

IIS "G. Cigna G. Baruffi F. Garelli"	Programmazione svolta Piano didattico annuale
---	---

Materia	Informatica
---------	-------------

Rif. Programmazione dipartimentale	Segue la "Progr. Del Dipartimento di Matematica e Informatica sede: "G.Baruffi""
Triennio	Amministrazione Finanza e Marketing Articolazione Sistemi Informativi Aziendali
Classe	Classe 4^A SIA
Libro	Titolo: Corso di Informatica - quarto anno Editore: PEARSON Autori: Barbero, Vaschetto

UNITÀ DI APPRENDIMENTO 1 : - Object Oriented Concetti avanzati

CONTENUTO:

- Il paradigma Object Oriented differenze rispetto all'approccio *imperativo* (analisi dei vari paradigmi di programmazione)
- I Concetti di Classe oggetto, attributi e metodi
- Linguaggio UML
 - Cenni al linguaggio UML
 - Il diagramma delle Classi
- I Principi di *Data Hiding* e di *Incapsulamento* (la visibilità nel paradigma Object oriented), l'interfaccia
- il costruttore
- L'esecuzione di un programma concepito con la metodologia OO: lo scambio di messaggi tra oggetti e la dot notation
- Concetti Avanzati del paradigma OO:
 - Relazione IS-A ed *ereditarietà* : Superclasse Sottoclasse e meccanismo dell'ereditarietà (notazione UML sul diagramma delle classi). Ereditarietà Singola Vs. Ereditarietà Multipla
 - Motivazione all'adozione dell'ereditarietà
 - Evoluzione della Visibilità dei membri di una classe nella relazione IS-A , lo specificatore *protected*
 - design pattern *cenni*
 - L'ereditarietà con ridefinizione dei metodi – il polimorfismo degli oggetti: overriding e overloading Altre relazioni tra classi differenti da quella IS-A (composizione aggregazione)

SPERIMENTAZIONI

- I costrutti offerti dal linguaggio Python per supportare il paradigma Object Oriented
- La definizione di una classe (convenzioni sulla nomenclatura dell'identificatore di una classe). *class*.
- La "dichiarazione" dei membri della classe
 - variabili di Istanza (approfondimento: differenza con le variabili locali, inizializzazione e visibilità)
 - metodi di Istanza la firma di un metodo
 - il metodo costruttore
- specificatore di accesso ai membri di una classe *public* e *private*
- Ereditarietà in python
- la valutazione di un'espressione di assegnazione in presenza di ereditarietà
- la valutazione di una chiamata di metodo in presenza di estensione e ridefinizione (binding dinamico)
- keyword *this* e *super* (in presenza di ereditarietà invocazione membri superclasse..)
- File e operazioni per la persistenza
- Le collezioni in Python (oltre l'array, strutture dati dinamiche)
-
- Si illustrano/ripassano alcune scorciatoie di uso frequente nell'IDE pycharm CE
- esercitazione volte a implementare applicazioni semplificate oo: gestione di una rubrica, di conti in una contabilità semplificata, di libri (biblioteca), di dipendenti..

UNITÀ DI APPRENDIMENTO 2 : - DBMS Relazionali

CONTENUTO:

- Introduzione ai DBMS nel contesto del sistema informativo aziendale (Motivazioni alla loro introduzione visto il fallimento di approcci alternativi)
- L'architettura di un DBMS e i modelli usati per descrivere i dati a differenti livelli.
- Progettazione di una Base di dati Relazionale:
 - Modellazione Concettuale (Modello ER)
 - il Modello Relazionale dei dati
 - Da modello concettuale ER a modello logico Relazionale
 - ~~Forme normali~~
- Utilizzo di un DBMS Relazionale
 - algebra relazionale (operazioni proiezione, selezione, join e loro applicazione)
 - operatori insiemistici (Unione, differenza, intersezione, prodotto cartesiano e loro applicazione)
 - il linguaggio SQL
 - ♦ DDL (tipi di dati, creazione domini, comandi per la dichiarazione delle tabelle, vincoli di integrità referenziale, cancellazione e modifica)
 - ♦ DML ,(operatori aggiornamento inserzione, modifica cancellazione) DQL (gli operatori algebrici in SQL:selezione,proiezione,join e varianti, IN, LIKE,ORDER BY, SUM, AVG,MAX,MIN,COUNT, GROUP BY e HAVING)

SPERIMENTAZIONI:

- La gestione della persistenza degli applicativi, FILE. Gestione con Record a lunghezza variabile e fissa. La persistenza in Python Esercitazioni (manipolazione file json, e testuali).
- SQL, DML e DQL : Applicazione degli operatori dell'algebra relazionale e insiemistici gradualmente a modelli di una sola relazione e poi alcuni casi con più relazioni. Interagendo da linea di comando verso DBMS, o inserendo le query in codice Python
- SQL DDL Creazione dello schema logico progettato

UNITÀ DI APPRENDIMENTO 3 : - Sviluppo di Applicazioni multi tier “web”

CONTENUTO:

- Comunicazione tra Calcolatori, il paradigma client/server e il protocollo applicativo HTTP
- Linguaggi del web (HTML, CSS, *cenni XML*)
- Sviluppo di Applicazioni multi-tier con business logic (Application server), DBMS, e livello di presentazione

SPERIMENTAZIONI

- Pagine web statiche HTML CSS (eventualmente js scenario per convalida dati e DOM)
- Realizzazione di applicazioni con tecnologie Application Server Java (JSP,Servlet) (eventualmente PHP) che facciano uso di dati su DBMS

INDICAZIONI PER GLI STUDENTI CON INSUFFICIENZE E PER EVENTUALI ESAMI INTEGRATIVI O DI IDONEITÀ:

OBIETTIVI MINIMI:

1. Conoscere il significato di: oggetto o istanza di una classe e la nozione di ereditarietà tra classi
2. Saper rappresentare e comprendere diagrammi delle classi UML (a difeferenti livelli di dettaglio)
3. Conoscere il significato di sistema informatico e sistema informativo
4. Saper argomentare il fallimento dell' approccio alla gestione dei dati "algoritmico" comprendendo la ridondanza e le anomalie da essa ingenerate.
5. Saper spiegare cosa si intende per indipendenza logica e fisica tra programmi e persistenza.
6. Saper definire DBMS (non limitandosi solo a enunciare le parole da cui deriva tale acronimo)
7. Saper spiegare il modello Relazionale e l'algebra relazionale .
8. Conoscere le fasi di progettazione di una base di dati (saperle collocare nel più ampio sviluppo di un sistema informatico) e comprendere quali di queste sono indipendenti dall'adozione di un particolare modello logico, o meno
9. Progettazione di una base di dati relazionale a partire da requisiti in linguaggio naturale sapendo **giustificare e spiegare le scelte adottate.**

INDICAZIONI METODOLOGICHE PER LO STUDIO INDIVIDUALE ESTIVO:

- Riprendere in mano sul libro, e sugli appunti, le nozioni teoriche elencate negli obbiettivi minimi,
- **Queste parti saranno oggetto di un ampio test complessivo appena iniziato l'anno successivo.**

Per rafforzare le competenze di analisi, progettazione e applicazione tecnologica:

DA SVOLGERE E CONSEGNARE A INIZIO ANNO:

- **DB:** Svolgere per intero, in maniera **completa** e graficamente chiara, due esercizi di progettazione di una base di dati, prestando molta cura a giustificare e spiegare le scelte fatte a corredo del progetto concettuale (corredandolo di ogni vincolo non specificabile nel diagramma ER). Infine dopo aver ottenuto lo schema logico provare a fornirne una traduzione SQL (ripassando ovviamente le istruzioni CREATE TABLE, valutando i tipi opportuni presenti nello standard, e mettendo **tutti** i vincoli progettati di chiave, di integrità referenziale, utilizzare anche CHECK per esprimere vincoli ulteriori di filtraggio) **uno sarà assegnato dal docente l'altro dovrà attenersi un problema di gestione dati inventato e proposto dall'allievo.**
- **HTML – CSS :** svolgere il seguente percorso **acquisendo la certificazione** (dovete registrarvi, bisogna fare i 5 progetti): <https://www.freecodecamp.org/learn/2022/responsive-web-design/>
- **SQL :**
 - sqlbolt.com (già fatto a lezione: **tutto documentando su un file le singole risposte date e il giorno che son state fatte**)
 - <https://www.sql-practice.com> (**tutto: documentando su un file le singole risposte date e il giorno**)

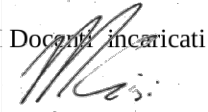
TIPOLOGIA DI PROVE CHE DOVRANNO SOSTENERE :

le prove consteranno in:

- Progettazione di una base di dati relazionali (fornendo il codice sql che crea la base di dati) e svolgimento di alcune query

Mondovì a.s. : 2023-24

Docenti incaricati



I RAPPRESENTANTI DEGLI STUDENTI: