() { return name; }
    public int getQuantity() { return quantity; }
    public float getPriceForUnit() { return priceForUnit; }
}

import java.util.*; import java.io.*;
public class Es4 {
    public static void main(String[] args) {
        ArrayList<Product> products = new ArrayList<>();
        File f = new File("file.txt");
        try {
            Scanner in = new Scanner(f);
            while(in.hasNextLine()) { products.add(new Product(in.nextLine())); }
        } catch (FileNotFoundException e) { return; }
        Product expensive = products.get(0);
        System.out.println("i prodotti esammati sono:");
        for(int i=0; i<products.size(); i++) {
            Product tmp = products.get(i);
            if(tmp.getPriceForUnit() > expensive.getPriceForUnit()) { expensive = tmp; }
            if(tmp.getQuantity() == 0) { System.out.println(tmp.getName()); }
        }
        System.out.println("il prodotto piu' costoso e': " + expensive.getName() + " con un prezzo ad unita' di: " +
            String.valueOf(expensive.getPriceForUnit()));
        for(p:Product in products) {
            if(p.getQuantity() > 0) { System.out.println("prezzo medio " + p.getName() + " = " + String.valueOf(p.getPriceForUnit()/(float)p.getQuantity())); }
        }
    }
}
```