

Cognome e Nome _____

Matricola _____

Appello del 15 gennaio 2025 - Java

Esercizio 1 [6 punti]

Si considerino le seguenti classi.

```
class A {  
  ① public int metodo(A a) {  
      return 1;  
  }  
  ② public int metodo(B b) {  
      return 15;  
  }  
}
```

```
class B extends A {  
  ③ public int metodo(A a) {  
      return 2;  
  }  
  ④ public int metodo(B b) {  
      return 25;  
  }  
}
```

Si dica cosa viene stampato a video dal seguente codice:

```
A a = new A();  
B b = new B();  
A ab = new B();  
System.out.println(a.metodo(b));  
System.out.println(a.metodo(ab));  
System.out.println(b.metodo(b));  
System.out.println(b.metodo(ab));  
System.out.println(ab.metodo(b));  
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 (b)		
a. metodo (ab)		
b. metodo (b)		
b. metodo (ab)		
ab. metodo (ab)		
ab. metodo (b)		

Esercizio 2 [6 punti]

Si consideri la seguente sequenza di numeri interi:

[10, 20, 16, 5, 9, 33, 27, 3, 31]

1. [3 punti] Mostrare **passo-passo l'evoluzione del min-heap** che si ottiene, a partire dall'heap vuoto, inserendo le chiavi nell'ordine indicato. Dopo aver inserito tutte le chiavi, effettuare **due estrazioni** del minimo mostrando l'heap che si ottiene dopo ogni estrazione.
2. [3 punti] Mostrare **passo-passo l'evoluzione della tabella hash** di dimensione 11 (con hashing doppio), che si ottiene inserendo le chiavi nell'ordine indicato.

Esercizio 3 [11 punti]

Si assuma, senza scrivere nulla, l'esistenza di una *classe Autore* con i seguenti attributi di istanza:

- **codice** (**int**, privato)
- **nome** (**String**, privato)
- **cognome** (**String**, privato)

ed i seguenti metodi di istanza:

- *costruttore* che crea un oggetto della *classe Autore* dati codice, nome e cognome
- metodi di accesso in lettura per tutti gli attributi;
- metodo **equals(Object o)** che restituisce **true** se **o** è un Autore con lo stesso codice di self, (indipendentemente da nome e cognome), **false** altrimenti.
- metodo **toString()** che restituisce la rappresentazione come stringa di un autore, nella forma `Autore n m, codice c`, dove `n`, `m` e `c` rappresentano il nome, il cognome e il codice dell'autore, rispettivamente.

[4 punti] Si scriva una *classe Libro* con i seguenti attributi di istanza:

- **codice** (**int**, privato)
- **titolo** (**String**, privato)
- **autori** (**ArrayList<Autore>**, privato)

ed i seguenti metodi di istanza:

- *costruttore* che crea un oggetto della *classe Libro* assegnando codice e titolo (e creando un `arrayList` vuoto di autori)
- metodo **addAutore (Autore a)** che aggiunge **a** alla lista degli autori del libro, qualora non sia già presente in essa.
- metodo **equals(Object o)** che restituisce **true** se **o** è un libro con lo stesso codice di self, (indipendentemente dagli altri campi), **false** altrimenti.
- metodo **toString()** che restituisce la rappresentazione come stringa di un libro, nella forma `Libro t, codice c`, dove `t` e `c` rappresentano il titolo e il codice del libro, rispettivamente.

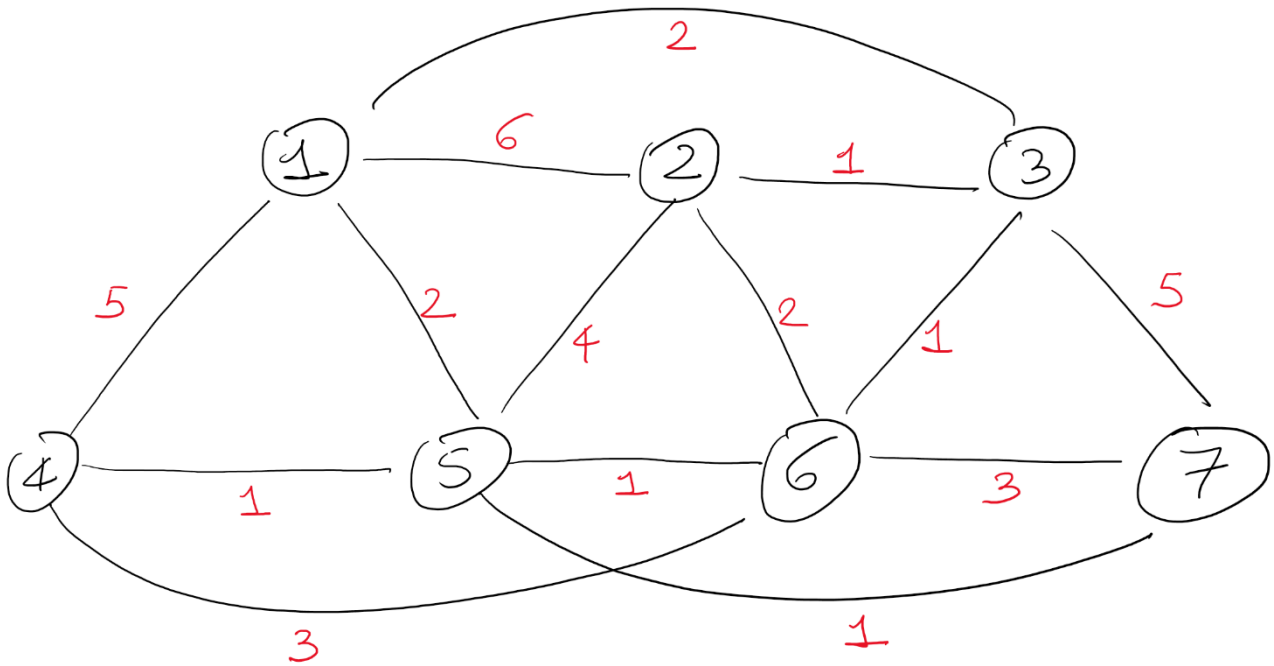
[4 punti] Si scriva una *classe LibroFasciaEtà* che estende la classe **Libro** ed ha i seguenti attributi di istanza:

- **minAge** (**int**, privato)
- **maxAge** (**int**, privato)
- ed i seguenti metodi di istanza:
- *costruttore* con tutti i parametri necessari, che crea un oggetto della *classe LibroFasciaEtà* assegnando **minAge** e **maxAge** (e richiamando opportunamente il costruttore della superclasse)
- metodo **boolean isSuitable (int age)** che restituisce **true** se **self** è un libro adatto per l'età **age**, cioè se **age** è compreso tra **minAge** e **maxAge** estremi inclusi, **false** altrimenti.
- metodo **toString()** che sfruttando il metodo omonimo della superclasse restituisce la rappresentazione come stringa di un libro, nella forma `Libro t, codice c, adatto da min a max anni`, dove `t`, `c`, `min` e `max` rappresentano il titolo, il codice, l'età minima e massima consigliata del libro, rispettivamente.

[3 punti] Aggiungere alla classe **Libro**, un metodo **static ArrayList<Libro> filtraLibri (ArrayList<Libro> libri, int age)** che restituisce una nuova lista contenente tutti e soli i libri della lista **libri** che sono istanza di **LibroFasciaEtà** e sono adatti per l'età **age**.

Esercizio 4 [11 punti]

Si consideri il grafo diretto in figura.



- a. [5.5 punti] Mostrare una possibile esecuzione **passo-passo** dell'algoritmo di **Dijkstra** per i cammini minimi da singola sorgente, a partire dal nodo sorgente **4**. Ad ogni passo, bisogna mostrare il contenuto della coda con priorità utilizzata dall'algoritmo e del vettore dei padri.
Si descriva il cammino minimo dal nodo 4 al nodo 2. Che peso ha?
- b. [5.5 punti] Mostrare una possibile esecuzione **passo-passo** dell'algoritmo di **Prim** per il minimo albero ricoprente, a partire dal nodo sorgente **2**. Ad ogni passo, bisogna mostrare il contenuto della coda con priorità utilizzata dall'algoritmo e del vettore dei padri.
Che peso complessivo ha il minimo albero ricoprente?

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.