



## PROGRAMMA EFFETTIVAMENTE SVOLTO

Anno Scolastico 2021 / 2022

**Classe/Sede:** 4^D2 – sede ITI

**Docente:** prof. Riccardo Crosato

**Codocente (ITP):** prof. Mattia Bedani

**Materia insegnata:** INFORMATICA

**Testi adottati:** Dispense, slide, esercizi e progetti risolti, redatti dai docenti. Tutto il materiale del corso è disponibile nei corsi e-learning sulla piattaforma d'Istituto (<https://www.v-learning.it/iis/course/view.php?id=831>) e, per il modulo sullo sviluppo di app per Android, <https://www.v-learning.it/iis/course/view.php?id=214>).

### CONTENUTI DISCIPLINARI

#### **Modulo 1: Ripasso e approfondimenti (settembre – ottobre, 34 ore ca.)**

- 1.1 – Ripasso su algoritmi e linguaggio C#: programmazione strutturata; metodologia top-down; teorema di Bohm-Jacopini, stub di un progetto C#; tipi di dato predefiniti; stringhe e array; metodi; passaggio di parametri per valore e per riferimento
- 1.2 – Approfondimenti: algoritmi ricorsivi; algoritmi di ordinamento per fusione (merge sort) e partizione (quick sort); algoritmo di ricerca binaria iterativo e ricorsivo; interpolazione di stringhe e using di classi statiche; definizione di classi statiche come raccolta di metodi sugli array; cicli paralleli (ciclo Parallel.For)
- 1.3 – Complessità computazionale e teoria della computazione (cenni): funzione di complessità di un algoritmo, complessità asintotica di un algoritmo e di un problema; notazioni “o grande”, “omega grande”, “theta grande”; problemi polinomiali e non polinomiali; problemi trattabili e intrattabili; classi NP e NP-completi

#### **Modulo 2: Analisi e programmazione orientata agli oggetti in C# (novembre – maggio, 57 ore ca.)**

- 2.1 – Basi della programmazione orientata agli oggetti (OOP): vantaggi e caratteristiche della programmazione ad oggetti; gli oggetti dai punti di vista astratto e concreto; concetti di classe e oggetto; aspetti principali della programmazione orientata agli oggetti (astrazione, incapsulamento, ereditarietà e polimorfismo).  
Creazione e utilizzo di oggetti in C#: definizioni di classi in C#;, operatore new; livello di accesso ai membri di una classe; metodi; costruttori e distruttore; garbage collector e suo ruolo; sovraccarico (overloading) di costruttori e metodi; corpi di espressione in metodi e costruttori; membri statici; cenni ai design patterns (design pattern Singleton). Enumerazioni.
- 2.2 – Proprietà e indicizzatori in C#: definizione di proprietà get/set; proprietà automatiche; inizializzazione delle proprietà automatiche in linea; definizione di indicizzatori; corpi di espressione in proprietà e indicizzatori

- 2.3 – Unified Modeling Language (UML): caratteristiche e applicazioni del linguaggio; diagramma delle classi; relazioni tra classi e rappresentazione in UML; interfacce (definizione e rappresentazione in UML); diagramma UML degli oggetti; diagramma dei package; classi contenitore; classe C# List<> per gestire liste di oggetti
- 2.4 – Classi e metodi generici: definizione di metodi statici generici; inferenza di tipo; definizione di classi basate su un tipo generico; vincoli sui parametri di tipo
- 2.5 – Ereditarietà: ereditarietà e gerarchie di classi; modificatori di accesso legati all'ereditarietà; ridefinizione dei metodi; riferimento alla superclasse; chiamata di un metodo della superclasse; chiamata dei metodi costruttori della superclasse; tipi di ereditarietà; la classe Object; copia superficiale e copia profonda
- 2.6 – Polimorfismo e classi astratte: conformità di tipo, tipo statico e tipo dinamico di un oggetto; conversioni esplicite e implicite, operatori is e as; collegamento statico e dinamico; tabella dei metodi virtuali; classi astratte
- 2.7 – Reflection: metadati di un programma; classi Type, Activator, MethodInfo e ConstructorInfo; esempi di applicazione della reflection (ricavare i tipi definiti in un assembly, creare un oggetto di un certo tipo a run-time, generazione di codice e compilazione a run-time)
- 2.8 – Interfacce: ruolo delle interfacce; interfacce in C#; rappresentazione in UML; interfaccia IEnumerator; conformità tra interfacce e classi; implementazione implicita ed esplicita di più interfacce; ereditarietà multipla tra interfacce; problema dell'ereditarietà a diamante; esempi di utilizzo delle interfacce dell'SDK IComparable<> e IEquatable<>, IEnumerable.

### **Modulo 3: Applicazioni ad interfaccia grafica (ottobre – febbraio, 37 ore ca.)**

- 3.1 – Sviluppo di applicazioni ad interfaccia grafica in C# con Windows Forms: utilizzo del designer di GUI in Visual Studio; controlli comuni: form, pulsanti, etichette, caselle di testo e di controllo; visualizzazione modale e non
- 3.2 – Controlli avanzati e applicazioni a più form: caselle combinate, liste, data grid; controlli contenitori; manipolazione e creazione di controlli a run-time; gestione dei menu e barre degli strumenti; finestre di dialogo; applicazioni a più finestre e passaggio di dati tra finestre; primitive grafiche 2D
- 3.3 – Sviluppo di applicazioni ad oggetti con interfacce GUI e classi

### **Modulo 4: File, linguaggio XML e formato JSON (febbraio – maggio, 15 ore ca.)**

- 4.1 – Gestione dei file e classi C# per la loro manipolazione: concetti generali sui file; formati file testo di interscambio (CSV e a larghezza fissa); concetti di record e tracciato record; classe FileStream; classi wrapper BinaryReader, BinaryWriter, StreamReader, StreamWriter; classi per la gestione del file system (File, Directory, FileInfo, DirectoryInfo); uso dell'istruzione using per gestire le risorse
- 4.2 – XML: caratteristiche del linguaggio; struttura di un documento e tag; regole sintattiche; documenti ben formati e validi; cenni a Document Type Definition (DTD) e XML Schema per definire grammatiche XML; serializzazione in XML di oggetti da C# (classe .NET XmlSerializer, condizioni per serializzare oggetti di una classe, attributi XmlElement e XmlAttribute applicabili ad una classe)
- 4.3 – JSON: sintassi e utilizzi, schema JSON, keyword di validazione; serializzazione in JSON da C# (classeDataContractJsonSerializer); accesso da C# ad un web service che restituisce dati in JSON.

### **Modulo 5: OOP e strutture dati (maggio, 5 ore ca.)**

- 5.1 – Strutture lineari: concetto di ADT (tipo di dato astratto); lista (definizione e operazioni); pila (definizione e operazioni); coda (definizione e operazioni); esempio di implementazione ad oggetti di una lista concatenata
- 5.2 – Strutture non lineari (cenni): albero binario (definizione e terminologia, visite in ampiezza e profondità, albero binario di ricerca); grafo (definizione e terminologia).

### **Modulo 6 (PCTO): Sviluppo di app per Android (aprile – giugno, 23 ore ca.)**

- 6.1 – Linguaggio Java: caratteristiche del linguaggio; differenze rispetto a C#; ambiente di sviluppo Eclipse; risorse e documentazione tecnica su Java e Android

- 6.2 – Introduzione alla piattaforma Android: cenni storici, diffusione, versioni e livelli API; componenti del sistema; sviluppo per Android; kit di sviluppo e installazione su sistemi Windows e GNU/Linux; Android Development Tools; SDK Manager
- 6.3 – AVD (Android Virtual Device): creazione e configurazione di dispositivi virtuali; utilizzo di dispositivi reali; strumenti di gestione e debug (ADB, Android Debug Bridge, collegamento in telnet)
- 6.4 – Struttura di un progetto e sviluppo di una prima app: caratteristiche e funzionalità dell'IDE Android Studio; concetti essenziali e componenti fondamentali di un'app Android; struttura del progetto; analisi del progetto: file per la definizione dell'interfaccia utente (UI, User Interface) e delle risorse; file manifest
- 6.5 – Risorse di un progetto Android: resources e assets; esternalizzare le risorse; unità di misura; supporto di differenti risoluzioni e indipendenza dalla densità; sintassi XML; risorse alternative; tipologie di risorse; file di risorse e classe R; accesso alle risorse dal sorgente Java e da XML; risorse valore; localizzare un'app
- 6.6 – Definizione dell'interfaccia utente: assegnare la UI all'activity; UI designer in Android Studio; widget di base della UI (TextView, EditText, Button, Radio Button, ImageView, ...); disposizioni dei controlli con layout di tipo ConstraintLayout
- 6.7 – Gestione degli eventi: interfacce Listener; varie modalità per implementare un event listener; associare l'event handler da XML; sviluppo di un'applicazione completa con accesso alla rete attraverso la libreria http Volley
- 6.8 – Layout per l'interfaccia utente: layout manager; classi View e ViewGroup; parametri layout\_...; LinearLayout; FrameLayout; RelativeLayout; GridLayout; ConstraintLayout; visibilità delle view; WebView e sviluppo di web app
- 6.9 – Activity (cenni): applicazioni a più Activity; Intent implicite ed esplicite e passaggio/ricezione di dati tra Activity; ciclo di vita di un'activity.

**Formazione per gli stage (maggio, 2 ore):** concetti essenziali su sistemi informativi e informatici, database relazionale e linguaggio SQL, modello client/server e varie tipologie di server; Content Management System (CMS).

**Educazione civica (marzo, 3 ore):** visione di un film e discussione sull'argomento "protezione dei dati personali e la sorveglianza di massa".

Valdagno, 8 giugno 2022

*Firma degli studenti  
rappresentanti di classe*

\_\_\_\_\_  
\_\_\_\_\_

*Firma dei Docenti*

\_\_\_\_\_  
\_\_\_\_\_

*(l'originale di questo documento, firmato dagli insegnanti e dai rappresentanti degli studenti, è depositato in segreteria didattica in data 08/06/2022)*