



**Ministero dell'Istruzione e del Merito
Istituto di Istruzione Secondaria Superiore
"Alessandro Greppi"**

Via dei Mille 27 – 23876 Monticello B.za (LC)
www.istitutogreppi.edu.it

Programma svolto a.s. 2022-2023

Classe 3IA
Materia: Informatica
Professore Gennaro Ciro Malafronte
ITP: Luca Melcarne

Testi adottati:

[Fundamentals of computer programming with C#](#) di Svetlin Nakov & Co.

Dispense integrative a cura del Prof. Malafronte, reperibili sulla piattaforma O365 con le credenziali del gruppo Info3IA2223:

<https://istitutogreppi.sharepoint.com/sites/Info3IA2223> (3IA) – a cura del Prof. Malafronte

Gli esempi e gli esercizi svolti durante il corso sono disponibili su:

<https://github.com/GreppiDev> – a cura del Prof. Malafronte

Programma

Programmazione Strutturata

Conoscenze

Computazione, problem solving, algorithm design

- Teorema di Böhm-Jacopini

- Architettura del calcolatore (cenni)

- Tipologie di linguaggi

 - Machine Languages

 - Assembly Languages and Assemblers

 - High-Level Languages, Compilers and Interpreters

- The .NET Framework

 - CIL and JIT, Assemblies, Managed Code, Linking, Garbage Collection

- Stages in Software Development

 - Documentation

 - Software Development

- Primitive types and variables

 - Concetto di variabile

 - I tipi predefiniti del C#

 - Valori di default

 - Interi, float, double, decimal

 - Numero di cifre significative

 - Errori di rappresentazione

 - Booleano - boolean.

 - Carattere – char

 - Stringa – string

 - Tipi valore e tipi riferimento, Valori letterali, Var keyword

- Operatori ed espressioni

 - Categorie di operatori

 - Associatività degli operatori

 - Precedenza degli operatori

 - Operatori aritmetici, logici, bit a bit, di uguaglianza e di confronto

 - Type conversion and casting

 - Espressioni

- Console Input e Output

 - Composite Formatting – string.Format(), Console.WriteLine(),

 - String Interpolation

 - String Concatenation

- Conditional Statements

 - Costrutti fondamentali

 - If, if .. else

 - Switch

 - Constant pattern

 - Type pattern

 - Operatore condizionale ?:

- Loops

 - Do while

 - While

 - For

 - break e continue

 - Tipo enumerativo – enum

 - Generazione di numeri casuali

- Array, Struct e Date

 - Definizione di array

 - Array multidimensionali

 - Arrays of Arrays

 - Multidimensional arrays vs jagged arrays (array of arrays)

 - Parallel arrays

 - foreach Loops

 - Struct

 - Array di record (con struct)

 - Array paralleli vs. array di record

- Date
- DateTime - struct
- TimeSpan – struct - un intervallo di tempo
- Parsing di DateTime
- Esempi di DateTime.Parse()
- Esempi con TryParse
- Metodi
 - Generalità sui metodi
 - Passaggio dei parametri
 - Passaggio di parametri di tipi di valore
 - Passaggio di parametri di tipi di riferimento
 - Scambio di tipi di valore
 - Passaggio di tipi riferimento per valore
 - Passaggio di tipi riferimento per riferimento
 - Scambio di due stringhe
 - Passaggio di array mediante ref e out
 - Differenza tra il passaggio a un metodo di una struct e di un riferimento a un'istanza di classe
- Complessità computazionale degli algoritmi.
- Notazioni O(n) e confronto tra funzioni asintotiche
- Problema della ricerca del minimo (massimo) di un array
- Problema dell'ordinamento di un array
 - Algoritmi di ordinamento iterativi
 - Selection Sort
 - Insertion Sort
 - Bubble Sort
- Ricerca di una chiave in un array ordinato (ricerca dicotomica iterativa)
- Ricorsione
 - Caratteristiche comuni di un algoritmo ricorsivo
 - Confronto tra algoritmi di ordinamento ricorsivi e algoritmi iterativi
 - Scrittura di metodi ricorsivi
 - Algoritmi di ordinamento ricorsivi
 - Merge Sort (cenni)
 - Quick Sort (cenni)
 - Torre di Hanoi
 - Ricerca di una chiave in un array ordinato (ricerca dicotomica ricorsiva)

Abilità

Saper codificare un algoritmo in linguaggio C#.

Saper strutturare un algoritmo utilizzando le strutture di base del linguaggio C#

Saper scrivere metodi per la suddivisione dei compiti di un algoritmo

Competenze

Saper risolvere problemi attraverso l'uso del calcolatore, usando il linguaggio C# per codificare algoritmi anche complessi.

Programmazione Object Oriented

Conoscenze

- Creating and using Objects
- Defining Classes
 - Modificatori di accesso
 - Defaults
- Namespaces vs Assemblies
- Capitalization conventions
- Object Oriented Programming (OOP)
 - Implementazione esplicita delle interfacce
 - La classe Object: dove tutto ha inizio
 - Metodi di Object
 - is operator in C#
 - as operator in C#
 - Metodo GetType
 - Confronto tra oggetti: Interfaccia IComparable
 - Confronto tra oggetti: Interfaccia IComparable<T>: Metodo CompareTo e overload degli operatori >, <, >=, <=

- Confronti aggiuntivi: Classe astratta Comparer<T>
- Uguaglianza di oggetti: Interfaccia IEquatable<T>, override di Equals() e GetHashCode() di System.Object, override degli operatori == e !=
- Uguaglianze aggiuntive: interfaccia IEqualityComparer<T>
- Clonazione di oggetti
 - MemberwiseClone()
 - Clonazione superficiale (shallow copy)
 - Clonazione profonda (deep copy)

NET API

<https://docs.microsoft.com/it-it/dotnet/api/>

<https://docs.microsoft.com/it-it/dotnet/api/system.collections.generic>

<https://docs.microsoft.com/it-it/dotnet/api/system>

Esempi notevoli di classi e interfacce utili nelle applicazioni

- List<T> Class
- HashSet<T> Class
- Dictionary<TKey,Tvalue> Class
- Stack<T> Class
- Queue<T> Class

Gestione delle stringhe in C#

Exception Handling

- Exception Hierarchy
- Hierarchy of Runtime Exceptions

Streams and I/O

- File and Stream I/O
- Files and Directories
- Streams
 - Stream architecture
 - Stream adapters

Readers and Writers

Dive into files

- Ottenere informazioni su un file
- FileInfo class
- DirectoryInfo class
- Percorsi assoluti e relativi
- Approccio semplice con i file di testo
- Scrittura su un file di testo - StreamWriter(fileName)
- Lettura da un file di testo - StreamReader(fileName)

File delimitati - CSV file

- Creazione di un CSV
- Lettura da un CSV

Classi di riferimento

- File class
- Directory class

FileStream class

Path class

Esempi di operazioni di I/O comuni

- Copiare una directory e tutto il suo contenuto in un altro percorso
- Cancellare una directory e tutto il suo contenuto oppure cancellare un file
- Eliminare file
- Rinominare una directory o un file oppure spostare una directory o file in un altro posto
- Visualizzare un albero di directory in maniera ricorsiva – emulazione del comando tree di Windows
- Gestione dei caratteri jolly (wildcards)
- Percorsi predefiniti in Windows

FileStream class (file binari)

- Posizionamento
- Lettura
- Scrittura
- File di testo a partire da un file binario
- Lettura e scrittura di file binari mediante BinaryReader e BinaryWriter

Applicazioni con interfaccia grafica (GUI) con Windows Forms

Elementi fondamentali di Windows Forms

- Form, Panel, Button, Label, TextBox

- Concetti di Control e Layout
- Posizionamento di un Control all'interno di un container
- Concetti di Margin, Padding, Anchor, Dock
- Concetto di interfaccia responsive (adattabile alla dimensione della finestra)
- Modello ad eventi
 - Gestione degli eventi (Click, KeyPress, KeyDown)
- Rappresentazione di collection mediante ListView
- Gestione di eventi su ListView (SelectedIndexChanged, MouseDown)
- MenuStrip e ContextMenuStrip
- Gestione di file mediante OpenFileDialog e SaveFileDialog

Abilità

- Saper scrivere classi in C# con campi, properties, costruttori e metodi
- Saper implementare interfacce e classi astratte
- Saper applicare i principi dell'ereditarietà e del polimorfismo per creare gerarchie di classi
- Saper scrivere programmi che si interfacciano con file e stream.
- Saper scrivere applicazioni con interfaccia grafica con Windows Forms

Competenze

- Saper scrivere programmi con tecnica Object Oriented per risolvere problemi anche complessi.
- Saper modellare sistemi anche complessi con la tecnica dell'Object Oriented Programming

Attività di laboratorio

Per ogni unità didattica sono stati sviluppati numerosi esempi in laboratorio, contestualmente alle lezioni teoriche.

Monticello, lì.....

Gli Alunni

Prof. Gennaro Ciro Malafronte

Prof. ITP Luca Melcarne
