

Programmazione dell'attività didattica di  
Informatica  
Classe IV sez. B a.s. 2017-2018

---

Michele Nunziata  
16 novembre 2017

## **1 Obiettivi Generali dell'Indirizzo di Studi**

Analizzare, dimensionare, gestire piccoli sistemi per l'elaborazione, la trasmissione, l'acquisizione delle informazioni sia in forma analogica che digitale, in applicazioni di vario tipo ma particolarmente rivolte ai settori tecnico-industriali e scientifici. Le conoscenze dovranno essere tali da consentire una agevole partecipazione alla realizzazione di progetti più grandi e alla loro gestione.

### **1.1 Obiettivi specifici**

Completare e consolidare le conoscenze e la competenze acquisite nel corso del terzo anno.

### **1.2 Competenze operative**

- Capacità di utilizzo di un moderno sistema.
- Capacità d'impostare problemi.
- Capacità di verificare la bontà di una soluzione.
- Capacità di organizzare e documentare un lavoro.

- Conoscenza delle metodologie proprie dell'informatica.
- Conoscenza e capacità di analizzare criticamente aspetti tecnici, organizzativi, culturali e sociali dell'uso dell'informatica nei vari settori di applicazione

## 2 Argomenti da trattare

### 2.1 Modulo 1 - Allocazione sequenziale e concatenata di memoria

Viene ripreso e completato lo studio dei tipi di dati offerti dal linguaggio C.

Unità didattica 1 - Tipi di dati.

- Richiami ai tipi di dato standard.
- Le stringhe nel linguaggio C.
- Gli array in C.
- Il tipo struct : vettori di strutture.

Unità didattica 2 - La programmazione modulare.

- L'astrazione procedurale.
- Il concetto di interfaccia : ambiente e passaggio dei parametri.
- Regole di visibilità.
- La programmazione ricorsiva.

Unità didattica 3 - Algoritmi notevoli su array.

- Algoritmi di ordinamento : BubbleSort, ShakerSort, InsertSort, SelectionSort.
- Tipi astratti di dati (ADT con allocazione sequenziale) : stack, coda.
- Algoritmi per la ricerca: sequenziale e binaria o dicotomica.

Unità didattica 4 - Il tipo puntatore e le strutture dati ricorsive.

- I puntatori e l'allocazione dinamica di memoria.
- Confronto tra allocazione statica e dinamica di memoria.
- Le liste: il tipo astratto (ADT) lista.
- Implementazione dinamica dell' ADT lista.

- Implementazione dinamica dell'ADT stack : Lo stack nella chiamata a sottoprogrammi e nella ricorsione.
- Implementazione dinamica dell'ADT coda.
- Le liste circolari, richiuse non circolari e condivise.
- Le liste multiple.
- Gli alberi: alberi con radice, binari, binari bilanciati
- Implementazione dinamica dell'ADT albero binario di ricerca.

Laboratorio L'attività di laboratorio sarà dedicata alla realizzazione degli ADT e al loro utilizzo. Sarà utilizzato il linguaggio C in ambiente Linux.

## 2.2 Modulo 2 - I file nel linguaggio C

Questo modulo costituisce una importante introduzione allo studio delle organizzazioni di dati su memoria di massa.

Unità didattica 1

- L'organizzazione fisica delle memorie di massa e metodi di accesso.
- Record fisico e record logico.
- L'accesso sequenziale, diretto, accesso misto.
- I file sequenziali: apertura, gestione e chiusura di un file.
- I file di testo.
- I file di record : lettura e scrittura di blocchi di dati.
- Accesso diretto ad un file di record.

Laboratorio. Saranno realizzate applicazioni rivolte alla gestione dei files sequenziali con accesso sequenziale e diretto. Particolare attenzione sarà rivolta ai file di testo.

## 2.3 Modulo 3 - La programmazione ad oggetti.

La programmazione orientata agli oggetti, ha avuto nel tempo crescenti consensi fino a divenire il paradigma di programmazione più utilizzato nelle aziende di produzione software.

Unità didattica 1 - Introduzione agli oggetti.

- Nascita e scopo della programmazione ad oggetti.
- Classi di oggetti.
- Definire una classe.
- Creare oggetti : metodi costruttori.
- Il linguaggio UML per la descrizione delle classi e delle istanze.
- Comunicazione ed interazione tra oggetti.
- Il metodo main().

Unità didattica 2 - Information hiding e l'incapsulamento.

- Vantaggi dell'incapsulamento.
- Attributi e metodi di classe.
- Oggetti come parametri.

Unità didattica 3 - Ereditarietà e polimorfismo.

- Ereditarietà.
- Specializzazione, tipi di ereditarietà e classi astratte.
- Polimorfismo.
- Le eccezioni.
- La persistenza.

Unità didattica 4 - GUI ed eventi.

- Le interfacce utenti.
- Oggetti componenti e oggetti contenitori.
- Gerarchie di oggetti e di classi GUI.
- Inserimento ed annidamenti di oggetti GUI.
- La gestione degli eventi.

Laboratorio Saranno realizzate semplici applicazioni OOP in linguaggio Java. Sarà utilizzato l'IDE Eclipse e l'ambiente grafico SWT.

### **3 Metodi**

Le varie unità didattiche saranno realizzate, negli aspetti teorici, con lezioni frontali, nelle quali sarà dato ampio spazio al confronto ed alla valutazione critica.

### **4 Mezzi e strumenti**

Notevole importanza sarà data al lavoro di laboratorio inteso non solo come pura esercitazione pratica ma soprattutto luogo nel quale verificare la bontà delle soluzioni individuate nella parte teorica. Il linguaggio di riferimento sarà il C, mentre per la programmazione ad oggetti sarà utilizzato il linguaggio Java.

### **5 Verifica**

Le verifiche saranno realizzate tramite questionari e colloqui individuali, prove scritte e prove di laboratorio. I questionari saranno svolti alla fine di ciascuna unità didattica e potranno essere seguiti da colloqui individuali ogni volta che se ne ravvisa la necessità. Le prove scritte avranno una cadenza mensile mentre le prove di laboratorio saranno relative agli aspetti salienti delle unità didattiche svolte.

## **6 Valutazione dei colloqui e dei questionari**

### **6.1 Conoscenze**

- Non conosce o conosce solo parzialmente e non sempre correttamente i temi proposti.
- Conosce in modo corretto ma puramente scolastico i temi proposti.
- Conosce ampiamente ed approfonditamente i temi proposti.

### **6.2 Competenze**

- Non sa individuare i concetti chiave.
- Sa analizzare alcuni aspetti significativi.
- Sa analizzare i vari aspetti significativi e individua appropriate soluzioni.

### **6.3 Capacità**

- Non sa esprimere giudizi personali né operare scelte proprie.
- Esprime giudizi e scelte adeguati ma non sempre adeguatamente motivati.
- Esprime giudizi e scelte adeguati ampiamente e criticamente motivati.

## **7 Valutazione delle prove scritte e di laboratorio**

### **7.1 Congruenza con l'argomento proposto**

- Traccia svolta parzialmente.
- Traccia svolta completamente.
- Traccia svolta completamente con soluzioni particolarmente efficienti.

### **7.2 Conoscenza delle strutture informative di base**

- Non sa individuare né utilizzare le strutture di base.
- Le individua e le utilizza con difficoltà.
- Le individua e le utilizza con padronanza.

### **7.3 Capacità di risoluzione delle procedure richieste**

- Non propone alcuna soluzione.
- Propone soluzioni complete o parziali ma documentate.
- Propone soluzioni complete ed efficaci adeguatamente documentate.

## **8 Scrutinio finale**

Saranno ritenuti sufficienti gli allievi in possesso dei seguenti requisiti:

- Conoscenza precisa delle parti principali di un Elaboratore visto a blocchi funzionali.
- Conoscenza del linguaggio di programmazione utilizzato in laboratorio, tale da consentire una corretta traduzione di un procedimento espresso in forma algoritmica.
- Conoscenza degli algoritmi di base.
- Capacità di scrivere e rendere eseguibili programmi risolutivi di semplici problemi.