Week two notes

Monday 4/29/19

* Disposable
* Using block
* Serialization, file I/O, XML, JSON
* Review async
* SoC, KISS, DRY, CML comments

Unmanaged resources

* Open file
* When we want control over when that resources is freed

IDisposable

* Dispose has to be called by programmer if this interface implemented
* Using statement/block calls dispose method
* “Disposable pattern” -look up best practice for

Using statement/block

Using(var httpClient=newHttpClient())

{

//open file

} //file is closed

Separation of concerns

* Loose coupling
* Maintainability
* Unit testability

DRY - Don’t Repeat Yourself

Documentation/documentation tools like DocFX(VS,VS Code parse these for hover-tooltips)

//regular comments

///<summary> adds two integers</summary>

///<remarks></remarks>

/// <param name=”a”>The First integer</param>

///<returns>The sum</returns>

///<throws name=” “>when </throws>

Public int Add(int a, int b)

{

}

Serialization

Writing data to a file?

* Requirements
* Pick a good enough format to reconstruct object from data you output to file
* Formats
  + XML
  + JSON
  + YAML
  + Roll your own text(avoid this when you can)
  + Rol your own binary(also avoid)
  + CSV
* XMLSerializer (old, deprecated, doesn’t support generics)
* JSON.NET aka Newtonsoft JSON (popular json for serialization using .net)

NuGet

.csproj

PackageReference

Under EACH project right click dependencies>manage NuGet packages

(have to install newtonsoft for each project/solution)

GAC (Global Assembly Cache)-global storage for all third party .dll assemblies on your computer

All methods that call async have to (async “infects” everything else)

Process for when writing asynchronous code (best when we do disk/network access,)

1. Look for method that ends in “Async” instead of the regular one
2. Await the returned task
3. This method needs “async” modifier
4. The return type is wrapped in a Task<…>
5. By convention, your async method should itself be named “….Async”

Using git in visual studio 2019

Tuesday 4/30/2019

SQL (structured query language)

* relational
* table, columns with data types, rows
* language for talking to data
* many variants: MySQL, SQLite, Transact-SQL(T-SQL, aka SQL Server), Oracle SQL, SAQL, PostgreSQL, we mainly stick to Transact-SQL(T-SQL, aka SQL Server)
* server software for SQL “SQL server” not micrsofts, referes to any servers that can host SQL databases

we created Azure sql database

named guerrero1904sql>resource group training-rg>location south central US>no elastic pool>basic>sample data(leave none for future)

sql commands

* --comments--
* --many statements in one file like this…
* The execute button F5 will run all the statements and show all the outputs
* When we don’t want to run the whole file, we highlight what we want to run and then hit F5
* SQL has many commands/statements
* The first category of them is called DML (Data Manipulation Language)
  + Most complicated/important DML statement is the SELECT statement
  + \* selects all rows/columns from a table