

Cognome e Nome \_\_\_\_\_

Matricola \_\_\_\_\_

## Appello del 4 luglio 2024

### Esercizio 1 [8 punti]

Si considerino le seguenti classi.

```
class A {  
    protected int n;  
  
    public A(int n) {  
        this.n = n;  
    }  
  
    public A(int n, int m) {  
        this(n+m);  
    }  
  
    1 public int metodo(A a) {  
        n = n - a.n;  
        return n;  
    }  
  
    2 public int metodo(B b) {  
        n = n + b.n;  
        return n;  
    }  
}
```

```
class B extends A {  
    public B(int n) {  
        this(n, n+2);  
    }  
  
    public B(int n, int m) {  
        super(n, m);  
    }  
  
    3 public int metodo(A a) {  
        a.n = n - a.n;  
        return a.n;  
    }  
  
    4 public int metodo(B b) {  
        b.n = n + b.n;  
        return n;  
    }  
}
```

Si dica cosa viene stampato a video dal seguente codice:

```
A a = new A(5, 6);  
B b = new B(3);  
A ab = new B(1, 2);  
System.out.println(a.metodo(ab));  
System.out.println(b.metodo(b));  
System.out.println(ab.metodo(a));  
System.out.println(ab.metodo(ab));
```

Si giustifichi la risposta individuando in particolare le firme associate a tempo di compilazione e a tempo di esecuzione ad ogni chiamata di metodo (riempire le caselle della tabella con **1**, **2**, **3**, **4** o **ERRORE** se non è possibile identificare nessun metodo):

	Tempo di compilazione	Tempo di esecuzione
a.metodo(ab)		
b.metodo(b)		
ab.metodo(a)		
ab.metodo(ab)		

---

### Esercizio 2 [10 punti]

[5 punti] Mostrare **passo-passo l'Heap di minimo** che si ottiene a partire dall'Heap vuoto a seguito delle seguenti operazioni da eseguire in sequenza

- a. Inserimento dei valori **9, 50, 7**
- b. Estrazione di un elemento
- c. Inserimento dei valori **36, 4, 8**
- d. Estrazione di due elementi
- e. Inserimento dei valori **28, 10, 2**
- f. Estrazione di un elemento

[5 punti] Mostrare **passo-passo la tabella Hash** di dimensione 13 **con hashing interno (Hashing doppio)** che si ottiene a partire dalla tabella seguente

		<b>28</b>	<b>29</b>					<b>8</b>				
0	1	2	3	4	5	6	7	8	9	10	11	12

a seguito delle seguenti operazioni da eseguire in sequenza

- a. Inserimento dei valori **15, 27, 41**
- b. Cancellazione di **28**
- c. Inserimento dei valori **2, 7**

---

### Esercizio 3 [9 punti]

*Si vogliono gestire, in Java, i movimenti associati al credito di SIM di telefonia mobile.*

Si assuma, senza scrivere nulla, l'esistenza di una classe astratta **Movimento** con

- una variabile di istanza **idSim** (tipo String, private), che contiene il numero telefonico associato alla SIM a cui il movimento si riferisce;
- una variabile di istanza **tempo** (tipo long, private), che rappresenta l'istante del movimento espresso in secondi trascorsi dalle ore 00:00 del 1° gennaio 2000.

e i seguenti metodi di istanza:

- un costruttore che crea un movimento dati campi **idSim** e **tempo**;
- metodo di accesso **public String getIdSim()**.
- metodo di accesso **public long getTempo()**.
- metodo pubblico astratto **public abstract double getImporto()** che restituisce l'importo in Euro del movimento. In particolare, se il movimento è un movimento di spesa viene restituito un valore minore o uguale a zero, mentre se è un movimento di ricarica credito viene restituito un valore positivo.

La classe astratta **Movimento** è estesa dalle sottoclassi (non astratte) **Chiamata**, **Sms** e **Ricarica**.

Si tenga conto del fatto che la classe **Chiamata** ha:

- una variabile di istanza **durata** (tipo int, private) che contiene la durata in secondi della chiamata;
- una variabile di istanza **destinatario** (tipo String, private), che contiene il numero chiamato.

La classe **Sms** deve avere:

- una variabile di istanza **destinatario** (tipo String, private), che contiene il destinatario dell'SMS.

La classe **Ricarica** deve avere:

- una variabile di istanza **importo** (tipo double, private), che contiene l'importo della ricarica e deve valere almeno 0,01.

Si assuma, senza scrivere nulla, che tutte le sottoclassi abbiano i costruttori (che prendono in input gli opportuni parametri e che richiamino opportunamente il costruttore della classe **Movimento**), i metodi di accesso per tutti i campi e l'implementazione del metodo **double getImporto()**.

Si assuma, senza scrivere nulla, l'esistenza di una classe **Gestore** con una variabile di istanza **movimenti** (tipo `ArrayList<Movimento>`, private), contenenti tutte le chiamate, gli sms e le ricariche effettuate da tutte le SIM del gestore.

Si scriva la classe **ChiamataRoaming** che estende **Chiamata** e ha un parametro (privato) **statoEstero** di tipo String con corrispondente **metodo di accesso** pubblico.

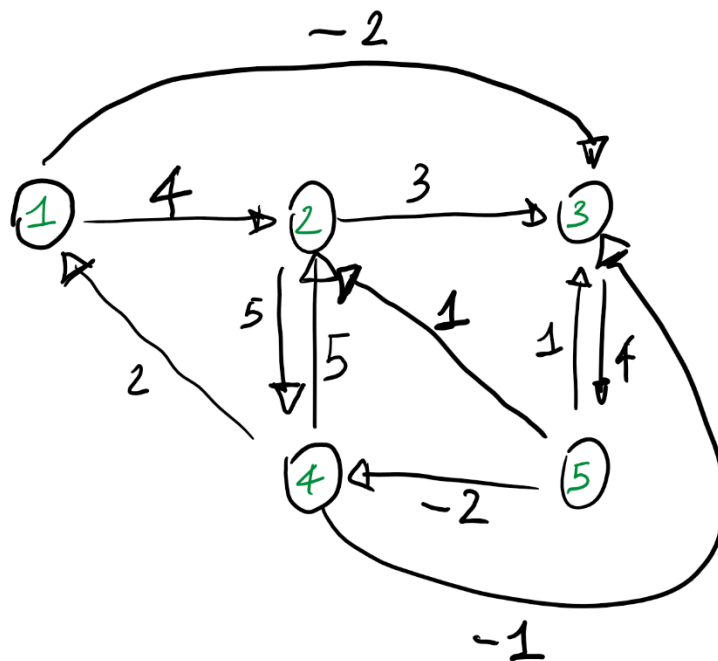
Si scriva un costruttore con gli opportuni parametri, che richiami il costruttore della classe **Chiamata**.

Si implementi anche il metodo **getImporto()**, tenendo conto del fatto che le chiamate effettuate all'estero hanno un costo pari a **Euro  $0,50 + 1,2 * x$** , dove x è il costo che la chiamata avrebbe se fosse dall'Italia (cioè il valore restituito dal metodo **getImporto()** della superclasse **Chiamata**).

Si aggiunga alla classe **Gestore** il seguente metodo:

```
public double getTotaleSpesaChiamateEstero (String idSim, String statoEstero) che restituisce un valore non negativo uguale al totale degli importi delle chiamate effettuate da idSim nello stato statoEstero.
```

#### Esercizio 4 [7 punti]



Mostrare una possibile esecuzione **passo-passo** dell'algoritmo di **Bellman-Ford** per calcolare i cammini minimi a partire dalla sorgente **4**. Mostrare, per ogni iterazione completa che si fa sugli archi, l'evoluzione del **vettore delle distanze** e del **vettore dei padri**, assumendo di visitare, in ogni iterazione, gli archi in ordine crescente rispetto alle etichette dei nodi da cui escono e, a parità di nodo da cui escono, in ordine crescente rispetto alle etichette dei nodi di destinazione. Dire, giustificando la risposta, se il grafo possiede cicli di peso negativo.

#### Regole per lo svolgimento della prova scritta:

- Per svolgere il compito si hanno a disposizione **150** minuti.
- Scrivere **subito** nome, cognome, matricola su **OGNI FOGLIO (compreso questo)**.
- Durante la prova scritta **non** è possibile abbandonare l'aula.
- Non è ammesso **per nessun motivo** comunicare in qualsiasi modo con altre persone
- È possibile consultare appunti, libri e dispense.
- Qualsiasi strumento elettronico di calcolo o comunicazione (telefoni cellulari, calcolatrici, palmari, computer, etc...) deve essere **completamente disattivato e depositato in vista sulla cattedra**
- Mettere in vista sul banco un valido documento di identità.