

## (A) CONOSCENZA TERMINOLOGICA

Dare una breve descrizione dei termini introdotti:

- Classe
- Istanza di una classe
- Membri privati e pubblici
- Diagramma della classe
- Costruttore senza parametri
- Costruttore con parametri
- Specificatori di accesso
- Metodi di default
- Metodi accessori (**get**)
- Metodi modificatori (**set**)

## (B) CONOSCENZA E COMPETENZA

Rispondere alle seguenti domande producendo anche qualche esempio

B1) Conoscenza

1. Cosa indica il termine *classe*?
2. Cosa vuol dire che i membri di un oggetto possono essere *pubblici* o *privati*?
3. Cosa è il *metodo costruttore*?
4. Cosa è il *diagramma della classe*?
5. Cosa sono i *metodi di default*?

B2) Competenza

1. Come si disegna il *diagramma di classe*?
2. Qual è lo schema della *dichiarazione di una classe*?
3. Quali sono le differenze tra *metodi accessori* e *metodi modificatori*?
4. A cosa serve il *metodo costruttore* e quali sono le sue caratteristiche?
5. Qual è l'utilità di rappresentare una classe mediante il *diagramma della classe*?

## (C) ESERCIZI DI COMPrensione

1. Una classe è un ..... che rappresenta tutti gli ..... aventi le stesse ..... Le caratteristiche di un ..... sono rappresentate dagli ..... e dai ..... Un oggetto è una ..... di una classe.
2. L'interfaccia di una classe esprime ..... possono fare gli oggetti generati da essa, mentre l'implementazione specifica ..... vengono eseguiti i vari metodi.
3. La dichiarazione di una classe ..... memoria, poiché la classe è un concetto ....., mentre la dichiarazione di un oggetto ..... memoria in base al numero e al tipo degli attributi e dei metodi.
4. Il metodo ..... senza parametri è un metodo di ..... che consente di inizializzare un oggetto con attributi di valore ....., mentre il ..... con parametri permette di creare un oggetto con valori degli ..... che vengono passati di volta in volta.
5. I metodi di ..... sono il costruttore senza ....., il costruttore con ....., i metodi ....., detti anche metodi ....., che accedono in lettura agli attributi e i metodi ....., detti anche metodi ....., che permettono di accedere in scrittura agli attributi.
6. Associare i metodi di sinistra con l'effetto posto a destra, scrivendo la lettera corrispondente.

1	<input type="checkbox"/>	Metodi get	A	Creazione oggetto in stato standard
2	<input type="checkbox"/>	Metodi set	B	Creazione oggetto con un determinato stato
3	<input type="checkbox"/>	Costruttore con parametri	C	Accesso in lettura agli attributi
4	<input type="checkbox"/>	Costruttore senza parametri	D	Accesso in scrittura agli attributi

7. Per ciascuna delle seguenti frasi, indicare se vera o falsa:

Metodi di default	Vero	Falso
Il costruttore è un metodo di default		
In una classe può esistere un solo metodo costruttore		
In una classe i metodi di default devono essere pubblici		
I metodi accessori consentono accesso in lettura		
I metodi modificatori consentono accesso in scrittura		
Il costruttore può avere nome diverso da quello della classe		
I metodi set consentono di creare un oggetto in un certo stato		

## (D) ESERCIZI DI APPLICAZIONE

1. Creare una classe *Angolo* con i metodi di default. Aggiungere, successivamente, i metodi per:
  - a. trasformare una misura di angolo interamente in secondi;
  - b. trasformare una misura di angolo da gradi a radianti (tralasciando primi e secondi)
2. Progettare una classe *Razionale* con i metodi di default. Aggiungere, successivamente, i metodi per:
  - a. trasformare un numero razionale nel numero reale equivalente;
  - b. trasformare un numero reale in percentuale
3. Progettare una classe *Polinomio* che consenta di operare con polinomi di grado MAX, in una variabile a coefficienti interi, rappresentati dalla coppia (*grado*, *array dei coefficienti*), fornendo i metodi di default. Aggiungere, successivamente, i metodi per:
  - a. calcolare il valore di un polinomio, noti i valori dei coefficienti e della variabile;
  - b. visualizzare il grado di un polinomio
4. Progettare una classe *Vocabolario*, che consenta di operare con vocaboli bilingue, fornendo le operazioni che si ritengano necessarie.
5. Progettare una classe *Motocicletta* prevedendo un insieme di attributi e metodi per gestire le luci, lo stato del motore, la marcia, la velocità ed il livello di carburante.
6. Progettare la classe *Quadrato* con attributi lato e colore. Prevedere tra i metodi, il calcolo dell'area e del perimetro.
7. Implementare la classe *Array*, indicandone attributi, costruttori, alcuni metodi (lettura, stampa, riempimento con un dato valore, ricerca di un elemento, numero di occorrenze di un elemento).
8. Implementare la classe *Utente*, che contiene Username e Password di un utente di un servizio, e prevedere metodi per consentire o rifiutare l'accesso al servizio stesso.
9. Implementare la classe *Resistore*, che simula un componente resistivo, con i suoi attributi e i metodi per il calcolo della resistenza equivalente del collegamento in serie ed in parallelo.
10. Come l'esercizio precedente, ma per la classe *Condensatore*.
11. Come per l'esercizio precedente, ma per la classe *Induttore*.
12. Implementare la classe *Stringa*, fornendo i metodi di default. Successivamente, aggiungere metodi per:
  - a. ricerca di un carattere in una stringa;
  - b. calcolare il numero di occorrenze di un dato carattere;
  - c. prelevare il carattere in una data posizione di una stringa.
13. Implementare la classe *Centralina* che svolga le funzioni del computer di bordo di un'automobile.