

**ISTITUTO DI ISTRUZIONE SUPERIORE
DI VITTORIO - LATTANZIO**

**PROGRAMMAZIONE DIDATTICA INDIVIDUALE
ANNUALE A.S. 2018/2019**

MATERIA

INFORMATICA

**CLASSE: 4° B
(Lattanzio)**

DOCENTE: **Prof.ssa M. Lucia Antonini**
I.T.P. : **Prof. S. BELLANTONE**

INQUADRAMENTO DELLA CLASSE

La classe è composta da 17 alunni di cui: uno con sostegno, 2 situazioni BES per DSA e 2 alunni inseriti nel gruppo classe in questo anno scolastico, di cui uno proveniente da altro Istituto, ed uno – ripetente – spostato da altra classe nel mese di Ottobre. Gli studenti, a parte i 2 nuovi inserimenti, appartengono allo stesso gruppo classe dello scorso anno, pertanto gli alunni si conoscono e non ci sono rapporti conflittuali tra di loro, anzi molto buono il dialogo educativo sia tra loro che con il corpo docente.

Conoscendo la classe, si deve purtroppo affermare che le lacune accumulate lo scorso anno e non adeguatamente colmate, si ripercuoteranno indubbiamente sul profitto dell'anno in corso, essendo gli argomenti della disciplina in oggetto una prosecuzione e un approfondimento di quanto appreso lo scorso anno. Per alcuni alunni si evidenzia in questa fase iniziale un atteggiamento non adeguato per continuità di frequenza. Buona comunque nel complesso la motivazione allo studio e l'interesse specifico per la materia, e si evidenzia comunque la presenza di un gruppo molto motivato e con picchi di eccellenza.

OBIETTIVI GENERALI

La finalità principale nella trattazione di questa disciplina segue gli indirizzi riportati nel documento attuativo della *riforma degli Istituti Tecnici*, come riportato nella programmazione di dipartimento, a cui si fa riferimento per la definizione degli obiettivi minimi di apprendimento e nuclei essenziali del programma.

Gli obiettivi formativi trasversali prioritari sono: Utilizzo degli strumenti informatici in relazione all'analisi dei dati e alla modellazione dei problemi comprendere ed utilizzare un linguaggio di comunicazione, elaborare un metodo di lavoro autonomo e costruttivo, unitamente all'acquisizione di una metodologia di sviluppo del problema, alla formalizzazione del procedimento risolutivo e la sua validazione, relativa individuazione delle risorse informatiche necessarie.

L'effettivo conseguimento delle competenze previste è imprescindibile dello svolgimento di una attività di laboratorio focalizzata su una pratica continua di *problem-solving* e/o di *project-work*. È di conseguenza necessario individuare gli strumenti e gli ambienti in modo che, oltre ad essere presenti nel laboratorio scolastico utilizzato per la disciplina, risultino utilizzabili dagli studenti anche per il lavoro a casa.

RISULTATI DI APPRENDIMENTO ATTESI

In riferimento alle linee guida si elencano i risultati di apprendimento individuati per la materia, coerenti qua quanto definito nella programmazione di Dipartimento, a cui si richiama per la definizione degli obiettivi minimi di apprendimento:

Competenze:

- Riconoscere le situazioni in cui utilizzare il paradigma a oggetti
- sviluppare applicazioni informatiche per reti locali o servizi a distanza;
- saper leggere e saper documentare le applicazioni con linguaggio UML
- Utilizzare a livello base la lingua inglese per la consultazione di manuali tecnici del settore
- Utilizzare le risorse in rete per la ricerca autonoma di soluzioni ad eventuali problemi applicativi
- Collaborare al raggiungimento degli obiettivi all'interno del gruppo partecipando all'analisi e alla realizzazione delle soluzioni dei problemi proposti

Conoscenze:

- Sviluppare applicazioni informatiche con programmazione guidata dagli eventi e interfaccia grafica
- Paradigma di programmazione a oggetti
- Linguaggio di programmazione lato client per la gestione locale di eventi in pagine web.
- Lessico e terminologia tecnica di settore anche in lingua inglese

Abilità:

- Progettare e implementare applicazioni secondo il paradigma ad oggetti.
- Progettare e realizzare interfacce grafiche utente
- Progettare e realizzare pagineweb statiche con interazione locale.
- Utilizzare il lessico e la terminologia tecnica di settore anche in lingua inglese

METODOLOGIA D'INSEGNAMENTO

La trattazione dei vari argomenti sarà svolta con modalità il più possibile individualizzata e tenendo conto dei livelli di apprendimento dei vari argomenti, al fine di consentire un'acquisizione delle tematiche e delle competenze piana, graduale ed esauriente.

Il metodo utilizzato è quello deduttivo finalizzato a fornire conoscenze sul piano non solo teorico ma anche operativo con un marcato orientamento al problem-solving e/o project-work. Le lezioni saranno di tipo frontale con la partecipazione della classe nelle osservazioni ed esercitazioni alla lavagna. Per le attività da sviluppare in laboratorio con il supporto e la collaborazione dell'Insegnante Tecnico Pratico.

Le metodologie didattiche utilizzate realizzano un'equilibrata combinazione tra:

- lezioni frontali, dove con l'aiuto degli strumenti disponibili (Computer, Lim in classe, dispense, ecc.) vengono descritti gli argomenti di studio, stimolando nel contempo la partecipazione costruttiva della classe.
- Esercitazioni pratiche e di laboratorio dove vengono provate e sperimentate le abilità progettuali e realizzative acquisite
- Discussioni di casi: saranno sollecitate situazioni di confronto su tematiche inerenti gli argomenti trattati al fine di far emergere problemi, dubbi e contributi utili al rafforzamento dell'azione formativa.
- Attività di analisi e progettazione di soluzioni software.
- Lavori di gruppo durante i quali gli studenti tenderanno a sviluppare diverse strategie formative. L'attenzione principale sarà dedicata agli aspetti di analisi e progettazione in cui sviluppare le competenze di cooperazione, confronto e professionalità. A tal fine saranno proposti semplici problemi per abituare gli allievi a risolvere casi reali progressivamente più difficili. I gruppi saranno invitati a relazionare agli altri studenti il lavoro svolto.

Come strumenti e supporti alla didattica saranno utilizzati :

- il libro di testo: INFORMATICA PER ISTITUTI TECNICI TECNOLOGICI vol, B - Casa Editrice ATLAS – Autori *LORENZI AGOSTINO / MORIGGIA VITTORIO / RIZZI ANDREA*
- materiale in formato elettronico riguardo le guide software, materiali di approfondimento on-line associati ai libri di testo o in file scaricati da internet,
- lezioni ed esercitazioni fornite tramite piattaforma e-learnig, per favorire al massimo l'interattività allievo-docente e la flessibilità dei tempi di studio e assimilazione.

VERIFICHE E VALUTAZIONI

Riguardo il criterio di valutazione, si terrà conto, per ogni singolo studente:

- dell'impegno dimostrato,
- della continuità di partecipazione e di studio della disciplina nel corso dell'anno,
- delle capacità espressive e della qualità e quantità delle nozioni apprese,
- di una visione globale dell'allievo nel suo progredire, tenendo conto di particolari fattori oggettivi che abbiano potuto influire negativamente sull'apprendimento.

I giudizi di valutazione ed i corrispettivi numeri rispetteranno i criteri fissati nella griglia di valutazione approvata dal collegio docenti e dalle indicazioni del Consiglio di Classe.

Gli indicatori di riferimento sono:

per la produzione scritta

- a) correttezza e proprietà di linguaggio
- b) pertinenza alla traccia
- c) conoscenza dei contenuti
- d) coerenza logica
- e) rielaborazione personale

per il colloquio:

- a) correttezza nell'uso dei linguaggi specifici
- b) conoscenza degli argomenti e capacità di individuarne gli elementi fondanti
- c) capacità di effettuare collegamenti
- d) capacità di esprimere giudizi motivati

per l'attività di laboratorio:

- a) cooperazione
- b) autonomia
- c) responsabilità
- d) rispetto delle consegne

La verifica dell'apprendimento si baserà su verifiche scritte, orali e sviluppo di progetti e ricerche.

Se dalle verifiche dovesse risultare la necessità di azioni di recupero per uno o più alunni, queste saranno condotte:

1. riproponendo i concetti più importati non acquisiti,
2. attraverso la costituzione di gruppi di lavoro in cui saranno affiancati alunni con difficoltà ad alunni che hanno invece già raggiunto gli obiettivi verificati.

3. con esercitazioni individualizzate e mirate al recupero della carenza e materiale didattico di supporto, erogato su piattaforma e-learning
4. con interventi mirati al singolo.

RAPPORTI CON LE FAMIGLIE

Gli incontri con le famiglie avverranno con il ricevimento in orario antimeridiano e i 2 incontri generali pomeridiani stabiliti in sede di programmazione delle attività scolastiche.

PERCORSO FORMATIVO

Le unità didattiche prevedono lezioni teoriche e lezioni pratiche di laboratorio

Il percorso formativo segue la programmazione di dipartimento a cui si fa riferimento per la definizione degli obiettivi minimi di apprendimento e nuclei essenziali del programma.

<p>Ripasso Argomenti propedeutici Anno precedente <i>periodo attuazione: Settembre</i></p> <ul style="list-style-type: none"> • Strutture fondamentali di un algoritmo • Operatori aritmetici, di confronto e booleani • Identificatori; variabili; costanti • Tipi di dato; operatori; commenti e documentazione • Funzioni , parametri e valori di ritorno • Ambiente locale e globale
--

<p>Unità didattica 1 : Programmazione orientata agli oggetti (TEORIA) <i>periodo attuazione: Trattazione argomenti Ottobre - Novembre</i></p> <ul style="list-style-type: none"> • Competenze • Descrivere i concetti di base della programmazione ad oggetti • Individuare gli aspetti della metodologia orientata agli oggetti • Definire le classi con attributi e metodi • Descrivere le classi attraverso diagrammi • Applicare i principi della programmazione ad oggetti utilizzando il linguaggio Java 	
<p>Conoscenze</p>	<p>Abilità</p>
<ul style="list-style-type: none"> • Caratteristiche della metodologia OO: Orientamento agli oggetti: oggetti e classi • Incapsulamento – Ereditarietà - Polimorfismo • Dichiarazione delle classi con attributi e metodi - Livelli di visibilità • Disegnare i diagrammi di classe in UML • Conoscere i diagrammi UML per la rappresentazione di Classi, Oggetti, attributi e Metodi • Creazione e utilizzo di oggetti • Attributi e metodi statici • Mascheramento dell'informazione (Information hiding) • Interfaccia degli oggetti • Applicazione dell'ereditarietà e del polimorfismo agli oggetti: Sottoclasse e Gerarchia delle classi • Array di oggetti • Manipolazione di stringhe 	<ul style="list-style-type: none"> • Saper individuare i passi della metodologia orientata agli oggetti • Definire classi con attributi e metodi • Disegnare i diagrammi di classe in UML • Descrivere la gerarchia delle classi con il grafo di gerarchia • Conoscere i diagrammi UML per la rappresentazione di Classi, Oggetti, attributi e Metodi

Unità didattica 2 : Ambiente di programmazione JAVA (TEORIA E LABORATORIO)

<i>periodo attuazione: Trattazione argomenti Ottobre - Dicembre)</i>	
Competenze	
<ul style="list-style-type: none"> • Individuare gli aspetti caratterizzanti di un nuovo linguaggio di programmazione • Progettare la struttura generale di un programma • Utilizzare correttamente la sintassi e i costrutti del linguaggio 	
Conoscenze	Abilità
<ul style="list-style-type: none"> • L'ambiente di programmazione • Compilazione e interpretazione di un programma Java • La struttura di un'applicazione • Identificatori; variabili; costanti • Tipi di dato; operatori; commenti e documentazione • Operazioni su standard input e standard output • Operatori di incremento prefissi e postfissi • Strutture di sequenza, selezione e ripetizione • Array a una e due dimensioni • Array di oggetti • Creazione di oggetti • Applicazione dell'ereditarietà e del polimorfismo agli oggetti • Manipolazione di stringhe • Eccezioni 	<ul style="list-style-type: none"> • Saper scrivere , compilare ed eseguire un programma Java • Saper rappresentare le operazioni di I/O standard • Saper dichiarare le variabili e le costanti • Saper inserire frasi di commento nel programma • Saper utilizzare le strutture di controllo • Saper gestire semplici eccezioni • Dichiarazione, allocazione e utilizzo di array mono e bidimensionali • Saper utilizzare i livelli di visibilità di attributi e metodi • Saper creare sottoclassi applicando l'ereditarietà • Saper applicare l'overriding e l'overloading agli oggetti • Saper manipolare stringhe
<i>Ambiente di sviluppo in Java : Eclipse</i>	
Unità didattica 3 : Strutture dinamiche e gestione dei file (TEORIA E LABORATORIO)	
<i>periodo attuazione: Trattazione argomenti Gennaio-Febbraio</i>	
Competenze	
<ul style="list-style-type: none"> • Comprendere la differenza tra gestione statica e gestione dinamica della memoria • Individuare le soluzioni dei problemi basate sull'uso di liste di dati dinamiche • Analisi di strutture di tipo pila e/o coda • Memorizzare dati su supporti di memoria di massa 	
Conoscenze	Abilità
<ul style="list-style-type: none"> • Creazione dinamica di aree di memoria • Array dinamici • Gestione automatica della memoria • Modalità di gestione delle strutture: pila, coda, lista, albero binario 	<ul style="list-style-type: none"> • Saper individuare le caratteristiche delle strutture dinamiche • Conoscere le tecniche di gestione delle strutture dati dinamiche • Implementare le classi per array dinamici • Implementare le classi per pila, coda, albero • Distinguere i diversi tipi di file • Gestire le diverse modalità di accesso ai file utilizzando il linguaggio Java
Unità didattica 4 : Interfaccia grafica e gestione degli eventi in java (TEORIA E LABORATORIO)	
<i>periodo attuazione: Trattazione argomenti Febbraio-Marzo</i>	
Competenze	
<ul style="list-style-type: none"> • Comprendere l'importanza della progettazione dell'interfaccia per l'utente • Utilizzare gli oggetti per costruire interfacce grafiche • Gestire gli eventi generati dall'utente utilizzando finestre di dialogo e menu 	
Conoscenze	Abilità

<ul style="list-style-type: none"> • Caratteristiche generali dell'interfaccia per l'utente; interfacce a carattere e interfacce grafiche • Gli oggetti dell'interfaccia grafica • Le librerie AWT e SWING • I contenitori e le componenti • La disposizione degli elementi grafici • La gestione degli eventi 	<ul style="list-style-type: none"> • Predisporre un ambiente di sviluppo • Saper progettare semplici interfacce usando gli elementi grafici fondamentali • Impostare le proprietà degli elementi grafici • Impostare il layout • Utilizzare le finestre di dialogo standard • Aggiungere i menu all'applicazione • Saper gestire gli eventi associati ai principali elementi grafici • Gestire gli eventi provocati dal sistema e quelli provocati dall'utente
<p>Unità didattica 5 : Linguaggi web lato client (TEORIA E LABORATORIO) <i>periodo attuazione: Trattazione argomenti Aprile-Maggio</i></p>	
<p>Competenze</p> <ul style="list-style-type: none"> • Progettare , realizzare e gestire pagine Web statiche e dinamiche lato client 	
Conoscenze	Abilità
<p>APPLET</p> <ul style="list-style-type: none"> • Caratteristiche generali di un'applet • Applicazioni e applet • Richiami al linguaggioHTML • Classi Applet e JApplet • Metodi della classe Applet • Attivazione dell'applet da una pagina Web • Passaggio di parametri all'applet • Restrizioni e problemi di sicurezza nell'uso delle applet • Differenza tra applicazione e applet Java • Eventi del mouse • Metodi per disegni • Metodi per le immagini <p>JAVASCRIPT</p> <ul style="list-style-type: none"> • Caratteristiche principali del linguaggio JavaScript • Gestione degli eventi in JavaScript 	<ul style="list-style-type: none"> • Utilizzare le componenti grafiche per costruire interfacce nelle pagine Web • Creare un'applet con la gestione dei suoi metodi principali con disegni e immagini • Scrivere pagine Web che richiamano l'esecuzione di un'applet • Inserire uno script JavaScript nella pagina HTML • Organizzare le istruzioni JavaScript in funzioni • Scrivere il codice per la gestione degli eventi • Gestire l'interazione con l'utente • Immagini

OBIETTIVI MINIMI (comuni a tutte le classi parallele, individuati dal Dipartimento)

Si considerano noti gli argomenti scritti nel modulo 0. Si ritiene fondamentale il conseguimento degli obiettivi minimi per il passaggio alla classe successiva

OBIETTIVI MINIMI (comuni a tutte le classi parallele, individuati dal Dipartimento) ù	
Conoscenze	Abilità
<p>Il linguaggio Java</p> <ul style="list-style-type: none"> • Orientamento agli oggetti: oggetti e classi • Dichiarazione e utilizzo di una classe • Attributi e metodi • Creazione e utilizzo di oggetti • Livelli di visibilità • Mascheramento dell'informazione, • Ereditarietà • Dichiarazione e utilizzo di sottoclassi • Gerarchia delle classi 	<ul style="list-style-type: none"> • Saper individuare i passi della metodologia orientata agli oggetti • Definire classi con attributi e metodi • Disegnare i diagrammi di classe in UML • Conoscere i diagrammi UML per la rappresentazione di Classi, Oggetti, attributi e Metodi • Creare e utilizzare oggetti • Saper scrivere , compilare ed eseguire un

<ul style="list-style-type: none"> • Polimorfismo • La struttura di un'applicazione Java • Compilazione e interpretazione di un linguaggio Java • Operazioni su standard input e standard output • Array ad una dimensione • Array di oggetti • Eccezioni • Gestione automatica della memoria • array dinamici • pila, coda, lista, albero binario • Caratteristiche generali dell'interfaccia grafica • Gli oggetti dell'interfaccia grafica • I contenitori e i componenti • La disposizione degli elementi grafici • La gestione degli eventi <p>Il linguaggio Javascript</p> <ul style="list-style-type: none"> ✓ Caratteristiche principali del linguaggio JavaScript ✓ Gestione degli eventi in JavaScript 	<ul style="list-style-type: none"> • programma Java • Saper dichiarare le variabili e le costanti • Saper inserire frasi di commento nel programma • Saper utilizzare le strutture di controllo • Saper gestire semplici eccezioni • Dichiarazione, allocazione e utilizzo di array monodimensionali • Dichiarazione, allocazione e utilizzo di array di oggetti • Saper utilizzare i livelli di visibilità di attributi e metodi • Saper creare sottoclassi applicando l'ereditarietà • Saper applicare l'overriding e l'overloading agli oggetti • Saper manipolare stringhe • Saper individuare le caratteristiche delle strutture dinamiche • Conoscere le tecniche di gestione delle strutture dati dinamiche • Saper progettare semplici interfacce usando gli elementi grafici fondamentali • Creare i contenitori finestra e pannello • Creare le componenti da inserire nei contenitori • Usare i gestori del layout per disporre gli elementi grafici • Saper gestire gli eventi associati ai principali elementi grafici • Utilizzare le componenti grafiche per costruire interfacce nelle pagine Web • Scrivere pagine Web statiche • Inserire uno script JavaScript nella pagina HTML • Organizzare le istruzioni JavaScript in funzioni • Scrivere il codice per la gestione degli eventi • Gestire l'interazione con l'utente
--	---