



PROGRAMMA SVOLTO

ANNO SCOLASTICO 2022/2023

CLASSE 4^a 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: 2022/2023

Docente: D'ALBERTI GIOVANNI GIUSEPPE

Classe: 4^a B I.T.

Docente tecnico-pratico: TAMBURELLO MARIO

Libro di testo: Informatica per Istituti Tecnici Tecnologici B – Lorenzi A., Rizzi A. – Ed. Atlas

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 E GESTIONE DEI FILE

- LE STRUTTURE DATI DINAMICHE
 - Le liste
 - La dichiarazione di liste di oggetti
 - L'uso dei principali metodi di una lista
- I FILE DI TESTO
 - Creazione apertura e chiusura di un file di testo
 - Lettura e scrittura di un file di testo
 - Accesso ad un file in modalità sovrascrittura oppure modifica

I LINGUAGGI DI PROGRAMMAZIONE OOP: C# VS JAVA

- CONFRONTO TRA I LINGUAGGI C# E JAVA:
 - I tipi di dati
 - Le dichiarazioni di variabili e costanti
 - 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
 - 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 *Apache NetBeans*:
 - Ricerca, download ed installazione
 - Avvio dell'ambiente di sviluppo
 - Creazione e memorizzazione di un nuovo progetto di tipo *Java Application*
 - L'interfaccia grafica di *Apache NetBeans*, i pannelli: **Projects**, **Files** e **Navigator**
 - 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

- I FILE IN JAVA

- Le classi **File**, **Scanner** e **FileWriter**
- Lettura/scrittura di file di testo tramite i metodi: **createNewFile()** della classe **File**; **next()**, **nextInt()**, ecc., **close()** della classe **Scanner**; **write()** e **close()** della classe **FileWriter**.

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
- Tematiche trattate:
 - L'economia digitale
 - La rivoluzione digitale e i suoi effetti sull'economia globale
 - Intelligenza artificiale, globalizzazione e sviluppo economico: come affrontare le sfide della rivoluzione digitale?

Alcamo li, 7 giugno 2023

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

Il docente tecnico pratico
prof. Mario Tamburello

Gli studenti

