

(A) CONOSCENZA TERMINOLOGICA

Dare una breve descrizione dei termini introdotti:

- Oggetti reali
- Oggetti software
- Astrazione
- Implementazione
- Classi esterne
- Specificatori di accesso
- Metodi **set** e **get**
- Metodo costruttore
- Classe

(B) CONOSCENZA E COMPETENZA

Rispondere alle seguenti domande producendo anche qualche esempio

B1) Conoscenza

1. Cosa indicano la parte *pubblica* e *privata* di una classe?
2. Cosa indica lo *specificatore di accesso*?
3. A cosa serve il *metodo costruttore*?
4. Come *comunicano gli oggetti* all'interno di un'applicazione?

B2) Competenza

1. Qual è la struttura di una *classe Java*?
2. Come si definiscono attributi e metodi di un *oggetto software*?
3. Qual è lo schema di un'applicazione con *classi interne*?
4. Qual è lo schema di un'applicazione con *classi esterne*?
5. Qual è la sintassi del metodo *costruttore con parametri*?
6. Qual è la sintassi del metodo *costruttore senza parametri*?
7. Come si effettua il *test di una classe*?
8. Come si *crea un oggetto* di una data classe?

(C) ESERCIZI DI COMPrensIONE

1. La programmazione OOP è nata intorno agli anni ed è stata un'evoluzione della programmazione Grazie alla, è possibile il del software, per cui si evita di dover riscrivere codice già scritto, una facile del software, per adattarlo a nuove esigenze e la Intesa come protezione sui dati e sui programmi.
2. Un programma Java è già di per sé una Quando si lancia l'esecuzione di un'applicazione Java, ad esempio *Numeri*, viene creato automaticamente l'..... di classe *Numeri* al quale viene inviato il consistente nel
3. A scopo di, il programmatore utilizza gli di accesso, dichiarando i membri, ossia gli e i di tipo o a seconda che sia consentito o meno l'accesso ad essi dall'esterno.
4. Nel progetto di una classe, indichiamo con metodi di quei metodi che in genere sono sempre presenti: abbiamo i metodi, del tipo con o senza, che hanno il compito di l'oggetto, i metodi accessori, detti anche metodi, per accedere in lettura agli attributi, e i metodi modificatori, detti anche metodi che permettono di accedere in scrittura agli attributi.
5. Un oggetto viene creato da una data classe mediante una istanza di questa. La creazione avviene perché viene invocato il metodo
6. L'utilizzo di membri, e, da parte dell'oggetto si fa con il simbolo secondo la sintassi:

$$\text{oggetto.nome_metodo (lista_parametri)}$$

$$\text{oggetto.nome_attributo}$$
7. Associare il termine di sinistra, con la descrizione a destra, scrivendo la lettera corrispondente nella casella.

1	<input type="checkbox"/>	get
2	<input type="checkbox"/>	set
3	<input type="checkbox"/>	costruttore con parametri
4	<input type="checkbox"/>	costruttore senza parametri
5	<input type="checkbox"/>	new

- A** Inizializza oggetti standard
- B** Accede all'oggetto in lettura
- C** Inizializza oggetti con valori dati
- D** Operatore di creazione oggetto
- E** Accede all'oggetto in scrittura

8. Per ciascuno degli esercizi seguenti, completare le parti mancanti, correggere eventuali errori e dare una breve descrizione dell'applicazione
 - a. **public class** Automobile
 { **int** ruote=4, marcia, serbatoio, cilindrata,
 capacitaMax=45;
 String marca, colore;
 boolean statoMotore;
 public Automobile (**String** marca, **String** colore, **int** cilindrata)
 { }

```

        public Automobile()
        { .... }
    }
b. public class TriangoloRettangolo
    {
        int Cateto1, Cateto2;
        public TriangoloRettangolo(int C1, int C2)
        { .... }
        float Ipotenusa( )
        { .... }
        float Superficie( )
        { .... }
        static boolean Equivalenti (TriangoloRettangolo T1, TriangoloRettangolo T2)
        { .... }
        boolean Equivalenti(TriangoloRettangolo T2)
        { .... }
    }
c. class CompactDisc
    {
        String canzonif[10];
        int durata[10];
        public CompactDisc
        { .... }
        public setCanzone (String c, int i )
        { .... }
        public setDurata (int t, int i )
        { .... }
        public getDurata (....)
        { .... }
    }
9. Data la classe
public class Numero
{
    public static int x = 7;
    public int y = 3;
} // end class
e la classe TestNumero contenente le seguenti istruzioni:
Numero a = new Numero();
Numero b = new Numero();
a.y = 5;
b.y = 6;
a.x = 1;
b.x = 2;
System.out.println("a.y = " + a.y);
System.out.println("b.y = " + b.y);
System.out.println("a.x = " + a.x);
System.out.println("b.x = " + b.x);
stabilire l'output prodotto

```

(D) ESERCIZI DI APPLICAZIONE

Realizzare una applicazione che risolva ciascuno dei seguenti problemi

1. Implementare la classe *Vettore*, con attributi opportuni e metodi per inserimento, eliminazione, visualizzazione dei dati e per la ricerca di un dato elemento.
2. Implementare la classe *Automobile*, con attributi *marcia*, *livelloCarburante*, *marca*, *colore*, *cilindrata*, *statoMotore*, *capacitaMax* e metodi *setLivello()*, *getLivello()*, *aumentaMarcia()*, *scalaMarcia()*, *avviaMotore()*, *spegniMotore()*.
3. Implementare la classe *Punto* prevedendo, oltre ai metodi di default, il metodo *proiezioneX()* che dà le coordinate del punto proiezione sull'asse X e il metodo *proiezioneY()* analogo sull'asse Y.
4. Progettare la classe *Rettangolo* e fornire i metodi più comuni per trattare i suoi oggetti.
5. Progettare la classe *Insieme* che simuli un insieme di numeri interi e fornire la classe dei metodi opportuni per gestire i suoi oggetti.
6. Progettare la classe *TriangoloRettangolo* e fornire i metodi più comuni per trattare i suoi oggetti.

(E) ESERCITAZIONI PRATICHE
Esercitazione n. 1

Obiettivi: dichiarazione di metodi e di attributi, implementazione di metodi, creazione di classi e di oggetti.

Problema: implementare la classe *Rettangolo*.

- 1) Attivare l'ambiente di sviluppo (TextPad, Eclipse, ecc)
- 2) Creare un nuovo file e salvarlo come *Rettangolo.java*. Scrivere il codice per implementare la classe *Rettangolo* avente come attributi:
 - a. Base, intero
 - b. Altezza, interoe come metodi:
 - a. **void** setBase(**int** b)
 - b. **void** setAltezza(**int** a)
 - c. **int** getBase()
 - d. **int** getAltezza()
 - e. **int** area()
 - f. **int** perimetro()
 - g. **float** diagonale()
 - h. *Rettangolo*(**int** a, **int** b) (costruttore con parametri)
 - i. *Rettangolo*() (costruttore senza parametri)
- 3) Compilare il file *Rettangolo.java* fino ad ottenere un codice corretto.
- 4) Creare un nuovo file e salvarlo con il nome *TestRettangolo.java* e implementarlo in modo che svolga le seguenti operazioni:
 - a. crei tre oggetti R1, R2 ed R3 di classe *Rettangolo*, con le caratteristiche indicate nella tabella **Tab. 1**:
 - b. imposti le dimensioni (base ed altezza) per il solo oggetto R1;
 - c. stampi, in ordine, l'area ed il perimetro di ciascun rettangolo, accompagnando i valori con delle frasi esplicative.

Oggetto	Base	Altezza
R1	0	0
R2	2	7
R3	1	8

Tab. 1 – Dati di prova

(E) ESERCITAZIONI PRATICHE
Esercitazione n. 2

Obiettivi: dichiarazione di metodi e di attributi, implementazione di metodi, creazione di classi e di oggetti.

Problema: implementare la classe *Libro*.

- 1) Attivare l'ambiente di sviluppo (TextPad, Eclipse, ecc)
- 2) Creare un nuovo file e salvarlo come *Libro.java*. Scrivere il codice per implementare la classe *Libro* avente come attributi:
 - a. `codLibro`
 - b. `nCopie`
 - c. `autore`
 - d. `editore`
 - e. `titolo`
 - f. `prezzo`
 e come metodi
 - a. due costruttori, uno con parametri ed uno senza;
 - b. `getCopie()`, `getAutore()`, `getEditore()`, `getTitolo()`, `getPrezzo()`
 - c. `set(Copie())`, `setAutore()`, `setEditore()`, `setTitolo()`, `setPrezzo()`
- 3) Compilare file *Libro.java* fino ad ottenere un codice corretto.
- 4) Creare un nuovo file e salvarlo con il nome *TestLibro.java* e implementarlo in modo che crei 2 oggetti *L1* ed *L2* con i valori degli attributi riportati nella seguente tabella:

Oggetto	codLibro	nCopie	autore	editore	titolo	prezzo
<i>L1</i>	1210	2	A. Manzoni	Bompiani	I promessi sposi	25.50
<i>L2</i>	1250	3	D. Alighieri	Einaudi	La divina commedia	22.30

- b. aggiornare l'attributo `nCopie` dell'oggetto *L2* con un nuovo valore letto da input;
- c. stampare i dati complessivi dei due libri, per verificare l'avvenuto aggiornamento.

(E) ESERCITAZIONI PRATICHE
Esercitazione n. 3

Obiettivi: dichiarazione di metodi e di attributi, implementazione di metodi, creazione di classi e di oggetti.

Problema: implementare la classe *Punto*.

- 1) Attivare l'ambiente di sviluppo (TextPad, Eclipse, ecc).
- 2) Creare un nuovo file e salvarlo come *Punto.java*. Scrivere il codice per implementare la classe *Punto* avente come attributi:
 - a. x (ascissa)
 - b. y (ordinata)e come metodi
 - a. due costruttori, uno con parametri ed uno senza;
 - b. `getX()`, `getY()`;
 - c. `setX()`, `setY()`, `stampaPunto()`, `distanza()`, `puntoMedio()`, `proiezioneX()`, `proiezioneY()`.
- 3) Implementare i metodi con opportune scelte degli eventuali parametri;
- 4) Compilare file *Punto.java* fino ad ottenere un codice corretto
- 5) Creare un nuovo file e salvarlo con il nome *testPunto.java* e implementarlo in modo che:
 - a. crei due oggetti di classe *Punto*, P1(3, 2) e P2 (0, 0) e ne stampi le coordinate nel formato consueto "P (ascissa, ordinata)";
 - b. stampi la distanza P1P2 (verificare il risultato con il calcolo manuale);
 - c. calcoli e stampi l'oggetto P3, punto medio tra P1 e P2 (verificare il risultato con il calcolo manuale)