

FUTURA

LA SCUOLA PER L'ITALIA DI DOMANI



Finanziato
dall'Unione europea
NextGenerationEU



Ministero dell'Istruzione
e del Merito



Italiadomani
Piano Nazionale di Ripresa e Resilienza



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 4IA
Materia: Informatica
Professore Gennaro Ciro Malafronte
ITP: Luca Melcarne

Testi adottati:

Dispense reperibili sulle piattaforme web del corso: <https://istitutogreppi.sharepoint.com/sites/Info4IA22232> (4IA) – 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 DEL CORSO

Delegati e Lambda Expressions

Conoscenze

- 1.1. Delegates in C#
- 1.2. Invoking Delegate
- 1.3. Multicast Delegates
- 1.4. Covariance and Contravariance of Delegates
- 1.5. Lambda Expressions
- 1.6. Function Delegate and its use with lambda
- 1.7. Action Delegate and its use with lambda
- 1.8. 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

- 1.1. Eventi in C#
- 1.2. 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

- 1.1. Finalità e uso del LINQ
- 1.2. Vantaggi del LINQ
- 1.3. Espressioni LINQ mediante notazione Fluent (metodi)
- 1.4. Applicazione del LINQ alle Collections
- 1.5. Principali operatori LINQ

Classification	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

- 1.1. Approccio informale ai database con SQLite
- 1.2. Concetto di database relazionale
 - 1.2.1. Utilizzo di SQLite in applicazioni
 - 1.2.2. Accesso in lettura/scrittura a SQLite, mediante interfaccia grafica con il software DB browser - <http://sqlitebrowser.org/>
- 1.3. Concetto di ORM (Objec Relational Mapper)
- 1.4. EF Core
 - 1.4.1. Concetti di base e utilizzo in applicazioni .NET Core
 - 1.4.2. CRUD operations su un database (Create, Read, Update, Delete)
 - 1.4.3. Creazione di modelli dei dati con EF Core
 - 1.4.3.1. Tipi di collegamenti tra le tabelle dei database relazionali
 - 1.4.3.2. Associazioni uno a uno, uno a molti, molti a molti
 - 1.4.3.3. Notazione Fluent per la descrizione dei vincoli delle entità e delle relazioni
 - 1.4.3.4. Tipi di dato
 - 1.4.3.5. Convertitori di valore
 - 1.4.4. Tipi di dato di SQLite (storage classes and data types)
 - 1.4.5. Migrazione di un database da riga di comando di Visual Studio
 - 1.4.5.1. 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

- 1.1. Thread e threading
- 1.2. Processi e Thread a confronto
- 1.3. PCB e TCB
- 1.4. Quando è opportuno usare i thread
- 1.5. Modello dei thread del C#

- 1.6. Foreground and Background Threads
- 1.7. Passaggio di parametri a un thread
- 1.8. Ciclo di vita di un thread
- 1.9. Sospensione e interruzione di un thread
- 1.10. Concetto di threadpool
- 1.11. Il problema delle risorse condivise tra thread
- 1.12. Race condition
- 1.13. Sezione critica
- 1.14. Tecniche di sincronizzazione tra thread
 - 1.14.1. Interlocked
 - 1.14.2. Data Partitioning
 - 1.14.3. Wait Based Synchronization
 - 1.14.3.1. Monitor e lock
 - 1.14.3.2. Mutex
 - 1.14.3.3. Semaphore
 - 1.14.3.4. SemaphoreSlim
 - 1.14.4. Algoritmi e modelli notevoli associati ai thread
 - 1.14.4.1. Modello produttore-consumatore
 - 1.14.4.2. Algoritmo del Barbiere
- 1.15. Concetto di deadlock
- 1.16. Task Parallel Library (TPL)
- 1.17. Costrutti per la programmazione concorrente con riferimento a thread e TPL
 - 1.17.1. fork/join
 - 1.17.2. join(count)
 - 1.17.3. cobegin/coend
- 1.18. TPL: parallelismo sui dati
 - 1.18.1. Parallel.For
 - 1.18.2. Parallel.ForEach
- 1.19. C# Task
 - 1.19.1. Programmazione asincrona basata su attività
 - 1.19.2. Attached and Detached Child Tasks
 - 1.19.3. Cancellazione di un'attività
 - 1.19.4. Cancellare un task e i suoi figli
 - 1.19.5. Gestione delle eccezioni con i task
- 1.20. Programmazione asincrona con async e await
- 1.21. Iniziare più attività contemporaneamente
- 1.22. Composizione di attività
- 1.23. Attendere le attività in modo efficiente
 - 1.23.1. WhenAll e WhenAny
 - 1.23.2. Flusso di controllo in programmi asincroni
- 1.24. Modelli di programmazione asincrona
- 1.25. Task-based asynchronous pattern (TAP)
- 1.26. 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

- 1.1. Concetto di URI e URL
- 1.2. Nomi di dominio e indirizzi IP
- 1.3. Network Reference Models
- 1.4. OSI e TCP/IP
- 1.5. Well known ports
- 1.6. Protocollo HTTP
 - 1.6.1. Richieste e risposte
 - 1.6.2. Request methods
 - 1.6.3. Status codes
 - 1.6.4. Message format
 - 1.6.5. Richieste e risposte HTTP in .NET
 - 1.6.5.1. Gestione del DNS
 - 1.6.5.2. Costruzione di una URI
 - 1.6.5.3. Classe HttpClient
 - 1.6.5.3.1. Scaricare del testo dalla rete
 - 1.6.5.3.2. Eseguire più richieste Web in parallelo tramite async e await
 - 1.6.5.3.3. Analizzare la risposta di una Get http mediante le proprietà della risposta
 - 1.6.5.3.4. Scaricamento e salvataggio di una pagina web
 - 1.6.5.3.5. Lettura e scrittura di file scaricati da Internet con metodi asincroni
 - 1.6.5.3.6. Gestione di file di grandi dimensioni
- 1.7. Extension Methods e librerie in C#
 - 1.7.1. Scrittura di metodi di estensione
 - 1.7.2. Scrittura di librerie
 - 1.7.2.1. Scrittura di una libreria per la gestione del proxy di rete
- 1.8. JavaScript Object Notation – JSON
 - 1.8.1. Concetti di base
 - 1.8.2. Nested Objects
 - 1.8.3. Nested Arrays
 - 1.8.4. 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

- 1.1. REST (REpresentational State Transfer) Api
- 1.2. Principi guida dell'architettura REST
- 1.3. Consumare REST API mediante l'oggetto HttpClient
- 1.4. CRUD operations con REST API
- 1.5. Utilizzo di Postman
 - 1.5.1. Testing di Api con Postman
 - 1.5.2. Collections di Postman
 - 1.5.3. Autorizzazioni in Postman

- 1.5.4. Postman Mock Server
- 1.6. Mock Server di REST API con il servizio Mockaroo - <https://www.mockaroo.com/>
- 1.7. Esempi di servizi REST
 - 1.7.1. Weather API (<https://openweathermap.org/api> e <https://open-meteo.com/>)
 - 1.7.1.1. Recupero dei dati relativi alle condizioni meteo correnti
 - 1.7.1.2. Previsioni meteo a 7 giorni e previsioni dettagliate a 48 ore
 - 1.7.2. Bing Maps API (<https://docs.microsoft.com/en-us/bingmaps/rest-services/>)
 - 1.7.2.1. Geocoding by address
 - 1.7.2.2. Geocoding by query
 - 1.7.2.3. Inverse Geocoding
 - 1.7.2.4. Static maps
 - 1.7.2.5. Route Calculation
 - 1.7.3. MediaWiki API
 - 1.7.3.1. MediaWiki Action API
 - 1.7.3.1.1. Ricerca fulltext in Wikipedia
 - 1.7.3.1.2. Estrazione del summary di una pagina da Wikipedia
 - 1.7.4. MediaWiki REST API
 - 1.7.4.1. Trovare le pagine che trattano un determinato argomento
 - 1.7.4.2. Trovare il summary di una pagina contenente le parole da cercare
 - 1.7.4.3. Trovare le sezioni di una pagina di Wikipedia
 - 1.7.5. Utilizzo di un parser per wikitext
 - 1.7.5.1. Scrittura di una libreria per l'estrazione del testo leggibile da un testo in formato wikitext

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.

Servizi Cloud (Microsoft Azure)

- 1.1. Generalità su Azure
- 1.2. Sottoscrizione di Azure per studenti
- 1.3. Azure Cognitive Services
 - 1.3.1. Speech to text
 - 1.3.1.1. Scrittura di applicazioni .NET Core Console che utilizzano il servizio speech to text di Azure
 - 1.3.2. Text to speech
 - 1.3.2.1. Scrittura di applicazioni .NET Core Console che utilizzano il servizio text to speech di Azure

Abilità

Saper configurare un servizio cognitivo come il text to speech o lo speech to text su Azure

Competenze

Saper sviluppare un'applicazione .NET Core Console che sfrutti i servizi cloud di Azure per fornire funzionalità evolute nell'ambito del riconoscimento vocale.

.NET MAUI

Conoscenze

- 1.1. .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/>

1.2. Pagine e Navigazione

1.2.1. Content Page

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

1.2.2. Navigazione gerarchica - NavigationPage

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

1.2.3. TabbedPage

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

1.2.4. FlyoutPage

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

1.2.5. Shell

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

1.3. Tipi di Layout di Pagina

1.3.1. Grid

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

1.3.2. AbsoluteLayout

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

1.3.3. FlexLayout

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

1.3.4. 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>

1.4. Persistenza in MAUI

1.4.1. Salvare in maniera persistente i dati localmente sul dispositivo.

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

1.4.2. 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>

1.4.3. Local File System (File System Helpers)

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

1.4.4. Cache directory

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

1.4.5. App data directory

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

1.4.6. Bundled files

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

1.5. Data Binding

1.5.1. Data Binding and MVVM

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

1.5.2. 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>

1.6. Un esempio completo di App .NET MAUI - MAUI Workshop (.NET 7)

1.6.1. <https://github.com/dotnet-presentations/dotnet-maui-workshop>

1.7. Local database in .NET MAUI (SQLite con il pacchetto sqlite-net-pcl)

1.7.1. Aspetti teorici:

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

1.7.2. 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>

1.7.3. Esercitazione guidata:

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

1.7.4. <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.

Programmazione lato server di REST API - Minimal API.NET

1.4. Classe HttpListener

1.4.1. Scrittura di un semplice server che accetta richieste HTTP mediante la classe HttpListener

1.5. Evoluzione del linguaggio C#

1.5.1. Principali novità di C# 8

1.5.1.1. Using declarations

1.5.1.2. Nullable reference types

1.5.1.3. Null-coalescing assignment

1.5.2. Principali novità di C# 9

1.5.2.1. Record types

1.5.2.2. Top level statements

1.5.3. Principali novità di C# 10

1.5.3.1. Record Structs

1.5.3.2. Global and implicit using

1.5.3.3. File scoped namespaces

1.6. Sviluppo di RESTful API con Minimal API .NET

1.6.1. Minimal API .NET a partire da un template ASP.NET Core Web API

1.6.2. Concetto di Server HTTP

1.6.2.1. Kestrel e IISExpress servers

1.6.3. Configurazione di lancio di una Minimal API .NET Core

1.6.4. Funzionamento di una Minimal API .NET Core

1.6.5. Gestione delle rotte

1.6.6. Utilizzo di SQLite in Minimal API .NET Core

- 1.6.7. Introduzione informale ai database relazionali operanti secondo il paradigma client/server
 - 1.6.7.1. Il database MySQL/MariaDb
 - 1.6.7.1.1. Installazione di MariaDb attraverso il gestore di applicazioni Laragon
 - 1.6.7.1.2. Avvio e Stop di MariaDb
 - 1.6.7.1.3. MySQL monitor (interfaccia console)
 - 1.6.7.1.4. HeidiSQL (interfaccia grafica)
 - 1.6.7.1.5. MySQL WorkBench (interfaccia grafica)
 - 1.6.7.1.6. Gestione delle credenziali dell'utente root
 - 1.6.7.1.7. Utilizzo di MariaDb/MySQL con EF Core
 - 1.6.7.1.8. Collegamento di un'applicazione Minimal API .NET Core con un DBMS
 - 1.6.7.1.9. Migrazione del database per MariaDb/MySQL
 - 1.6.7.1.9.1. Migrazione con Code First Approach (solo per sviluppo e testing)
 - 1.6.7.1.9.2. Creazione del database mediante uno script SQL (in produzione)
 - 1.6.7.1.10. Gestione degli account (diversi da root)
 - 1.6.7.1.10.1. Creazione e rimozione di account
 - 1.6.7.1.10.2. Gestione dei permessi degli account
 - 1.6.8. View Model per i dati (Data Transfer Object – DTO)
 - 1.6.9. Documentazione automatica delle API mediante Swagger
 - 1.6.10. Gestione degli Endpoints (Handlers)
 - 1.6.11. Binding dei dati con le Minimal API
 - 1.6.11.1. Binding predefiniti
 - 1.6.11.2. Cenni a Custom Binding con TryParse oppure con BindAsync
 - 1.6.11.3. Filtri su Minimal API
 - 1.6.12. ASP.NET Core Service Lifetimes
 - 1.6.12.1. Singleton
 - 1.6.12.2. Scoped
 - 1.6.12.3. Transient
 - 1.6.13. Data Validation
 - 1.6.13.1. Fluent Validator
 - 1.6.13.2. FluentValidator with ASP.NET Core

Abilità

Essere in grado di sviluppare Web API REST che gestiscano diversi Endpoints.

Saper creare un database a partire dalla Migration di un Data Model.

Competenze

Saper sviluppare un servizio di Web REST API mediante la tecnologia .NET Core con le Minimal API .NET Core.

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

