

ISTITUTO TECNICO INDUSTRIALE STATALE "MAX PLANCK"

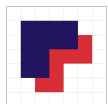
VIA FRANCHINI, 1 31020 - LANCENIGO DI VILLORBA (TV)

ORGANISMO DI FORMAZIONE ACCREDITATO PRESSO LA REGIONE VENETO: COD. N. 218



PROGRAMMAZIONE DI DIPARTIMENTO DI INFORMATICA

Classe 4
Articolazione: Informatica
Materia: Informatica
A.S. 2023-24



Programmazione.

Monte ore settimanale 6

La programmazione di Informatica è stata strutturata in base alle indicazioni delle Linee Guida Ministeriali.

Finalità ed obiettivi generali

Usare un linguaggio tecnico adeguato.

Applicare conoscenze e competenze per la produzione di elaborati pratici.

Essere in grado di rilevare analogie e differenze tra oggetti, eventi e situazioni con adeguati processi di astrazione che conducano a modelli di rappresentazione della realtà.

Finalità educative e didattiche trasversali.

Saper lavorare in gruppo coordinandosi con i propri compagni.

Verifiche e valutazioni (minime)

Primo periodo: 2 verifiche scritte (di cui una sostituibile con 1 verifica orale), 1 verifica pratica

Secondo periodo: 2 verifiche scritte (di cui una sostituibile con 1 verifica orale), 2 verifiche pratiche (di cui una sostituibile con attività progettuale svolta in cooperative learning)

Competenze generali comuni a tutte le unità

Usare un linguaggio tecnico adeguato.

Consolidare valide metodologie per la risoluzione dei problemi (problem-solving)

Finalità ed obiettivi specifici

Approfondire le conoscenze e le competenze circa progettazione, documentazione, testing e manutenzione dei programmi.

Sapere usare correttamente linguaggi di programmazione per applicare sia il paradigma della programmazione imperativa sia quello della programmazione ad oggetti.

Essere in grado di gestire strutture dati complesse e persistenti come i file.

Contenuti irrinunciabili

Utilizzo di un linguaggio sufficientemente appropriato.

Conoscenza teorica dei principi della programmazione ad oggetti (concetti classe, oggetto, polimorfismo, ereditarietà, incapsulamento dell'informazione) e delle caratteristiche di base degli archivi tradizionali.

Capacità di progettare autonomamente semplici programmi secondo il paradigma della programmazione ad oggetti e programmi per la gestione di base di archivi tradizionali (inserimento, variazione, cancellazione, scansione di dati su un file).

Capacità di progettare, se opportunamente guidati, programmi più complessi secondo il paradigma della programmazione ad oggetti, programmi per la gestione di strutture dati dinamiche e di archivi tradizionali, arricchendoli con una sufficiente documentazione e individuando le tipologie di controllo adatte alle attività da svolgere.

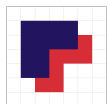
Conoscenze e abilità minime per l'accesso alla classe successiva

Conoscenza teorica dei principi della programmazione ad oggetti e ad eventi.

Capacità di progettare autonomamente algoritmi risolutivi utilizzando il paradigma OOP.

Capacità di utilizzo di strutture dati dinamiche.

Capacità di realizzare progetti con componenti grafici GUI

**Descrittori dei voti delle prove di verifica**

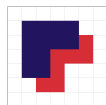
| Voti | Descrittori dei voti delle prove di verifica |
|------|--|
| 1 | Nessuna risposta data |
| 2 | L'alunno ha una scarsissima conoscenza degli argomenti e non sa nel modo più assoluto applicare le sue conoscenze e abilità ai più semplici esercizi. |
| 3 | Scarso: Le lacune e gli errori sono molto gravi e diffusi in termini di conoscenze, abilità ed organizzazione dei contenuti. |
| 4 | Gravemente insufficiente: sono presenti lacune gravi che comportano una visione frammentaria dei contenuti e scarsa comprensione del testo; sono presenti errori diffusi e/o gravi nell'applicazione delle tecniche operative e degli strumenti applicativi. |
| 5 | Insufficiente: non sono stati raggiunti del tutto gli obiettivi minimi in termini di conoscenze e abilità; la comprensione dell'argomento è superficiale; gli argomenti sono conosciuti in modo parziale; vi sono errori non gravi nell'applicazione degli strumenti operativi; ; l'allievo riesce, se opportunamente guidato dall'insegnante, a produrre soluzioni operative. |
| 6 | Sufficiente: sono stati raggiunti gli obiettivi minimi previsti, pur non essendo approfondite le conoscenze e abilità; vi è un limitato numero di errori non gravi e una sufficiente comprensione del testo; l'allievo riesce a produrre corrette soluzioni operative. |
| 7 | Discreto: sono stati raggiunti gli obiettivi minimi previsti, mostrando una certa autonomia nell'operatività e sicurezza nei contenuti; il linguaggio usato è appropriato ed è adeguata la comprensione del testo. |
| 8 | Buono: l'allievo mostra buona autonomia nell'impostazione e nell'organizzazione del lavoro e buona conoscenza dei contenuti, che esprime con lessico adeguato; l'allievo mostra capacità di rielaborazione dei contenuti. |
| 9 | Ottimo: l'allievo mostra conoscenza esauriente ed approfondita dei contenuti che sa esporre con assoluta proprietà di linguaggio; mostra piena autonomia operativa, ottime capacità di rielaborazione personale; sa effettuare opportuni collegamenti interdisciplinari. |
| 10 | Eccellente: l'allievo mostra conoscenza esauriente ed approfondita dei contenuti che sa esporre con assoluta proprietà di linguaggio; mostra piena autonomia operativa, ottime capacità di rielaborazione personale; sa effettuare tutti i collegamenti interdisciplinari e sa apportare validi contributi personali. |

Testo adottato

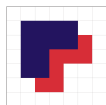
Fiorenzo Formichi, Giorgio Meini, Ivan Venuti

Corso di informatica Terza edizione per informatica vol. 2

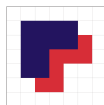
Zanichelli Editore



| Argomento | 1 | Programmazione ad Oggetti | | | | |
|--|--------------|---|--|---|-------|----------------------------|
| Unità | Prerequisiti | Conoscenze | Competenze | Metodi e mezzi | Tempi | Lavoro domestico |
| Evoluzione dei linguaggi, la programmazione ad oggetti | | Paradigmi di programmazione La programmazione ad oggetti Attributi e metodi Incapsulamento, ereditarietà, polimorfismo Cenni di linguaggio UML | Saper applicare le tecniche della programmazione ad oggetti. | LF LI Appunti Laboratorio Manuale | 12 | Studio teorico Esercizi |
| Il linguaggio Java | | Struttura del linguaggio Gli oggetti in Java Indipendenza dalla piattaforma e portabilità JVM: .class e .java. Ambiente di lavoro Eclipse. Oggetto e Classe. Istanziamento di un oggetto, operatore new() allocazione della memoria, deallocazione della memoria: garbage-collector. Attributi-metodi di una classe regole di visibilità degli attributi e dei metodi: public-protected-private. Modificatore static/final incapsulamento dei metodi e degli attributi. Principi dell'information-hiding. Il riferimento this e super. Ereditarietà. Overloading e Overriding. Interfacce ed ereditarietà. Classe Object | Sapere realizzare programmi in Java secondo il paradigma della programmazione ad oggetti | LF LI Appunti Laboratorio Manuale | 12 | Studio teorico Esercizi |

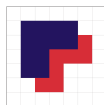


| Argomento | 2 | Strutture Dati Dinamiche | | | | |
|-----------------------------------|---|---|--|---|-------|----------------------------|
| Unità | Prerequisiti | Conoscenze | Competenze | Metodi e mezzi | Tempi | Lavoro domestico |
| Strutture dati in linguaggio Java | Funzioni, tipo di dato, gestione della memoria. | Variabili statiche e dinamiche, concetto di puntatore/reference, stack e heap. Strutture dati astratte implementazione statica e dinamica. Liste lineari definizioni e operazioni fondamentali: inserimento in testa ad una lista, inserimento in coda, stampa del contenuto; ricerca di un elemento; inserimento ordinato; eliminazione di un elemento. Pila, coda, alberi, alberi binari, strutture concatenate. | Sapere gestire alcune strutture astratte (pila, coda) memorizzandole in array e in liste dinamiche. | LF LI Appunti Laboratorio Manuale | 12 | Studio teorico Esercizi |
| Strutture statiche e dinamiche | AAPP | Array e liste dinamiche di oggetti. Pila e coda con strutture dinamiche. Classi per strutture dati | Gestione di liste di oggetti. | LF / LI Appunti Laboratorio Manuale | 10 | Studio teorico Esercizi |
| Funzioni ricorsive | Funzioni, memoria a Stack e RDA | Algoritmi ricorsivi in Java: esempi ed esercizi | Sapere scrivere e simulare il funzionamento delle procedure ricorsive per semplici algoritmi matematici. | LF LI Appunti Laboratorio Manuale | 12 | Studio teorico Esercizi |
| Puntatori e ricorsione | Concetto di puntatore/reference e ricorsione | Operazioni ricorsive per liste e ABR. | Sapere gestire strutture dati dinamiche con l'uso di strutture concatenate e quindi di puntatori/reference. Usare routine ricorsive in linguaggio Java per la creazione e l'attraversamento di un albero binario. | LF LI Appunti Laboratorio Manuale | 8 | Studio teorico Esercizi |



| Argomento | 3 | Gestione dei File, Eccezioni ed Interfacce Grafiche | | | | |
|---|------|---|--|---|----|----------------------------|
| File | AAPP | Oggetto File. Cenni alla gestione dei File sequenziali. Concetto di stream. Operazioni su un file. Serializzazione e File di Oggetti. | Comprendere il funzionamento dell'archiviazione dei dati su file. Saper utilizzare semplici costrutti di gestione degli stream. | LF / LI Appunti Laboratorio Manuale | 6 | Studio teorico Esercizi |
| Eccezioni | AAPP | Definizione di eccezione ed errore JVM e la generazione dell'oggetto eccezione. Costrutto try catch cascata di catch con eccezioni sempre più specializzate. Generazione di eccezioni personalizzate. | Saper gestire le eccezioni | LF / LI Appunti Laboratorio Manuale | 12 | Studio teorico Esercizi |
| Interfaccia grafica in Java e gestione eventi | AAPP | Elementi dell'interfaccia Programmazione guidata dagli eventi. Paradigma Model View Controller (MVC) | Realizzazioni di applicazioni grafiche che utilizzano gli eventi, strutturandole secondo il paradigma MVC. | LF LI Appunti Laboratorio Manuale | 14 | Studio teorico Esercizi |

| Argomento | 4 | REALIZZAZIONE DI UN PROGETTO | | | | |
|-----------|---|---|--|----------------|-------|------------------|
| Unità | Prerequisiti | Conoscenze | Competenze | Metodi e mezzi | Tempi | Lavoro domestico |
| | Tutti gli argomenti oggetti dei moduli precedenti | Modellazione della realtà da automatizzare. Linguaggio Java | Saper sviluppare un progetto completo concreto | LG LM VP | 24 | PR PG |



NOTE

1° **Prerequisiti:**

UUPP Unità didattiche precedenti
AAPP Argomenti precedenti

5° **Verifiche:**

Teorico: VS Verifica scritta
VO Verifica orale
Grafico: VG Verifica grafica
Pratico: VP Verifica pratica

4° **Metodi e mezzi:**

Metodo:

LF Lezione frontale
LI Lezione interattiva
LG Lavoro di gruppo
LM Lavoro manuale o pratico
EC Esercizi in classe

6° **Tempi:**

V Verifica scritta
T Teoria
P Laboratorio, Pratica (Esercitazione)
R-R Recupero, Ripasso

Supporto didattico:

LT Libro di testo
MA Manuali tecnici del laboratorio.
DI Dispense o materiali scaricabili dalla rete (es. datasheets)
AL Appunti della lezione

7° **Lavoro domestico:**

ST Studio teorico
EX Esercizi
PR Calcoli di progetto, Software, ecc ...
PG Produzione relazioni, disegni, ecc...

Supporto tecnico:

LA Laboratorio
VP Videoproiettore
LL Lavagna luminosa
CD Materiali in formato elettronico (CD-Rom, pagine web, ecc.)
PC Personal Computer, relativi pacchetti applicativi, internet.
AO Attrezzatura ordinaria del laboratorio