



# **PROGRAMMA SVOLTO**

ANNO SCOLASTICO 2023/2024

CLASSE 4<sup>a</sup> SEZ. B-IT

INDIRIZZO **INFORMATICA E TELECOMUNICAZIONI**

ARTICOLAZIONE **INFORMATICA**

PROF. ING. **GIOVANNI GIUSEPPE D'ALBERTI**

**MATERIA INFORMATICA**

## ISTITUTO TECNICO ECONOMICO e TECNOLOGICO "G. CARUSO"

**Materia:** INFORMATICA

**Anno:** 2023/2024

**Docente:** D'ALBERTI GIOVANNI GIUSEPPE

**Classe:** 4<sup>a</sup> B I.T.

**Docente tecnico-pratico:** LABRUZZO CATERINA

**Libro di testo:** Corso di Informatica Java Volume B – Camagni P., Nikolassy R. – Ed. Hoepli

### PROGRAMMA SVOLTO

#### LA PROGRAMMAZIONE OOP

- BASI DELLA OOP
  - Classi ed oggetti
  - Stato e comportamento di un oggetto
  - UML
- PILASTRI DELLA OOP
  - Astrazione
  - Incapsulamento
  - Ereditarietà
  - Polimorfismo
- CLASSI ED OGGETTI
  - Attributi e metodi di una classe
  - Il metodo costruttore e l'istanza di una classe
  - Class diagram
  - Superclassi e sottoclassi
  - Classi e metodi astratti
  - Classi di tipo interfaccia
  - Visibilità di metodi ed attributi: pubblica, privata e protetta
  - Attributi e metodi di classe

#### STRUTTURE DI DATI

- LE STRUTTURE DATI DINAMICHE
  - Le liste
  - La dichiarazione di liste di oggetti
  - L'uso dei principali metodi di una lista

#### I LINGUAGGI DI PROGRAMMAZIONE OOP: C# VS JAVA

- CONFRONTO TRA I LINGUAGGI C# E JAVA:
  - Linguaggi procedurale e ad oggetti
  - Linguaggi compilati, interpretati e misti
  - Piattaforme/ambienti di esecuzione
  - La Java Virtual Machine ed il Bytecode
  - I tipi di dati: primitivi e non primitivi con relative caratteristiche
  - Le dichiarazioni di variabili e costanti, le operazioni di casting
  - Gli operatori
  - Le istruzioni per l'input e l'output
  - Librerie e package
  - Sottoprogrammi e metodi
  - I costrutti di selezione: if, if-else, if-elseif e switch
  - I costrutti di iterazione: while, do-while, for e foreach
  - Le istruzioni break e continue

- La dichiarazione e gestione degli array

## IL LINGUAGGIO DI PROGRAMMAZIONE JAVA

### • CARATTERISTICHE GENERALI DEL LINGUAGGIO JAVA

- OOP
- Linguaggio di alto livello
- Seminterpretato
- Bytecode
- Java Virtual Machine
- Portabilità

### • LO SVILUPPO DI UN'APPLICAZIONE PROGRAMMATA AD OGGETTI

- L'ambiente di sviluppo *IntelliJ IDEA CE*:
  - Ricerca, download ed installazione
  - Avvio dell'ambiente di sviluppo
  - Creazione e memorizzazione di un nuovo progetto di tipo *Java Application*
- L'interfaccia grafica di *IntelliJ IDEA CE*, i pannelli: **Project Files** e **Code Editor**
- La finestra del codice **Source**:
  - La scrittura del codice di una classe
  - Il debug del codice di un'applicazione Java
  - L'esecuzione di un'applicazione Java
- File sorgente **.java** e file in bytecode **.class**

### • COSTRUTTI DI BASE IN JAVA

- Istruzioni per l'input: **System.out.println()**, **System.out.print()**
- Istruzioni per l'output la classe **Scanner** con i metodi **next()**, **nextLine()**, **nextInt()**, **nextByte()** e **nextDouble()**
- Istruzioni per la selezione: **if**, **if-else**, **switch-case**
- Istruzioni per l'iterazione: **while**, **do-while**, **for**
- La gestione delle eccezioni in Java
  - le istruzioni **try...catch** e **finally** (sintassi ed esempi)
  - le istruzioni **throw...new**, **ArithmeticException** (sintassi ed esempi)

### • CLASSI E OGGETTI

- Anatomia di una classe in Java: **package**, **import**, nome della classe, attributi e metodi della classe
- Sintassi per la creazione di una classe in Java: **attributi**, metodo **costruttore**, metodi **get()** e **set()**
- Sintassi per la creazione di sottoclassi che ereditano da superclassi in Java e loro uso
- Sintassi per la creazione di classi e metodi astratti in Java e loro uso
- Modificatori di visibilità degli attributi e dei metodi: **private**, **public**, **protected** e **final**
- Signature e corpo di un metodo
- Inserimento di commenti nei metodi
- Parametri e valore restituito da un metodo
- Sintassi per la creazione di un oggetto: l'operatore **new**
- Attributi e metodi **static**
- Sintassi per l'invocazione dei metodi
- La classe **Main** ed il metodo **main()**
- UML e la rappresentazione delle classi nella OOP:
  - Rappresentazione di classe semplificata
  - Rappresentazione di classe dettagliata
  - Gerarchie di classe: associazione, aggregazione, composizione
  - Ereditarietà tra classi
  - Implementazione di classi da interfacce

- Uso della web application **draw.io** per la rappresentazione grafica dei class diagrams.
- STRUTTURE DATI DINAMICHE: LE LISTE IN JAVA
  - Le classi ArrayList e LinkedList
  - Modalità di utilizzo della memoria di una lista
  - La creazione di oggetti da queste classi e utilizzo dei principali metodi add(), remove(), clear(), ecc.
  - L'istruzione **for each** e la gestione delle liste.

**IL LINGUAGGIO DI PROGRAMMAZIONE JAVA E LO SVILUPPO DI APP CON INTEFACCIA GRAFICA**

- LO SVILUPPO DI UN'APPLICAZIONE PROGRAMMATA AD OGGETTI CON INTELLIJ E SCENEUILDER
  - Gli ambienti di sviluppo *IntelliJ* e *SceneBuilder*:
    - Ricerca, download ed installazione
    - Avvio e settaggio degli ambienti di sviluppo
    - Installazione dei plugin **JavaFX** e **JavaFX RunTime for Plugins**
    - Le interfacce grafiche di *IntelliJ* e *SceneBuilder*
  - Creazione e memorizzazione di un nuovo progetto di tipo *JavaFX* con IntelliJ
  - Le classi Main e Controller di un progetto JavaFX
  - La creazione del file **view.fxml** per la definizione dell'interfaccia grafica dell'app tramite SceneBuilder
  - Il paradigma **Model-View-Controller** e la sua applicazione alla costruzione di app con IntelliJ e SceneBuider

**EDUCAZIONE CIVICA**

- Nucleo concettuale: Cittadinanza Digitale
- Attività svolta: Realizzazione di un e-book che illustra le tematiche sull'arte sviluppate nelle discipline Italiano, Storia e Inglese.
- Tematiche trattate: La rivoluzione digitale e i suoi effetti sull'economia globale.

Alcamo li, 7 giugno 2024

Il docente del corso  
 prof. ing. Giovanni Giuseppe D'Alberti

La docente tecnico pratica  
 prof.ssa Caterina Labruzzo

---



---

Gli studenti

---



---



---



---