

PROGRAMMA INFORMATICA IVC/IVD

a.s. 2024/2025 Prof.ssa Chiuchiolo Grazia

☐ **Le basi della programmazione ad oggetti**

☐ Astrarre significa semplificare

☐ Individuazione delle entità

☐ Il concetto di oggetto

☐ Allocazione degli oggetti in memoria

☐ Le classi

☐ **Gerarchie di classi ed ereditarietà**

☐ Ereditarietà singola ed ereditarietà multipla

☐ Caratteristiche e comportamenti

☐ Attributi *static*

☐ **La sintassi di base**

☐ Dichiarazione di classe

☐ Dichiarazione di attributi

☐ Modificatori

☐ Dichiarazione di attributi di classe

☐ Il metodo costruttore

☐ Incapsulamento

☐ C# in pratica

☐ Istanziare un oggetto

☐ **Applicazioni Console**

☐ Aggiunta di classi al progetto

☐ Dichiarazione di metodi

☐ Incapsulamento in C#

☐ **Polimorfismo**

☐ Firma di un metodo

☐ Overloading

☐ Parola chiave *this*

☐ Collaudare le classi

☐ **Ereditarietà in C#**

☐ Superclasse e sottoclasse

☐ Lo statement *base*

☐ Test di classe

☐ Il concetto di Overriding

☐ Virtual e override

☐ **Le classi nella programmazione visuale**

☐ La classe *Form*

☐ Scrivere una classe su più file

☐ Uso di altre classi nella programmazione Windows

☐ Utilizzo di un oggetto in un form

☐ **Introduzione a XML**

☐ XML e HTML

☐ XML in pratica

☐ Correttezza del documento XML

☐ Caratteri speciali nei documenti XML

☐ **Serializzazione di oggetti**

☐ Esempio di serializzazione

☐ Listati classe Automobile e AutomobileTest

☐ *Output della classe AutomobileTest*

☐ Listati classi Proprietario – ProprietarioTest

☐ *Output della classe ProprietarioTest*

☐ Classe XmlSerializer

☐ Listato modificato della classe ProprietarioTest

☐ Output del progetto

☐ Deserializzare un oggetto

☐ Metodo deserializza nella classe ProprietarioTest

☐ **Serializzazione nativa**

☐ Classi serializzabili

☐ Listato della classe ProprietarioBinTest

☐ Analisi dell'output

☐ Deserializzazione binaria

☐ Listato modificato della classe ProprietarioBinTest

☐ Creazione di oggetti da file

☐ **Strutture dati statiche**

☐ Vettori

☐ Ordinamento efficiente di vettori (mergesort)

☐ Il mergesort in C#

☐ Implementazione della classe SortTest

☐ La classe astratta Array

☐ **Matrici regolari**

☐ Grafi e matrici

☐ *Grafi e occupazione di memoria*

☐ Implementazione di un grafo in C#

☐ Implementazione della classe GrafoTest

☐ Output di GrafoTest

☐ Serializzazione di un grafo

☐ Completamento del programma

☐ Matrici irregolari

☐ **Alberi e vettori**

☐ Implementazione di un albero

☐ Visita di un albero

☐ **Introduzione alle interfacce pubbliche**

☐ Ereditarietà multipla

☐ Descrizione e definizione delle interfacce in C#

☐ Implementazione di interfacce

☐ Regole d'uso delle interfacce

☐ Uso delle interfacce multiple

☐ **Strutture dati dinamiche**

☐ vettori dinamici tipizzati

☐ Vettori dinamici nativi in C#

☐ **Introduzione alle collezioni**

☐ **Classi di eccezioni**

☐ Definizione delle classi di eccezioni

☐ **La pila**

☐ Operazioni sulle pile

☐ L'interfaccia IStack

☐ La classe VettStack

☐ Implementazione della classe StackTest

☐ La classe Stack in C#

☐ **La coda**

☐ Operazioni sulle code

☐ L'interfaccia IQueue

☐ La classe VettQueue

☐ Implementazione della classe VettQueueTest

☐ La classe Queue nativa

☐ **Liste concatenate**

☐ Class ListNode

☐ Dal nodo alla lista

☐ Implementazione della classe ListaConcatenata

☐ Implementazione della classe ListaConcatenataTest

☐ Implementazione della classe ListaConcatenata<T>

☐ ListaConcatenataTest per la lista tipizzata

☐ Lista e complessità computazionale

☐ La classe LinkedList<T>

☐ **Strutture dati ottimizzate**

☐ Pila ottimizzata

☐ Coda ottimizzata

☐ **Educazione civica (solo IVC)**

☐ Dazi e informatica

Esercitazioni pratiche e applicazioni delle parti teoriche attraverso l'uso intensivo del laboratorio.

Cagliari 10/06/2025

Grazia Chiuchiolo