



PROGRAMMA EFFETTIVAMENTE SVOLTO

Anno Scolastico 2020 / 2021

Classe/Sede: 4[^]D1 – 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/enrol/index.php?id=14>) e, per il corso sullo sviluppo per Android, (<https://www.v-learning.it/iis/course/view.php?id=214>).

CONTENUTI DISCIPLINARI

Modulo 1: Ripasso e approfondimenti (settembre – ottobre, 36 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 e parametri
- 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; programmazione parallela (ciclo Parallel.For); corpi di espressione
- 1.3 – Complessità computazionale e teoria della computazione: 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; problemi decidibili e indecidibili: problema della fermata

Modulo 2: Analisi e programmazione orientata agli oggetti in C# (ottobre – aprile, 74 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 (istanza di una classe); aspetti principali della programmazione orientata agli oggetti (astrazione, incapsulamento, ereditarietà e polimorfismo); enumerazioni
- 2.2 – OOP in C#: definizione di una classe; classi e oggetti; livello di accesso ai membri di una classe; metodi; costruttori e distruttore; sovraccarico (overloading) di costruttori e metodi; garbage collector e suo ruolo; proprietà e indicizzatori; proprietà automatiche; 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; diagramma UML degli oggetti; diagramma dei package; classi contenitore; liste generiche (classe List<>)

- 2.4 – Classi e metodi generici: definizione di metodi statici generici; definizione di classi basate su un tipo generico; vincoli sui parametri di tipo
- 2.5 – Ereditarietà: ereditarietà e gerarchie di classi; 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; 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; ordinamento di raccolte (interfacce IComparable<> e IEquatable<>); interfaccia IEnumerable.

Modulo 3: Applicazioni ad interfaccia grafica (novembre – marzo, 29 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 e liste; controlli contenitori; manipolazione e creazione di controlli a run-time; gestione dei menu e barre degli strumenti; picture box; 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 (marzo – aprile, 32 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; funzioni JSON per JavaScript; serializzazione in JSON da C#, accesso ad un web service

Modulo 5: OOP e strutture dati (aprile – giugno, 8 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)
- 5.2 – Strutture non lineari: albero binario (definizione, visite in ampiezza e profondità, albero binario di ricerca); grafo (definizione e terminologia, rappresentazione dei grafi, problemi e applicazioni classiche, cammini semplici, euleriani e hamiltoniani, algoritmo di Dijkstra su grafi semplici pesati); implementazioni del tipo "grafo" mediante matrice di adiacenza, matrice pesata o liste di adiacenza

Modulo 6 (PCTO): Sviluppo di app per Android (aprile – giugno, 25 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; externalizzare 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: applicazioni a più Activity; Intent implicite ed esplicite e passaggio/ricezione di dati tra Activity; ciclo di vita di un'activity; stati di un'activity.

Valdagno, 4 giugno 2021

*Firma degli studenti
rappresentanti di classe*

Firma dei Docenti

