Immagine che contiene testo, schermata, Carattere, Elementi grafici

Descrizione generata automaticamente

Immagine che contiene schizzo, disegno, simbolo, logo

Descrizione generata automaticamenteImmagine che contiene Simmetria, stella

Descrizione generata automaticamente

**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. 2024-2025**

Classe 4IB

Materia: Informatica

Professore Pietro Petracca

ITP: Stefania Marsina

Testi adottati:

Dispense reperibili sulle piattaforme web del corso:

<https://istitutogreppi.sharepoint.com/sites/Informatica_4IB_2024_2025>

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

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

***PROGRAMMA DEL CORSO***

# Delegati e Lambda Expressions

### Conoscenze

* 1. Delegates in C#
  2. Invoking Delegate
  3. Multicast Delegates
  4. Covariance and Contravariance of Delegates
  5. Lambda Expressions
  6. Function Delegate and its use with lambda
  7. Action Delegate and its use with lambda
  8. Concetto di Closure

### Abilità

Saper scrivere delegati ed espressioni lambda

### Competenze

Saper scrivere delegati ed espressioni lambda per l’esecuzione di funzioni applicate in contesti di problem solving.

# Eventi

### Conoscenze

* 1. Eventi in C#
  2. Publisher e subscriber di eventi in C#

### Abilità

Saper scrivere eventi creando un modello publisher-subscriber.

### Competenze

Saper riconoscere un modello publisher-subscriber e codificarlo in un contesto di problem solving.

# LINQ

### Conoscenze

* 1. Finalità e uso del LINQ
  2. Vantaggi del LINQ
  3. Espressioni LINQ mediante notazione Fluent (metodi)
  4. Applicazione del LINQ alle Collections
  5. Principali operatori LINQ

|  |  |
| --- | --- |
| Classification | Standard Query Operators |
| Filtering | Where, OfType |
| Sorting | OrderBy, OrderByDescending, ThenBy, ThenByDescending, Reverse |
| Grouping | GroupBy, ToLookup |
| Join | Join |
| Projection | Select |
| Aggregation | Average, Count, Max, Min, Sum |
| Quantifiers | All, Any, Contains |
| Elements | ElementAt, ElementAtOrDefault, First, FirstOrDefault, Last, LastOrDefault, Single, SingleOrDefault |
| Partitioning | Skip |
| Concatenation | Concat |
| Conversion | AsEnumerable, Cast, ToArray, ToList |

### Abilità

Saper scrivere espressioni LINQ con l’uso dei metodi (Fluent API) applicate alle Collections.

### Competenze

Applicare le espressioni LINQ e lambda per risolvere problemi di manipolazione di dati contenuti in Collections.

# Persistenza dei dati: Database, ORM, EF Core

### Conoscenze

* 1. Approccio informale ai database con SQLite
  2. Concetto di database relazionale
     1. Utilizzo di SQLite in applicazioni
     2. Accesso in lettura/scrittura a SQLite, mediante interfaccia grafica con il software DB browser - <http://sqlitebrowser.org/>
  3. Concetto di ORM (Objec Relational Mapper)
  4. EF Core
     1. Concetti di base e utilizzo in applicazioni .NET Core
     2. CRUD operations su un database (Create, Read, Update, Delete)
     3. Creazione di modelli dei dati con EF Core
        1. Tipi di collegamenti tra le tabelle dei database relazionali
        2. Associazioni uno a uno, uno a molti, molti a molti
        3. Notazione Fluent per la descrizione dei vincoli delle entità e delle relazioni
        4. Tipi di dato
        5. Convertitori di valore
     4. Tipi di dato di SQLite (storage classes and data types)
     5. Migrazione di un database da riga di comando di Visual Studio
        1. Comandi Add-Migration e Update-Database

### Abilità

Saper scrivere un data model utilizzando il formalismo delle classi C# e le notazioni specifiche di EF Core.

Saper effettuare la migrazione di un database a partire dal data model.

Saper connettere un’applicazione .NET Core ad un database SQLite.

### Competenze

Saper scrivere applicazioni .NET Core Console che effettuano operazioni CRUD su un database SQLite.

# Multithreading, TPL e Programmazione Asincrona

### Conoscenze

* 1. Thread e threading
  2. Processi e Thread a confronto
  3. PCB e TCB
  4. Quando è opportuno usare i thread
  5. Modello dei thread del C#
  6. Foreground and Background Threads
  7. Passaggio di parametri a un thread
  8. Ciclo di vita di un thread
  9. Sospensione e interruzione di un thread
  10. Concetto di threadpool
  11. Il problema delle risorse condivise tra thread
  12. Race condition
  13. Sezione critica
  14. Tecniche di sincronizzazione tra thread
      1. Interlocked
      2. Data Partitioning
      3. Wait Based Synchronization
         1. Monitor e lock
         2. Mutex
         3. Semaphore
         4. SemaphoreSlim
      4. Algoritmi e modelli notevoli associati ai thread
         1. Modello produttore-consumatore
         2. Algoritmo del Barbiere
  15. Concetto di deadlock
  16. Task Parallel Library (TPL)
  17. Costrutti per la programmazione concorrente con riferimento a thread e TPL
      1. fork/join
      2. join(count)
      3. cobegin/coend
  18. TPL: parallelismo sui dati
      1. Parallel.For
      2. Parallel.ForEach
  19. C# Task
      1. Programmazione asincrona basata su attività
      2. Attached and Detached Child Tasks
      3. Cancellazione di un’attività
      4. Cancellare un task e i suoi figli
  20. Programmazione asincrona con async e await
  21. Iniziare più attività contemporaneamente
  22. Composizione di attività
  23. Attendere le attività in modo efficiente
      1. WhenAll e WhenAny
      2. Flusso di controllo in programmi asincroni
  24. Modelli di programmazione asincrona
  25. I/O di file asincrono

### Abilità

Riconoscere i vantaggi e i pattern di riferimento della programmazione concorrente e asincrona.

Saper utilizzare i thread e i semafori.

Saper utilizzare i Task e i metodi della libreria TPL.

### Competenze

Applicare i concetti e i costrutti della programmazione parallela e asincrona per realizzare applicazioni .NET Core Console che gestiscono contemporaneamente più flussi di lavoro.

# Programmazione di rete in .NET

### Conoscenze

* 1. Concetto di URI e URL
  2. Nomi di dominio e indirizzi IP
  3. Network Reference Models
  4. OSI e TCP/IP
  5. Well known ports
  6. Protocollo HTTP
     1. Richieste e risposte
     2. Request methods
     3. Status codes
     4. Message format
     5. Richieste e risposte HTTP in .NET
        1. Gestione del DNS
        2. Costruzione di una URI
        3. Classe HttpClient
           1. Scaricare del testo dalla rete
           2. Eseguire più richieste Web in parallelo tramite async e await
           3. Analizzare la risposta di una Get http mediante le proprietà della risposta
           4. Scaricamento e salvataggio di una pagina web
           5. Lettura e scrittura di file scaricati da Internet con metodi asincroni
           6. Gestione di file di grandi dimensioni
  7. Extension Methods e librerie in C#
     1. Scrittura di metodi di estensione
     2. Scrittura di librerie
        1. Scrittura di una libreria per la gestione del proxy di rete
  8. JavaScript Object Notation – JSON
     1. Concetti di base
     2. Nested Objects
     3. Nested Arrays
     4. Serializzazione e deserializzazione di oggetti JSON in .NET con System.Text.Json

### Abilità

Saper utilizzare la classe HttpClient per effettuare richieste HTTP (GET, POST, PUT, DELETE) verso un host remoto.

Saper utilizzare la classe HttpClient per gestire le risposte HTTP da un host remoto.

Applicare i pattern della programmazione asincrona alle richieste HTTP.

### Competenze

Saper scrivere applicazioni .NET Core Console che si connettono ad endpoints remoti ed effettuano chiamate e risposte http, utilizzando i principali verbi GET, POST, PUT, DELETE.

# REST API

* 1. REST (REpresentational State Transfer) Api
  2. Principi guida dell’architettura REST
  3. Consumare REST API mediante l’oggetto HttpClient
  4. CRUD operations con REST API
  5. Utilizzo di Postman
     1. Testing di Api con Postman
     2. Collections di Postman
     3. Autorizzazioni in Postman
     4. Postman Mock Server
  6. Esempi di servizi REST
     1. Weather API ( <https://openweathermap.org/api> e <https://open-meteo.com/> )
        1. Recupero dei dati relativi alle condizioni meteo correnti

### Abilità

Saper implementare CRUD operations verso un host HTTP remoto, conforme alle specifiche REST.

Saper consumare REST API di terze parti.

### Competenze

Saper scrivere un’applicazione .NET Core Console in grado di consumare servizi REST di terze parti.

# .NET MAUI

### Conoscenze

* 1. .NET MAUI Getting Started (Conoscenze di base sull’architettura, ambiente di sviluppo, applicazione demo, creazione di una interfaccia grafica in XAML per MAUI)

<https://learn.microsoft.com/en-us/training/modules/build-mobile-and-desktop-apps/>

<https://dotnet.microsoft.com/en-us/learn/maui/first-app-tutorial/intro>

<https://learn.microsoft.com/en-us/training/modules/create-user-interface-xaml/>

* 1. Pagine e Navigazione
     1. Content Page

<https://learn.microsoft.com/en-us/dotnet/maui/user-interface/pages/contentpage>

* + 1. Navigazione gerarchica - NavigationPage

<https://learn.microsoft.com/en-us/dotnet/maui/user-interface/pages/navigationpage>

* + 1. Shell

<https://learn.microsoft.com/en-us/dotnet/maui/fundamentals/shell>

* 1. Tipi di Layout di Pagina
     1. Grid

<https://learn.microsoft.com/en-us/dotnet/maui/user-interface/layouts/grid>

* + 1. AbsoluteLayout

<https://learn.microsoft.com/en-us/dotnet/maui/user-interface/layouts/absolutelayout>

* + 1. FlexLayout

<https://learn.microsoft.com/en-us/dotnet/maui/user-interface/layouts/flexlayout>

* + 1. StackLayout, HorizontalStackLayout, VerticalStackLayout

<https://learn.microsoft.com/en-us/dotnet/maui/user-interface/layouts/stacklayout>

<https://learn.microsoft.com/en-us/dotnet/maui/user-interface/layouts/horizontalstacklayout>

<https://learn.microsoft.com/en-us/dotnet/maui/user-interface/layouts/verticalstacklayout>

* 1. Persistenza in MAUI
     1. Salvare in maniera persistente i dati localmente sul dispositivo.

<https://learn.microsoft.com/en-us/training/modules/store-local-data/2-compare-storage-options>

* + 1. Preferences

<https://learn.microsoft.com/en-us/dotnet/maui/platform-integration/storage/preferences>

<https://learn.microsoft.com/en-us/dotnet/api/microsoft.maui.storage.preferences>

* + 1. Local File System (File System Helpers)

<https://learn.microsoft.com/en-us/dotnet/maui/platform-integration/storage/file-system-helpers>

* + 1. Cache directory

<https://learn.microsoft.com/en-us/dotnet/maui/platform-integration/storage/file-system-helpers#cache-directory>

* + 1. App data directory

<https://learn.microsoft.com/en-us/dotnet/maui/platform-integration/storage/file-system-helpers#app-data-directory>

* + 1. Bundled files

<https://learn.microsoft.com/en-us/dotnet/maui/platform-integration/storage/file-system-helpers#bundled-files>

* 1. Data Binding
     1. Data Binding and MVVM

<https://learn.microsoft.com/en-us/dotnet/maui/xaml/fundamentals/mvvm>

* + 1. MVVM Community Toolkit

<https://learn.microsoft.com/en-us/dotnet/communitytoolkit/mvvm/>

<https://learn.microsoft.com/en-us/dotnet/communitytoolkit/mvvm/generators/overview>

<https://learn.microsoft.com/en-us/dotnet/communitytoolkit/mvvm/generators/relaycommand>

<https://learn.microsoft.com/en-us/dotnet/communitytoolkit/mvvm/generators/inotifypropertychanged>

<https://learn.microsoft.com/en-us/dotnet/communitytoolkit/mvvm/generators/observableproperty>

* 1. Un esempio completo di App .NET MAUI - MAUI Workshop (.NET 8)
     1. <https://github.com/dotnet-presentations/dotnet-maui-workshop>
  2. Local database in .NET MAUI (SQLite con il pacchetto sqlite-net-pcl)
     1. Aspetti teorici:

<https://learn.microsoft.com/en-us/dotnet/maui/data-cloud/database-sqlite>

* + 1. Esempi:

<https://learn.microsoft.com/en-us/samples/dotnet/maui-samples/database-sqlite/>

<https://github.com/dotnet/maui-samples/tree/main/6.0/Data/TodoSQLite>

* + 1. Esercitazione guidata:

<https://learn.microsoft.com/en-us/training/modules/store-local-data/>

* + 1. <https://github.com/MicrosoftDocs/mslearn-dotnetmaui-store-local-data>

### Abilità

Saper descrivere l’architettura di un’applicazione mobile sviluppata in .NET MAUI.

Saper scrivere il codice per implementare ciascuno dei componenti e dei meccanismi di funzionamento descritti nella sezione conoscenze.

### Competenze

Saper realizzare applicazioni Android complete con i componenti di .NET MAUI.

**Attività di laboratorio**

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

Monticello, li, 03/06/2025

Gli Alunni Prof. Pietro Petracca

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

ITP Stefania Marsina

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