

# **PROGRAMMA DI INFORMATICA**

## **CLASSE IV O**

**ANNO SCOLASTICO 2018-19**

**DOCENTI: prof.ssa Nicolina Anna Ragosa - prof.ssa Irene Conti**

### **Unità didattica 1 Caratteristiche del linguaggio Java**

- Caratteristiche generali
- La struttura dei programmi
- Gli identificatori
- Variabili e costanti
- Tipi di dati
- Operatori
- La gestione dell'input/output
- Le strutture di controllo
- La struttura di dati: array
- Le eccezioni

### **Unità didattica 2 Classi e oggetti**

- Orientamento agli oggetti
- Gli oggetti e le classi
- Dichiarazione di una classe
- Dichiarazione degli attributi
- Dichiarazione dei metodi
- Progettazione ed implementazione di una classe
- Creazione ed uso di oggetti
- La gestione delle stringhe
- Array di oggetti
- Esempi di classi
- Utilizzo degli oggetti
- Realizzazione di programmi object-oriented
- Array di oggetti
- Ereditarietà
- Polimorfismo
- Le stringhe

### **Unità didattica 3 Programmazione guidata dagli eventi ed interfaccia grafica**

- L'interfaccia per l'utente
- Gli elementi dell'interfaccia grafica
- L'ambiente di programmazione
- Creazioni di applicazioni in NetBeans
- Etichette e pulsanti
- Caselle ed aree di testo
- Gestione degli eventi

#### **Laboratorio**

- Utilizzo dell'ambiente Java riguardante gli esercizi svolti in classe

Programmazione recupero informatica  
ANNO SCOLASTICO 2018/2019

CLASSE IV O

**Unità didattica 2 Classi e oggetti**

- Orientamento agli oggetti
- Gli oggetti e le classi
- Dichiarazione di una classe
- Dichiarazione degli attributi
- Dichiarazione dei metodi
- Progettazione ed implementazione di una classe
- Creazione ed uso di oggetti
- La gestione delle stringhe
- Esempi di classi
- Utilizzo degli oggetti
- Realizzazione di programmi object-oriented
- Array di oggetti
- Ereditarietà

**Di seguito sono proposti alcuni esercizi:**

- 1) Si scriva una classe **Lavoratore** le cui istanze rappresentano lavoratori. Ogni lavoratore è caratterizzato da un nome, un livello ed uno stipendio (mensile). La classe deve contenere una variabile di istanza di tipo stringa nome e due variabili di istanza intere, livello e stipendio, un costruttore che assegna al lavoratore un numero ed un livello specificati, ed un metodo stampaStipendio() che stampa lo stipendio del lavoratore. Scrivere quindi una classe **LavoratoreConStraordinariPagati** che estende la classe **Lavoratore**. La classe deve contenere una nuova variabile di istanza, oreStraordinario, un costruttore che assegna al lavoratore con straordinari pagati un nome, un livello e un numero di ore di straordinario specificati, un metodo stampaStraordinari() che stampa un messaggio contenente il numero di ore di straordinario. La classe deve inoltre contenere una variabile statica reale retribuzioneOraria, inizializzata a 10.0, corrispondente alla retribuzione di un'ora di straordinario (che si suppone uguale per tutti). La classe deve inoltre ridefinire il metodo stampaStipendio() per tenere conto della retribuzione delle ore di straordinario: alla retribuzione base (ereditata da **Lavoratore**) va sommata la retribuzione degli straordinari, ottenuta come retribuzione oraria dello straordinario per numero di ore di straordinario effettuate. Scrivere infine un programma di prova per collaudare le classi e i metodi.
- 2) Progettare e realizzare una classe **Automobile** con i seguenti attributi **Marca, Tipo,, Cilindrata;, ConsumoKm; Livellomax; livello serb**  
L'automobile ha una determinata resa del carburante (misurata in litri/chilometri) e una certa quantità di carburante nel serbatoio.

il livello iniziale del carburante è a zero. Fornire questi metodi: un metodo drive per simulare il percorso di un'automobile per una certa distanza, riducendo il livello di carburante nel serbatoio; un metodo getGas, per ispezionare il livello corrente del carburante;

un metodo addGas per far rifornimento.

Scrivere un programma che serve a testare le funzionalità della classe

- \* Definisce e crea oggetti di classe Automobile ,
- \* con dati inseriti da tastiera da tastiera.
- \* Effettua delle operazioni usando i metodi
- \* Visualizza le informazioni relative a ciascuna Automobile .

3) Si vuole definire una classe Java "E\_Book" che i cui oggetti rappresentano libri in formato elettronico.

Ogni oggetto E\_Book deve avere almeno le seguenti caratteristiche

Titolo, Autore, Numero Pagine. Viene fissato in 0.05 euro il costo unitario di ogni pagina indipendentemente dallo specifico E\_Book.

Dopo aver fornito il diagramma UML della classe E\_Book, si realizzi il codice java che la implementi prevedendo i seguenti metodi:

- un costruttore che prenda come parametri titolo, autore e numero pagine;
- un metodo che permetta di calcolare il prezzo determinato dal costo fisso dell'E\_Book (4 euro) più il costo totale delle pagine contenute in esso;
- getti/setting per ogni attributo (n.b.: modificando il numero delle pagine deve essere ricalcolato il prezzo); un metodo toString che restituisce una stringa con tutti i dati dell'oggetto su cui è invocato separati dal carattere “;”;
- un metodo equals che permetta di vedere se due E\_Book sono uguali, ritornando un valore opportuno;
- un metodo main che preveda la creazione di un paio di oggetti di tipo E\_Book, l'utilizzo su uno di questi del metodo toString visualizzandone il risultato e verifichi se i due oggetti di tipo E\_Book hanno o meno gli stessi valori per le variabili di ambiente

Per ogni esercizio diagramma Uml e codifica testata con il compilatore

La verifica di recupero sarà scritta

L'INSEGNANTE  
Nicolina Anna Ragosa

**Roma, 7 Giugno 2019**