

PROGRAMMAZIONE DIDATTICA

Disciplina: **INFORMATICA**

Classe: **4BINF I.T. Tecnologico - corso Informatica e Telecomunicazioni 2022/2023**

Finalità: Il corso di Informatica ha come fine principale quello di mettere in grado lo studente di affrontare la soluzione di un problema posto dalla richiesta di un ipotetico committente, scegliendo le metodologie e gli strumenti software più idonei, e di fornirgli inoltre la formazione di base che gli consenta di seguire con una certa autonomia l'evoluzione delle tecnologie informatiche.

Obiettivi:

L'obiettivo della programmazione curricolare annuale è far acquisire all'alunno le competenze per affrontare progetti di una certa dimensione anche con la metodologia orientata agli oggetti. Deve acquisire una certa flessibilità che gli consenta di analizzare e risolvere problemi di varia natura utilizzando il paradigma e gli strumenti più idonei alla natura del problema.

Al termine della quarta classe gli allievi dovranno raggiungere i seguenti obiettivi:

Svolgere un'analisi qualitativa degli algoritmi.

Applicare con proprietà tecniche di programmazione modulare.

Elaborare file ad organizzazione sequenziale e random.

Organizzazioni con archivi tradizionali - File sequenziali

Definire e manipolare strutture di dati connesse da puntatori.

Applicare il paradigma della programmazione orientata agli oggetti.

Applicare il paradigma della programmazione guidata degli eventi.

Progettare e costruire interfacce d'utente amichevoli.

Conoscere ed usare classi notevoli di oggetti contenitori.

Conoscere le basi della compilazione/interpretazione.

Analizzare criticamente le conoscenze pregresse d'informatica.

Contenuti:

La programmazione ad oggetti:

Classi ed oggetti.

Attributi e metodi.

Variabili e metodi di classe. Variabili locali e variabili istanza.

Il concetto di riferimento ad un oggetto e modello della memoria (stack e heap)

Incapsulamento e information hiding

Ereditarietà e gerarchie.

Classi astratte e interfacce.

Polimorfismo. Casting.

Eccezioni.

La modellazione UML e metodologie di progettazione.

Proposta di una metodologia di progettazione di una applicazione ad oggetti

Proposta di una metodologia di documentazione di una applicazione (UML)

Strutture dati ad oggetti in Java:

classe array, ArrayList, hashmap

definizione di wrapper di una classe

navigazione degli arraylist: metodo get

La gestione delle eccezioni.

Elementi di programmazione ad oggetti

Classi, oggetti e costruttori. Metodi di istanza e di classe. Variabili di istanza e principio di incapsulamento.

Qualificatori private, public e protected. Metodi accessori.

Fasi di allocazione di una variabile (in stack e nello heap).

Associazioni tra gli oggetti per composizione e aggregazione con esempi e casi d'uso legati al concetto di "avere".

Associazioni tra gli oggetti per ereditarietà con esempi e casi d'uso legati al concetto di "essere".

Array di classi base e concetto di late binding di un oggetto della classe derivata.

Le interfacce in Java, il concetto di contratto, implementazione delle interfacce in casi notevoli come la ricerca binaria e sort.

Realizzazione di un programma completo con GUI a carattere (menu) che gestisca array di oggetti derivati con funzioni di inserimento, cancellazione, ricerca e modifica di un oggetto. Schema generale

UML e progettazione delle classi per scenari di casi d'uso base.

Utilizzo di instanceof per la determinazione del tipo di una classe derivata (funzione di modifica).

Stringhe JSON e parsing di oggetti Json in Java.

| |
|--|
| <p>Richieste http bloccanti in java. Costruzione di un client, costruzione di una richiesta GET e POST. Metodo send() per l'esecuzione della richiesta. Codifica URL encoded dei parametri POST. Realizzazione di richieste di esempio su openData in Internet che restituiscono oggetti JSON.</p> <p><u>Il linguaggio Javascript</u></p> <p>Richiami di HTML e CSS (paragrafi, tabelle e liste, e stile di paragrafi e tabelle) I form in HTML e i tag correlati: input (radio, select, text, button) Gli attributi id per la individuazione di un elemento HTML nella pagina Javascript: costrutti base, differenze con il linguaggio java, le variabili non tipizzate. Allocazione di variabili locali e globali (let, var). Posizione consigliata del codice javascript all'interno di una pagina HTML. Il DOM di un documento HTML, metodi getElementById, innerHTML, style. Fasi del procedimento generale per la creazione e l'inserimento di un elemento HTML all'interno del DOM. Utilizzo del DOM per la costruzione dinamica di tabelle e select. Gestione degli eventi: attributi onclick e onchange, onSubmit, funzioni anonime, il riferimento this. Gli array in javascript: metodi toString, push e pop, metodi length, sort. Gestione dinamica form mediante l'oggetto FormData. Metodo get per la lettura dei parametri FormData per la validazione di input di testo, password, select, button, radio button, check box. Comando preventDefault() per l'intercettazione della action del form. Tipi di codifica dei parametri POST (JSON e url encoded). Istruzione URLSearchParams() per la codifica url encoded. Parsing delle stringhe JSON in Javascript. Eventi in JS, gestione eventi, gestione callback, messaggi. Promesse, gestione di promesse annidate. Blocco try catch e generazione di errori utente. Callback async e qualificatore await (davanti alle funzioni che restituiscono promise) per la emulazione di codice lineare/sequenziale. Realizzazione di app web che chiamano microservizi JWT. Flusso di controllo tipico per la gestione dell'autenticazione ad un microservizio REST basata su token JWT. Esempi applicativi su microservizi che realizzano operazioni matematiche.</p> <p>Esercizi tipici: Creazione di elementi HTML dinamici (tabelle, select), controllo della correttezza dei campi di un form (campi non vuoti, campi numerici e non numerici, correttezza di una mail), confronto dei valori dei campi di un form con i valori memorizzati in variabili o array, selezione e aggregazione di contenuti da più elementi dello stesso tipo (ad es. comporre una stringa con i valori dei campi di un form). Interrogazione HTTPS di microservizi JWT.</p> |
| <p>Strumenti di lavoro:</p> <ol style="list-style-type: none"> 1. Lavagna 2. Videoproiettore o LIM 3. Testo in adozione 4. Laboratorio di Informatica 5. Piattaforma e-learning moodle per slides, esercizi ed esempi su tutti gli argomenti del corso |
| <p>Modalità di lavoro:</p> <ol style="list-style-type: none"> 1. Spiegazioni in classe e/o in laboratorio 2. Utilizzo del laboratorio di Informatica 3. Studio e svolgimento di esercizi a casa 4. Recupero periodico |
| <p>Tipologie e numero di verifiche:</p> <ol style="list-style-type: none"> 1. Domande a risposta aperta/chiusa 2. Verifiche orali 3. Verifiche pratiche <p>Nel trimestre/pentamestre ogni studente sarà sottoposto mediamente a tre/quattro prove di verifica sommativa di tipologia varia.</p> |

I docenti

Sebastiano Melita

Fabrizio Manzella