

# ISSS MARCO POLO

A.S. 2023/2024

## UDA ORIENTATIVA

### CLASSE III B SIA

Sezioni	
1. TITOLO UDA	LA PROGRAMMAZIONE
2. CRITERI	<p><b>Rilevanza</b> L'Uda ha lo scopo di introdurre il paradigma della programmazione oggetti (OOP) e fornire le competenze per la risoluzione di semplici problemi attraverso l'utilizzo della definizione di classi e gli strumenti della OOP, utilizzando il linguaggio C++. In particolare il corso parte dall'introduzione del concetto di astrazione, per poi illustrare i meccanismi di progettazione e implementazione di sistemi di oggetti interagenti mediante la definizione di classi. La programmazione orientata oggetti è un paradigma di programmazione che si basa sulla creazione e manipolazione di oggetti. Un oggetto rappresenta un'entità del mondo reale e incapsula dati e metodi che operano su quei dati. Questo approccio consente di organizzare il codice in modo più modulare, rendendo più facile la gestione e la comprensione del software. Nella programmazione ad oggetti, ogni cosa diventa un oggetto che può fare delle azioni (chiamate metodi) o possedere delle caratteristiche (chiamate proprietà). Negli ambienti di simulazione gli oggetti possono rappresentare entità fisiche con attributi e comportamenti specifici. Un esempio di oggetto è l'automobile. Per gli studenti sarà possibile interagire con l'oggetto automobile attraverso l'interfaccia testuale per simulare specifici comportamenti: pedale del freno, leva del cambio, volante etc. La programmazione a oggetti è ampiamente usata in contesti</p>

reali, soprattutto in situazioni in cui è necessario modellare entità complesse e interrelate. Nel corso delle lezioni l'aula verrà allestita come "azienda laboratorio", in cui sarà possibile rappresentare e vivere le funzioni proprie di un'impresa reale. In tal modo gli studenti avranno la possibilità di verificare la corrispondenza tra modello simulato e modello reale e al contempo acquisire e utilizzare le competenze formativo-professionali necessarie per affrontare in modo consapevole le scelte lavorative future.

#### **Problematizzazione**

Uno dei principi fondamentali per una buona programmazione software è quello di codificare gli aspetti importanti riguardanti il sistema che si sta sviluppando in modo chiaro e ben localizzato. È un principio "generale" nel senso che può toccare ogni fase dello sviluppo.

Per un progettista è essenziale trattare i problemi di realizzazione di un sistema complesso in maniera il più possibile indipendente: decomporre ogni problema complesso in problemi più semplici ed aggregarne le soluzioni per ottenere la soluzione completa.

Questo approccio risulta essere uno strumento insostituibile anche per il programmatore il quale, attraverso le strutture del linguaggio scelto per implementare le soluzioni, si propone di realizzare l'effettiva separazione dei problemi.

#### **Storicizzazione**

La Programmazione ad Oggetti rappresenta, senza dubbio, il **modello di programmazione più diffuso** ed utilizzato degli ultimi dieci anni.

Le vecchie metodologie come la programmazione strutturata e procedurale, in auge negli anni settanta, sono state lentamente ma inesorabilmente superate a causa degli innumerevoli vantaggi che sono derivati dall'utilizzo del nuovo paradigma di sviluppo.

Eppure, contrariamente a quanto si possa pensare, le origini della Programmazione ad Oggetti sono abbastanza remote: i primi linguaggi "ad oggetti" furono il SIMULA I e il SIMULA 67, sviluppati da Ole-Johan Dahl e Kristen Nygaard nei primi anni '60 presso il Norwegian Computing Center.

Entrambi questi linguaggi adottavano già parecchie delle peculiarità che sono oggi tra i capisaldi della programmazione ad oggetti, come ad esempio le classi, le sottoclassi e le funzioni virtuali ma, a dispetto dell'importanza storica, tali linguaggi non riscosero particolare successo.

Negli anni '70 fu introdotto il linguaggio SmallTalk, considerato da molti il primo vero linguaggio ad oggetti "puro". Lo SmallTalk fu sviluppato in due periodi successivi: inizialmente da Alan Kay, ricercatore dell'università dello Utah e successivamente fu ripreso da Adele Goldberg e Daniel Ingalls, entrambi ricercatori allo Xerox Park di Palo Alto, in California. Ma, ancora una volta, il grande successo tardò ad arrivare. Probabilmente, la ragione

di ciò era da ricercare nel fatto che questo linguaggio (come il Simula) era considerato, per lo più, uno strumento destinato alla ricerca e allo studio più che allo sviluppo.

Fu, insomma, necessario attendere gli anni '80 con l'avvento del linguaggio ADA per assistere alla definitiva consacrazione della programmazione ad oggetti come modello da utilizzare. Probabilmente la vera svolta fu rappresentata dall'avvento del C++, creato da Bjarne Stroustrup.

Tra i più noti linguaggi di programmazione ad oggetti, oltre il Visual Basic .NET, si annoverano il C++, Java, Delphi e C#. L'Object Oriented Programming è oramai diventato il paradigma per eccellenza, avendo rimpiazzato quasi completamente quello procedurale. Uno dei più grandi vantaggi che ha introdotto è stato quello di permettere di vedere l'intero sistema come una collezione di classi, ognuna con un ben definito scopo ed una chiara responsabilità, la cui collaborazione permette di raggiungere l'obiettivo globale dell'applicazione. Strumenti come l'incapsulazione, l'ereditarietà e il polimorfismo, hanno contribuito ad un passaggio importante dall'approccio procedurale basato su procedure e funzioni, ad una migliore suddivisione basata sulle funzionalità. L'eccezionale interesse che gli sviluppatori di tutto il mondo riservano a questo paradigma non è legato a una moda del momento, ma a un'innegabile serie di benefici che la programmazione ad oggetti porta in sé nel miglioramento delle dinamiche di lavoro e nella semplificazione della vita dei programmatori.

#### **Ibridazione**

Dall'analisi dei Big Data alla Cyber Security, passando per lo sviluppo di App sono praticamente infiniti i settori in cui è necessario conoscere i fondamenti della OOP per essere competitivi sul mercato del lavoro.

Per le aziende, comprendere l'evoluzione tecnologica, saper progettare e organizzare sistemi software e applicativi, mantenersi al passo con i tempi rappresentano vere necessità che possono immediatamente diventare opportunità se si possiedono i giusti professionisti del settore della programmazione, che operano in una totale e continua sintonia con i trend di mercato anche più recenti

#### **“Percorribilità” dei saperi**

Uno dei vantaggi della programmazione ad oggetti sui quali si pone spesso poca attenzione è proprio il fatto che si tratta di un supporto del tutto naturale alla modellazione software ed è

	<p>facile immaginare come programmare secondo gli stessi canoni con cui si percepisce il mondo.</p> <p>Essa facilita la vita dello sviluppatore e rende la relazione tra reale e astratto diretta e lineare. La OOP si é imposta in modo quasi esclusivo nel campo dell' informatica e non possederne i fondamenti può risultare un ostacolo insormontabile nell'accesso al mondo del lavoro, soprattutto in un settore come questo, che risulta essere il più competitivo in assoluto.</p>
<b>3. DESTINATARI</b>	Alunni della classe Terza AFM curvatura SIA
<b>4. MONTE ORE COMPLESSIVO</b>	35 ore
<b>Azione AgAPAttri 5. COMPETENZE</b>	<b>1. COMPETENZE ORIENTATIVE</b>
	<input checked="" type="checkbox"/> <b>Competenza di pensiero sistemico</b>
	<input type="checkbox"/> <b>Competenza di previsione</b>
	<input checked="" type="checkbox"/> <b>Competenza di pensiero critico</b>
	<input checked="" type="checkbox"/> <b>Competenza di auto-consapevolezza</b>
	<input type="checkbox"/> <b>Flessibilità</b>
	<input checked="" type="checkbox"/> <b>Competenza normativa</b>
	<input checked="" type="checkbox"/> <b>Mentalità orientata alla crescita</b>
	<input type="checkbox"/> <b>Pensiero critico</b>
	<input checked="" type="checkbox"/> <b>Gestione dei processi di apprendimento</b>
	<input checked="" type="checkbox"/> <b>Senso del futuro</b>
	<input type="checkbox"/> <b>Adattabilità</b>
	<input checked="" type="checkbox"/> <b>Problem solving</b>
	<input type="checkbox"/> <b>Pensiero esplorativo</b>
	<b>2. COMPETENZE TRASVERSALI (PCTO)</b>
	<input checked="" type="checkbox"/> <b>Competenza personale, sociale e capacità di imparare a imparare</b>
	<input checked="" type="checkbox"/> <b>Competenza in materia di cittadinanza</b>
<input checked="" type="checkbox"/> <b>Competenza imprenditoriale</b>	
<input checked="" type="checkbox"/> <b>Competenza in materia di consapevolezza ed espressione culturali</b>	
<input checked="" type="checkbox"/> <b>Competenza collaborativa</b>	
<input checked="" type="checkbox"/> <b>Competenza digitale</b>	

### 3. GREEN COMP

Attribuire valori alla sostenibilità

Difendere l'equità

Promuovere la natura

Definizione del problema

Agentività politica

Azione collettiva

Iniziativa individuale

Altro

### 4. COMPETENZE DISCIPLINARI

**Interagire nel sistema azienda utilizzando i principali concetti relativi all'economia e all'organizzazione dei processi produttivi e dei servizi**

**Ricerca soluzioni efficaci rispetto alle soluzioni date**

**Gestire il sistema delle rilevazioni aziendali: accedere alla normativa pubblicitaria, civilistica e fiscale, con particolare riferimento alle attività aziendali**

**Usare software applicativi secondo procedure e regole aziendali a supporto di processi amministrativi, logistici e commerciali.**

**INGLESE: Saper esporre i contenuti relativi alla progettazione Business Plan con un lessico specifico e adeguato**

**Saper attingere a fonti di informazione digitale**

**Saper produrre presentazioni**

**MATEMATICA: Affrontare situazioni problematiche in contesti diversi avvalendosi dei modelli e degli strumenti matematici più adeguati**

**ITALIANO: Leggere e analizzare testi letterari; personalizzare e aggiornare i nuclei narrativi.**

**Argomentare in modo chiaro pensieri, emozioni e decisioni. Usare il lessico specifico. Produrre presentazioni attraverso codici linguistici diversi.**

**DIRITTO: Saper inquadrare il software sotto il profilo giuridico; saper distinguere i diritti morali e patrimoniali del software; saper confrontare il diritto d'autore e il brevetto**

**INFORMATICA: Saper scomporre un problema in sotto problemi. Saper identificare e modellare le caratteristiche essenziali degli oggetti del mondo reale nel contesto della programmazione, sviluppando abilità nell'astrazione. Saper lavorare in progetti di gruppo, requisito essenziale per lo sviluppo di software complessi.**

## 6. SAPERI

	Conoscenze	Abilità
	<b>ECONOMIA AZIENDALE</b> Il Patrimonio di funzionamento. Le immobilizzazioni immateriali come elementi delle attività del patrimonio. Il software. Il programma di gestione Zucchetti	Determinare e rappresentare il patrimonio di funzionamento. Riconoscere e classificare gli elementi del patrimonio. Individuare i software gestionali e i programmi ERP. Saper utilizzare il programma gestionale Zucchetti
	<b>INGLESE:</b> Business Plan	Memorizzare il lessico tecnico Saper reperire informazioni specifiche e globali nella comunicazione scritta e orale Saper comunicare in modo corretto e idoneo
	<b>MATEMATICA:</b> Sistema binario, passaggio dal sistema binario al decimale e viceversa, Visione del film "Imitation game" e dibattito	Saper convertire un numero dalla base 2 alla base 10 e viceversa
	<b>ITALIANO:</b> Passi scelti del romanzo di formazione di J. D. Salinger "Il giovane Holden" (come stimolo iniziale per la riflessione e il dibattito). Il concetto di sé e l'autostima; la resilienza e la capacità decisionale	Individuare i nuclei tematici di un'opera letteraria e collegarli alle proprie esperienze personali. Approfondire i contenuti oggetto della trattazione. Riconoscere i propri bisogni e le situazioni di benessere e di malessere. Esprimere pensieri, emozioni e aspirazioni, anche in relazione ad un futuro progetto di vita. Affrontare un problema con consapevolezza, responsabilità e autonomia. Sostenere le proprie scelte di vita motivandole
	<b>INFORMATICA:</b>  Concetti Fondamentali dell'OOP Linguaggi di Programmazione Strutture Dati e Algoritmi	Saper differenziare la programmazione ad oggetti da quella procedurale Capacità di progettazione e di uso delle principali strutture di dati, di progettazione e di uso di algoritmi per la soluzione di problemi, Riconoscere le tecniche per la valutazione delle prestazioni degli algoritmi

	Principi di Design del Software	
	<b>DIRITTO :</b> La tutela giuridica del software tra diritto d'autore e brevetto attraverso l'analisi di sentenze	-Capire e riflettere sulle norme e i valori che risiedono dietro le azioni di ogni individuo -Cogliere la rilevanza della concorrenza per il funzionamento del sistema economico, collegandola con la tutela dell'iniziativa economica

### PIANO DI LAVORO DELL'UDA

Fasi / titolo	Attività e strategie didattiche	Strumenti	Esiti/Prodotti intermedi/Compito autentico	Criteri/evidenze per la valutazione	Modalità di verifica /valutazione	Durata (ore)
1.  Introduzione all' <u>UDA</u>	Specificare il tipo di attività proposta e la strategia didattica utilizzata  Brainstorming  Debate  Approfondimento sulla tematica proposta	Indicare gli strumenti, i materiali e i documenti da utilizzare  LIM  <a href="https://www.youtube.com/watch?v=xVtdTsOwY_U">https://www.youtube.com/watch?v=xVtdTsOwY_U</a>	Indicare gli esiti in termini di prodotti intermedi attesi dalla fase  Prerequisiti  Conoscenza di base della programmazione e della realizzazione di algoritmi	Indicare le evidenze della competenza osservabili in azione e nel prodotto intermedio/finale e, se si intende somministrare una prova di verifica, i criteri di valutazione	Specificare: - se si valuta processo e/o prodotto e gli strumenti da adottare (rubriche, checklist; griglie) - se si intende somministrare una prova di verifica (struttura, semistrutturata, non strutturata)	2

			Conoscenza di base del linguaggio c++ (input, output, variabili, selezione, cicli)		Questionario semistrutturato diagnostico	
3. INGLESE	<p>Listening and Reading comprehension dei testi relativi al “Business Plan”</p> <p>Lezione frontale partecipata e lavori a coppie</p>	<p>Fotocopie Materiali multimediali Internet Uso della Lim</p>	<p>Presentazione orale</p>	<p>Competenze espositive dei contenuti con particolare attenzione al linguaggio specifico utilizzato; capacità di produzione di presentazioni</p>	<p>Valutazione della presentazione orale effettuata (griglie competenze e contenuti)</p>	2 ore
ITALIANO	<p>Attraverso la didattica orientativa (anche di tipo narrativo) e laboratoriale il percorso intende ampliare la conoscenza di sé, favorendo la scoperta di attitudini, interessi, punti di forza e debolezza ed accompagna lo/la</p>	<p>J. D. Salinger “Il</p>	<p>Promuovere il benessere psicofisico dello /la studente/essa</p> <p>Individuare interessi, attitudini, aspirazioni</p>	<p>Comprendere ed analizzare testi di vario genere</p> <p>Esporre in modo chiaro informazioni ed opinioni personali</p> <p>Argomentare il</p>	<p>Valutazione sulla base delle griglie adottate in sede dipartimentale</p>	4 ore



	<p>studente/essa verso scelte consapevoli ed autonome.</p> <p>Strategie utilizzate: Lezione dialogata, Brainstorming, Approccio autobiografico, Cooperative Learning, Problem Solving, Flipped Classroom</p>	<p>giovane Holden”</p> <p>Supporti multimediali</p> <p>Internet</p> <p>Uso della LIM</p> <p>Laboratorio di informatica</p>	<p>Operare scelte autonome</p> <p>Tipologia della prova di verifica: Elaborare un ipotetico Progetto di vita attraverso una modalità a scelta dello studente (scritta/orale/grafica/multimediale/)</p>	<p>proprio pensiero, le proprie emozioni ed aspettative</p> <p>Ricerca e4d approfondire</p> <p>Pianificare e realizzare un'attività/compito autentico</p>		
4. MATEMATICA	<p>Lezione frontale partecipata e visione del film</p>	<p>Supporti multimediali</p>	<p>Lavoro di gruppo</p>	<p>Competenze disciplinari</p>	<p>Valutazione del lavoro di gruppo</p>	<p>4 ore</p>

## Note per la compilazione

<p><b>la rilevanza in chiave orientativa</b></p> <p>La rilevanza è il criterio che permette di esplicitare gli oggetti e le strutture della disciplina, ma anche dei suoi metodi di indagine, delle procedure, dei linguaggi. Applicare un criterio di rilevanza significa soprattutto comprendere lo statuto dei saperi e implica, di conseguenza, una riorganizzazione qualitativa delle conoscenze e delle competenze.</p> <ul style="list-style-type: none"><li>• <b>Serve a individuare e definire che cosa insegnare.</b></li></ul>	<p><i>Descrivere il criterio di rilevanza che si intende sviluppare durante il percorso. Perché il tema scelto, i contenuti e l'esperienza sono rilevanti dal punto di vista orientativo? "Anticipano" situazioni di lavoro? L'esperienza si svolge in modo da simulare un contesto operativo? Perché il percorso è finalizzato a conoscere e rappresentare se stessi?</i></p>
<p><b>problematizzare</b></p> <p>Lavorare su "fenomeni", casi di studio e situazioni-problema è un modo per rendere efficace ed interessante i processi di apprendimento.</p> <ul style="list-style-type: none"><li>• <b>Serve a individuare i contesti semantici in cui collocare gli elementi essenziali e rilevanti.</b></li></ul>	<p><i>Descrivere in che modo si intende "problematizzare" i contenuti di apprendimento in chiave orientativa. Il tema-problema viene descritto come uno scenario (anche simulativo) in cui il gruppo-classe è chiamato a sviluppare le questioni e le domande che possono avere una attualizzazione rispetto alla realtà degli alunni e al loro vissuto.</i></p>
<p><b>Storicizzare</b></p> <p>Permette di leggere i saperi in modo non dogmatico, ma evolutivo e genealogico. Apre la strada all'idea che i saperi sono il risultato di un pensiero critico e di una conoscenza "in divenire".</p> <ul style="list-style-type: none"><li>• <b>Serve a posizionare e comprendere i contenuti nelle loro coordinate spazio-temporali.</b></li></ul>	<p><i>Collocare l'esperienza in un contesto in cui i problemi vengono "storicizzati" in relazione alle conoscenze curricolari, fornendo in questo modo uno sfondo orientativo nel tempo e nello spazio.</i></p>
<p><b>Ibridare</b></p> <p>Prima ancora di essere una caratteristica del curriculum, l'ibridazione caratterizza i saperi in quanto contenitori interconnessi e metamorfici.</p> <ul style="list-style-type: none"><li>• <b>Serve a pensare le conoscenze e le competenze come luoghi di "confine" e di attraversamento.</b></li></ul>	<p><i>Individuare e indicare le connessioni con altri campi del sapere in modo da valorizzare la dimensione orientativa dell'esperienza. Non si tratta soltanto di comprendere i collegamenti trasversali tra i vari campi del sapere (letteratura e fisica; storia dell'arte e geometria; scienze naturali e matematica; ecc.) ma anche all'interno di una stessa materia (es. l'intertestualità e la comparatistica negli studi letterari; l'antropologia negli studi classici; ecc.).</i></p>

**Rendere “percorribili” i saperi**

Fondamentale per una didattica orientativa, la “percorribilità” serve a fornire l’orizzonte di senso di quello che si impara; ad integrare conoscenze e competenze evitando una loro perenne contrapposizione e concorrenza.

- ***Serve ad imparare in funzione di un apprendimento autentico e significativo, collegato alla realtà e trasferibile in contesti diversi da quello della scuola.***

*Configurare l’orizzonte di senso del percorso didattico e la capacità delle conoscenze acquisite di trasferirsi in un agire competente, di essere operabili anche in contesti esterni alla scuola.*