

Notes

• .NET Framework

by

Hussain Siz.

managed - can be handled in run time.



Nanita

• .Net - It is a software environment for building and executing run time managed applications on different architectures of the underlined systems (multiple architectures.)

It consists of (.net)

i) Common Language Runtime [CLR]

It handles datatypes & manages execution of code at run-time for .Net Application on top of the services offered by its host.
It provides implementation.

ii) Common Type System :- [CTS]

A .Net datatype which is also called a managed type is either a directly accessible value type or a reference type which can only be accessed through an indirection.

CTS offers A Cross programming language supported for primitive value types & for implementing user defined reference & value types in an object oriented manner.

i) Virtual Execution System - [VES]

A set of related managed types are compiled as a unit of deployment called assembly which contains their metadata (binary description) and the intermediate language for codes is (binary instructions) called managed codes for their methods.

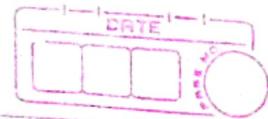
VES offers a cross machine architecture support for loading assemblies by translating the managed code of a method to its equivalent & possibly safe machine code just in Time (JIT) of its invocation.

Implementation of JIT Compiler.

2) Base Class Library [C BCL] :-

It is a collection of standard assembly containing managed types (which allow) allowing a .Net application to consume services offered by

a) Runtime - which include



support for built-in data types like arrays, reflection, & generic data types.

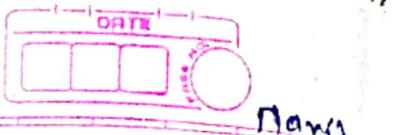
which includes support for basic I/O & OPI, multithreading, file access & network communication.

3) C# Programming Language:-

It is high level programming language designed specifically for writing code that targets CLR. It has following important features.

a) It offers C++ like but more expressive syntax based upon the CTS with optional support for pointers in unsafe context.

b) It is primarily object oriented based on single state inheritance with additional support for generic programming (version 2.0), functional programming (F#)



C# built-in types :- Primitive

C# Name	BCL Name	Identifier
- bool	Boolean	true/false value
- char	Char	single unsigned character.
- byte/ubyte	Byte/UInt8	8 bit unsigned signed int value.
- short/ushort	Int16/UInt16	16 bit unsigned signed int value.
- int/int32	Int32/UInt32	32 bit unsigned and signed int.
- long/ulong	Int64/UInt64	64 bit unsigned & signed integer.
- float	Single	32 bit single precision (real) value.
- double	Double	64 bit Double precision value.

2>

Non-Primitive

decimal	Decimal	96 bit high precision real value.
string	String	Character sequence reference.
Object	Object	non specific (polymorphic) type reference.

Class

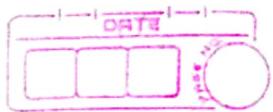
(Notes)

1. It defines a nullable reference type commonly used for data abstraction which requires extension.
2. Memory is dynamically allocated for its instance using `New` operator which also calls the constructor.
3. Its instance fields are only accessible through an injection which is slower.
4. memory assigned to its locally identified instance for its reference is deallocated during next garbage collection which occurs after allocating method returns.
5. It can include a parameterless ctor which is implicitly defined in absence of any explicitly defined Ctor.
6. It can extend any one other class and can be used as base for other classes.
7. It supports direct implicit conversion to a type in `from`.

Struct

(Notes)

1. It defines non nullable reference type commonly used for data abstraction which requires efficiency.
2. Memory is automatically allocated making `New` operator optional which can be used to call constructor.
3. Its instance field is accessible through direct addressing which is faster.
4. memory assigned to its locally identified instance is deallocated immediately after its allocating method returns.
5. It can not include a parameterless ctor which is logically supported as a zero field initializer.
6. It implicitly extends `System.ValueType` & can not be used as base for other types.
7. It supports explicit implicit conversion to a type in `from` indirectly through boxing. (Ans)



Access modifiers

Accessibility in

Current Project accessibility

External accessibility

1. private (default)

none

none

2. Internal

all

none

3. Protected

derived

derived

4. Internal protected

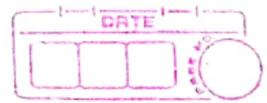
all

derived

5. public

all

all



Narita

Object Identity :-

two objects are considered to be identical if they refer to the same instance in the memory.

In either object x is identical to object y can be determined from expression

`System.Object.ReferenceEquals(x, y)`

Object Equality :-

two objects are considered to be equal if they refer to instances of class and have matching data in the memory.

In either object x is equal to object y can be determined from expression.

`x.GetHashCode() == y.GetHashCode() && x.Equals(y)`

State-fields



(Narita)

Abstract Class

1. It defines non-activatable reference type for aggregating the common type of state shared by different types of objects.
2. It can define instance fields.
3. It can contain public as well as non-public members which may or may not be abstract.
4. It can extend max exactly one other class which may or may not be abstract.
5. A class can only inherit from single abstract class.
6. Struct can not inherit from abstract class.

Why C++ does not support Multiple inheritance.

Behaviour-methods

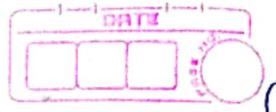


(Narita)

Interface

1. It defines non-activatable reference type for aggregating the common type of behaviour shared by different types of objects.
2. No instance fields.
3. It can only contain public abstract members.
4. It can extend multiple other interfaces.
5. A class can inherit from multiple interfaces.
6. Struct can inherit from multiple interfaces.

(Narita)



Write

Delegate:-

It is reference type whose instance can be bound (attached) to one or more methods each with particular list of parameters type & return type.

It is used for passing method in an argument so that it can be invoked using that argument.

It has following important characteristics.

1) A delegate object is a instance of a compiler generated class which extends System.MulticastDelegate

2) A delegate object can be bound (using assignment or addition) to any method whose signature is compatible with delegate type while allowing covariant (base-by-derived) substitution for return type and contravariant (desired by base) substitution for each parameter type provided their reference type.



③ A delegate object exposes invoke method. A call to which its equivalent to invoking the method binds to that delegate object.

Box:

enclosing a value type data within an object on the heap is called boxing.

the instruction (Box) for this operation is automatically inserted by the compiler in implicit conversion of value type to a reference type.

Unboxing:

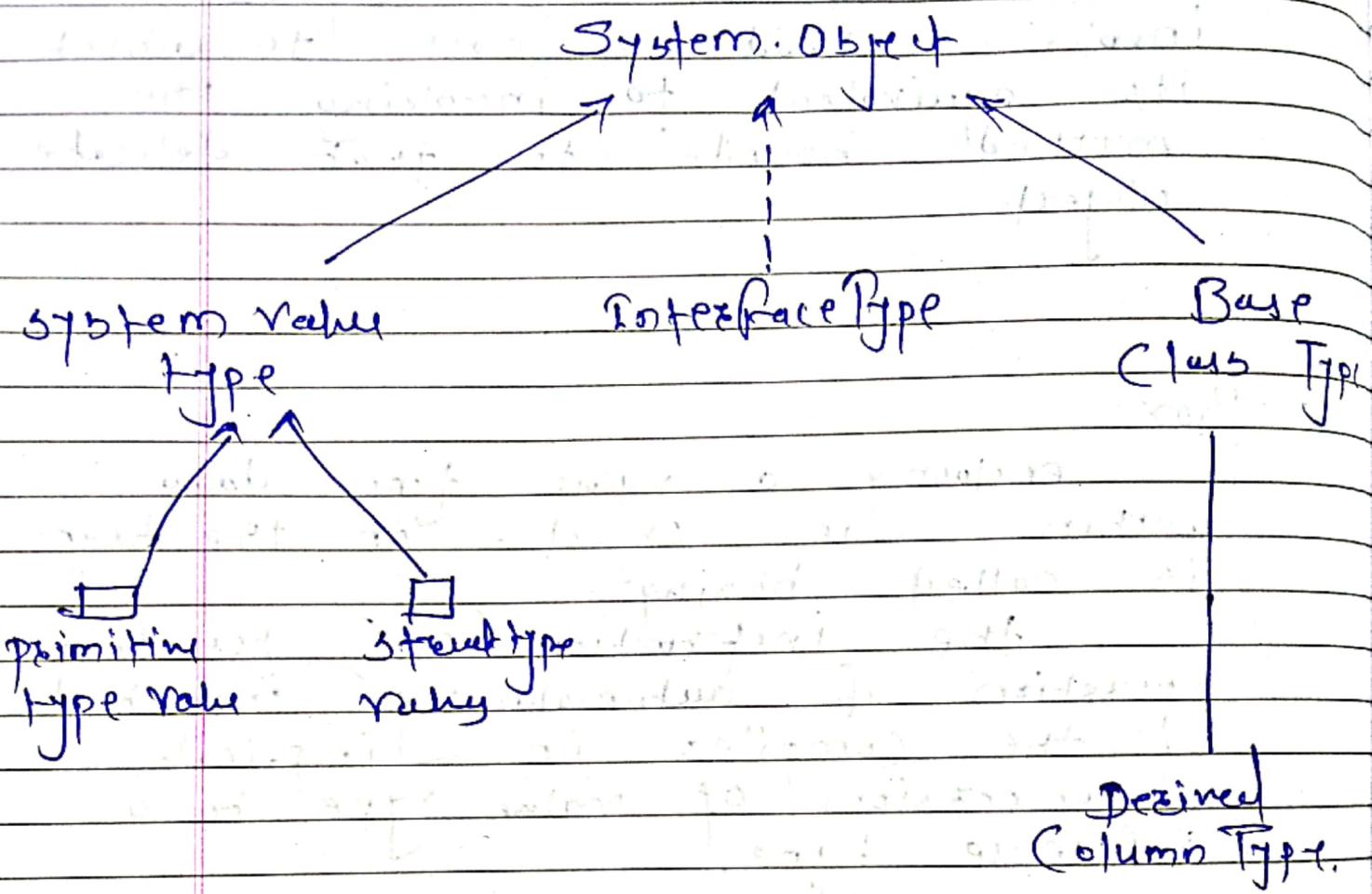
Extracting a value type data from its boxed object from heap is called unboxing.

The instruction (unboxing) for this operation is automatically inserted by compiler in explicit conversion from reference type to value type.

(Ans)



Implicit Conversion in C#



direct conversion →

conversion with boxing ↗

CLR Generics +

support offered by the intermediate language for implementing typesafe code which can be reuse with different datatypes. It enables a compiler to identify matching datatypes and to eliminate unnecessary type conversions. A generic declaration contains at least one open type parameters which can be substituted by any known type at compile time with following characteristics.

- 1) A type parameter is replaced by its substituted type at run time and as such at compile time it is treated as `System.Object` type as by default.
- 2) A Type parameter `T` can appear in declaration with following constraints which are enforced or checked at compile time.
 - i) `+ : struct/Class` :- It indicates that `T` can only be substituted by value/reference type.

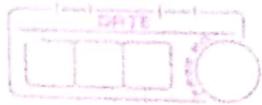
i) T:R :- It indicates that T can only be substituted by the type which supports implicit conversion to (or inherits from) a reference type R. And as such members of R can be applied to T.

ii) T::new() :- It indicates that T can only be substituted by a type which supports a parameterless constructor and as such new operator can be applied to T with zero arguments.

~~30-11-2023~~

3) A generic interface I to the type parameter T can be declared as, $I<\text{out } T>$ to indicate that T only appears as an output type in members of I and as such I is convenient once T for reference.

Convenient base by derived subtypes
Controversial derived by base



i) $I < \in T$:-

To indicate that
T only appears as an Ftp
type in members of I and
as such I is
over T for reference T.

Iteration for non Array source in C#

foreach (E item in source)

process(item)

}

equivalent to [compiler compiles]

for (var e = source GetEnumerator(), eMoveNext

{

E, item = e.Current,

process(item),

}

using `seq` =

`seq<List<v>`
↑
(indexKey)

`seq List<v>`

`seq<Set<k>`
↑
(identity)

`seq<Hashset<k>`
()

System.Collections.Generic

`IEnumerable<out T>`
↓
(Iterate)

`ICollection<T>`
↑
(Addition)

`seq<IDictionary<k,v>`
↑
()

`seq<SortedSet<k>`
()

`seq<Dictionary<k,v>`

`seq<SortedList<k,v>`

`seq<SortedDictionary<k,v>`

(adding, searching
One at a time)

$O(1), O(n), O(n)$

Language Integrated Queries [LINQ]

It is runtime support offered by .Net for syntactical support offered by C# for retrieving data from different data sources using declarative statements.

The programming model offered by link are based on

i) Query operators :-

These are standard methods which a data source must support so that it can be queried using link

The C# compiler translates a link statement into a chain of calls to link operators

ii) Enumerable Extensions :-

These are query operators defined implemented as extension methods in System.Linq.Enumerable class for System.Collections.Generic.IEnumerable interface which is commonly implemented by in memory data sources.



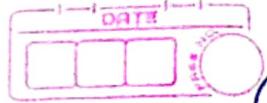
These methods except query operations as delegate objects which contain references to their implementations.

iii) AlQueryable Extensions:

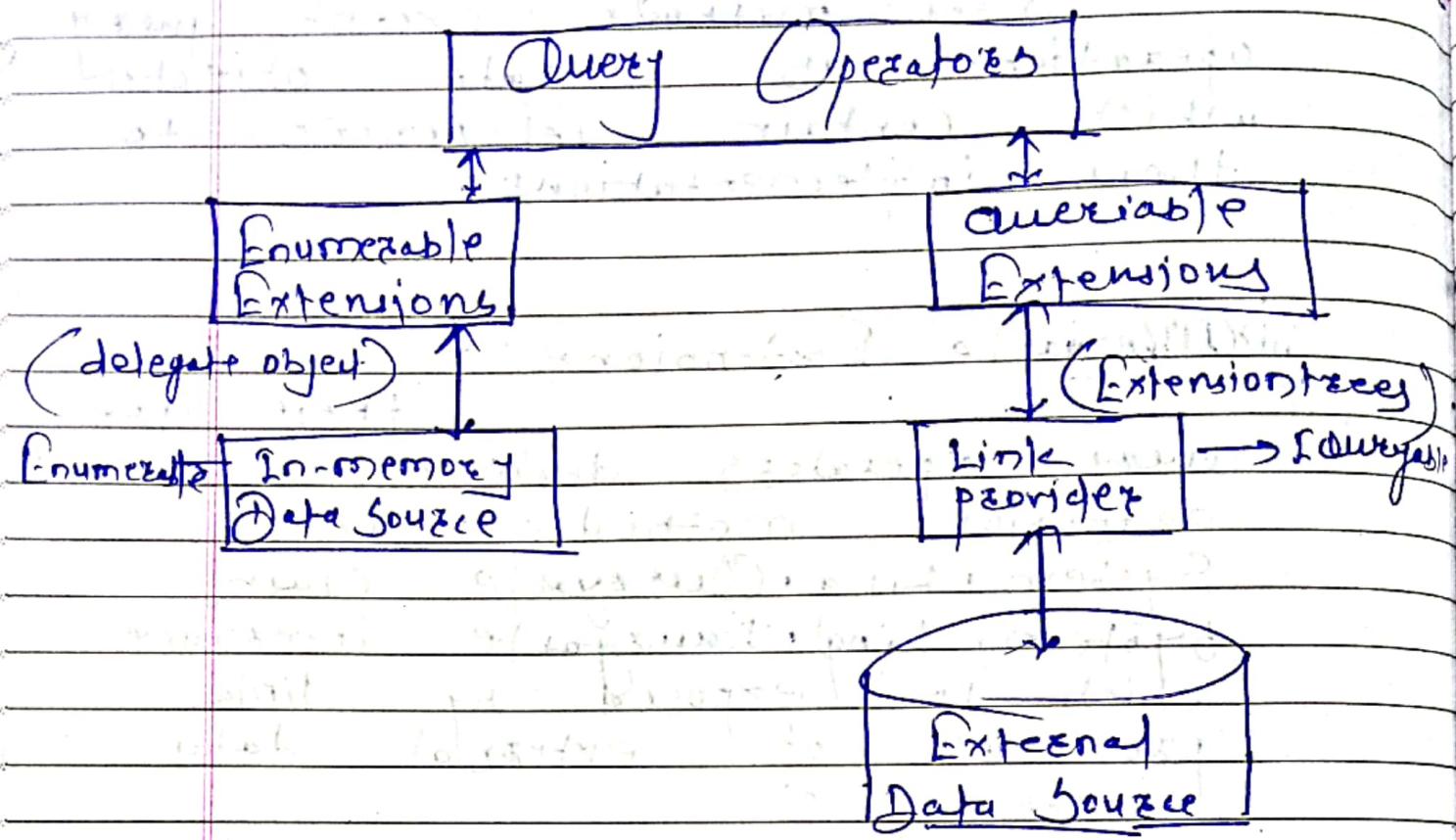
these are query operators defined in extension methods in System.Linq.Enumerable class, System.Linq.IQueryables interface which is exposed by link providers of external data source.

these methods except query operators as expression trees which contain instructions of their implementations.

(Private)

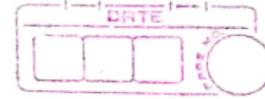


Navita



Notes

- * `Storage` is list, basically
- * `Where` is extension method.
anything that has this as first parameter is extensible method.



Reflexion in .Net

It is a mechanism which enables a program to examine the [metadata] structure of its object at run time.

Under CLR metadata of type is loaded on demand from its assembly as an instance of `System.Type`. The reference to which can be obtained using following methods.

i) From an object `obj` of the type
`Type T = obj.GetType()`

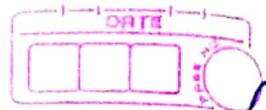
ii) From the literal name of a type `T` which is known at compile time.
`Type t = typeof(T)`

iii) From Fully qualified name

`N`(Namespace + Type)

`N(Namespace.Type, assembly)`
of a type which is discovered at runtime

Ravita



(Ngr)

Binding	static	Dynamic	Implementation
Class	Compile Time	Run-Time	Run-Time
Method	Compile Time	Compile Time	Run-Time



Attribute :-

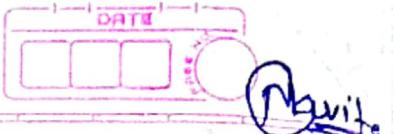
It is a user defined modifier which can be applied to a declaration of Assembly, class, field, property, method etc.

It is used for extending the metadata of its declaration target in programming language neutral/independent manner.

A custom attribute has following characteristics :-

- i) It is defined as a class which extends a System.Attribute.
- ii) The serialized bits of its instance is injected by compiler as custom data into the metadata of its declaration target.
- iii) Its presence can be examined at run-time using reflection upon its declaration target.

(Parity)

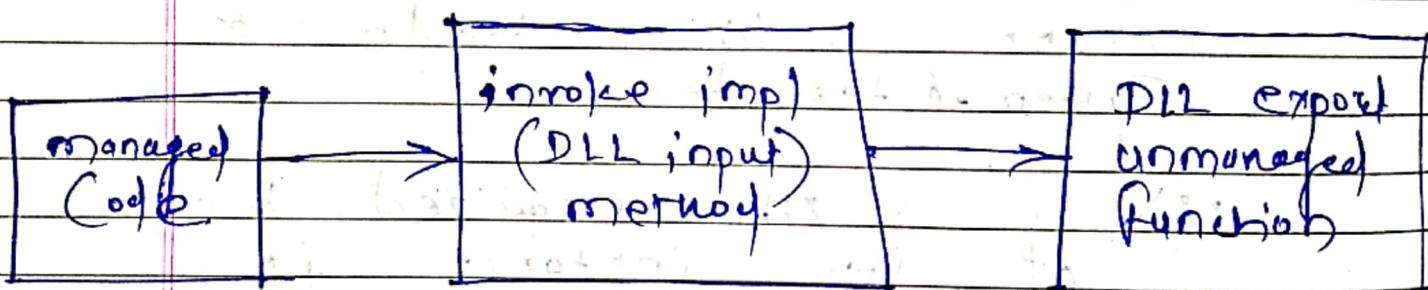


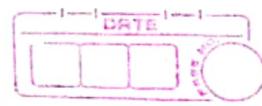
* Interop Services :-

It is functionality supported by .NET framework for invoking unmanaged (C, C++) code from managed (C#) code. It provides following facilities.

↳ Platform Invocation :- [P/Invoke]

platform invocation for calling unmanaged functions exported by native DLLs.





C datatype

char

long

long long

char*

const char*

struct T

const struct T

T (*function) (..)

Ans: C# datatype.

byte

int

long

byte[]

string

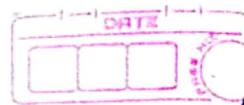
ref/out struct T

in struct T

delegate T Function(..)

Narita

2-12-20

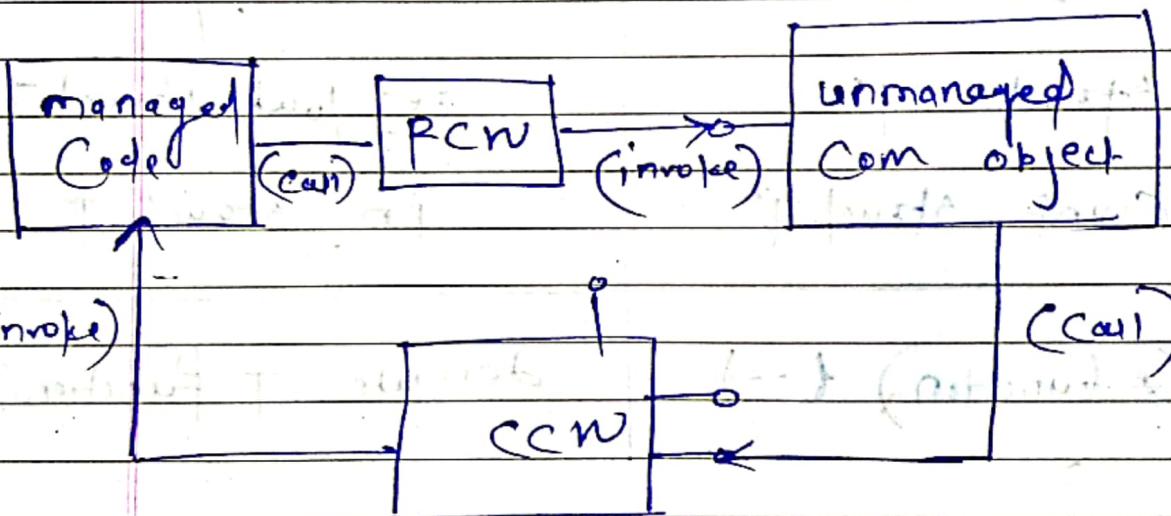


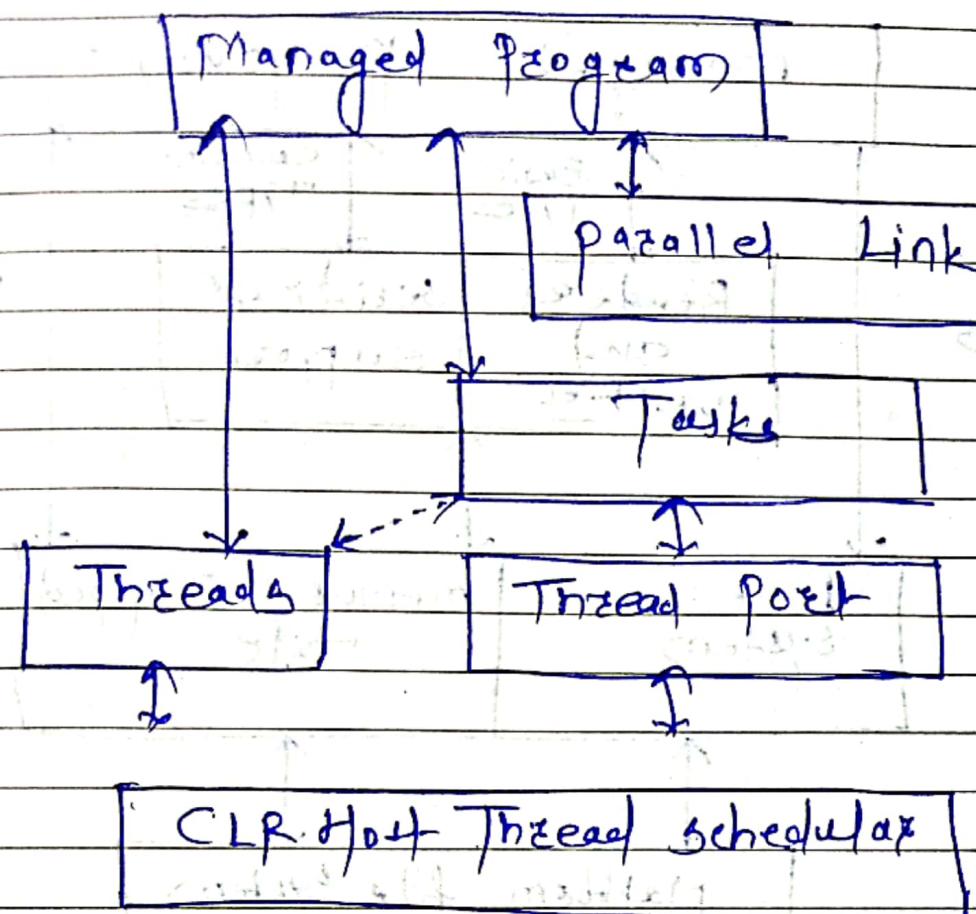
Navita

ii) Com-Interop :-

For consuming COM interfaces expose by unmanaged object using its Runtime Callable wrapper - and for passing a managed object to a COM interface through its COM Callable wrapper.

RCW → Runtime Callable Wrapper.





Note Thread and Task does same thing only difference is if you want to perform any operation and method has returning something use Task otherwise, go with Threads

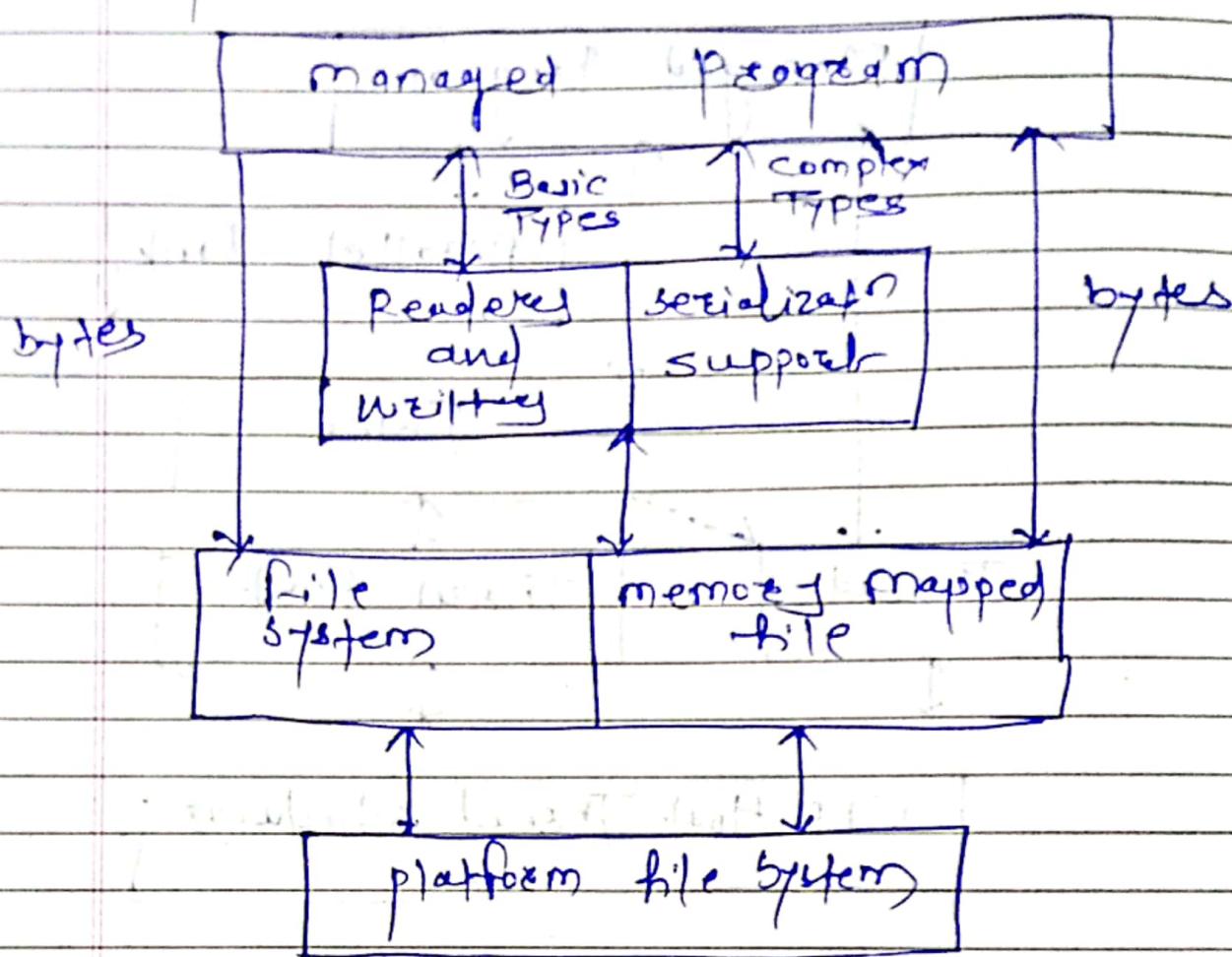
(Naveen)

3-12-90



Nov.

* Persistence in .Net



Persistence :-

Programm maintaining state of executions across its multiple it's is called Persistence

Object Civilization:

Converting an object into a series of bytes from which it can be reconstructed (deserialization) is called object civilization.

It is commonly used for

1. Persistence :-

Abwita

* Object serialization in .Net

.Net provides support for serialization which can serialize & deserialize any object that satisfies following Condition.

- 1) If type & the type it extends must be defined with `System.Serializable`.
- 2) It must not refer to a non serializable object from an instance field which is not defined with `System.NonSerializedAttribute`.

Event Driven Programming

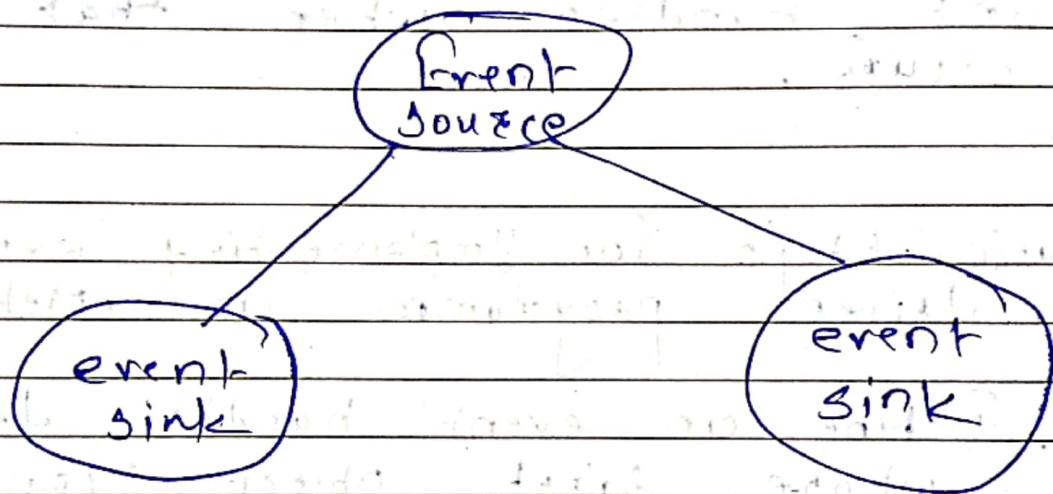
This is style of programming (push model) in which an object - called event source - releases notification called an event describing a certain change in its state to other objects which are interested in taking some action known as event handler when that change occurs.

* Basic steps for Implementing event driven programm in .Net.

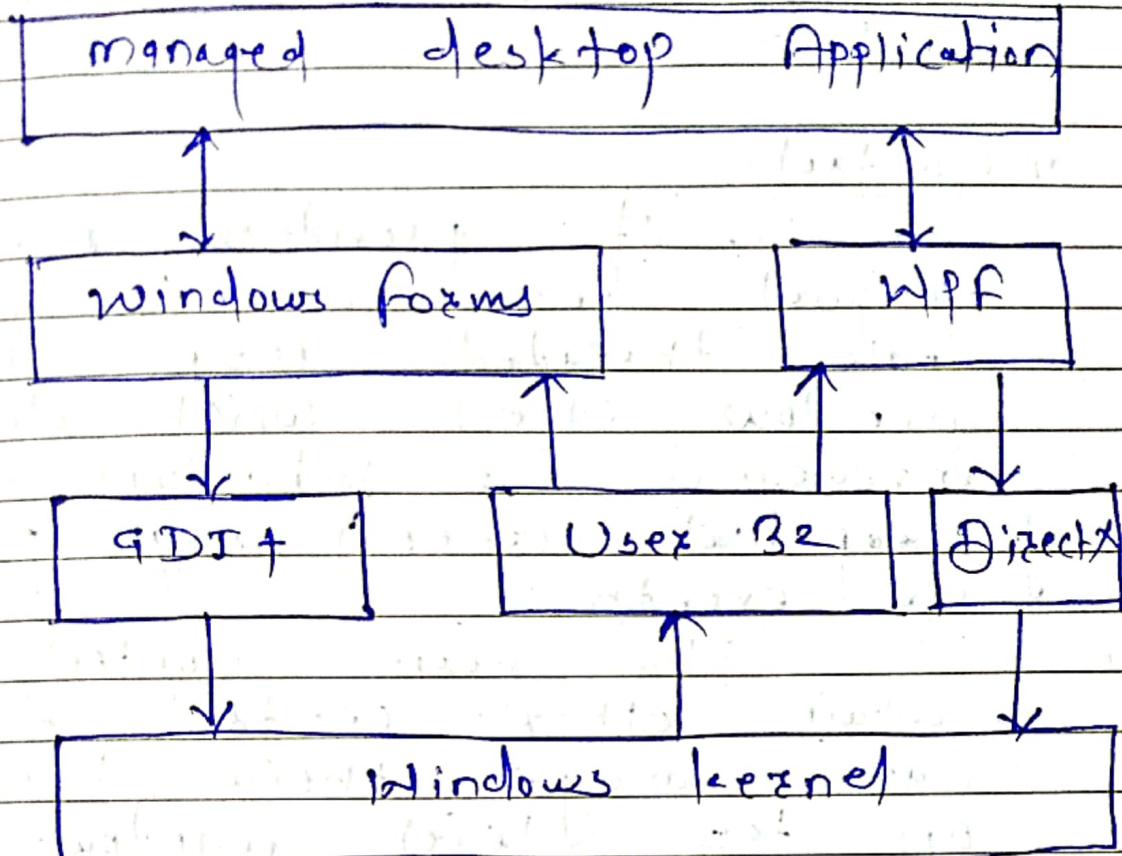
- 1) Define an event handler delegate whose first object type argument element identifies the sender of the event and second of System.EventArgs derived type arguments provides information relayed with information.

- 2) Define an event member of the above delegate type in the event source class and invoke this member to raise the event.

3) Define a method `comparable` of above delegate in event sink class. Then add this method to the event member of the event source so that it is in voice. When the event is raised



Desktop Programming in .Net.



* Windows Forms :-

If it is runtime support offered by .NET that enables runtime support managed desktop application to O/P basic graphical content using windows User32 & GDI+ API (Services)

[Graphical Device Interface - GDI]

Anita

The programming model of windows form is based on \leftrightarrow

1) Control

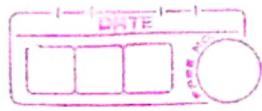
It provides managed support for working with single threaded msg chain window object (which describes appearance & behaviour of window of graphical screen) using properties & events.

The Form control can contain other control & can manage their layout on top level window

2) Application

It provides managed support for dispatching the messages receive by the owned thread of controls window object to message handling method (WndProc) of that control

A control handles a particular type of message (m) by raising its corresponding type of event by calling corresponding method (Onm) & passing it message specific data with event args



Graphics → It provides managed support for adding text to shapes & text on a graphical region using its drawing attribute such as brush, pen font etc...

When a control/window becomes invalid, its owner receives paint message & the control will invoke paint method by passing it graphics of invalidating region in event args.

Designer:

It provides managed support for visually activating a component type object & sending properties at programming time in a development environment.

A managed class can be used as component if it implements system component mode IComponent interface & supports parameters constructor.

5-12-20



Windows
Presentation
Foundation

+ Windows Presentation Foundation:-

It is a runtime support offered by .NET that enables a managed test of application to present rich graphical and multimedia content using windows user 32 and DirectX API.

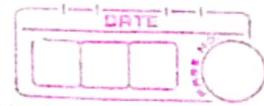
The programming model of WPF is based on

1) Dispatcher Object:-

It is an object whose functionality is directly accessible only to its owner (activating) thread which can handle invocations for other threads through messaging.

2) Dependency Object:-

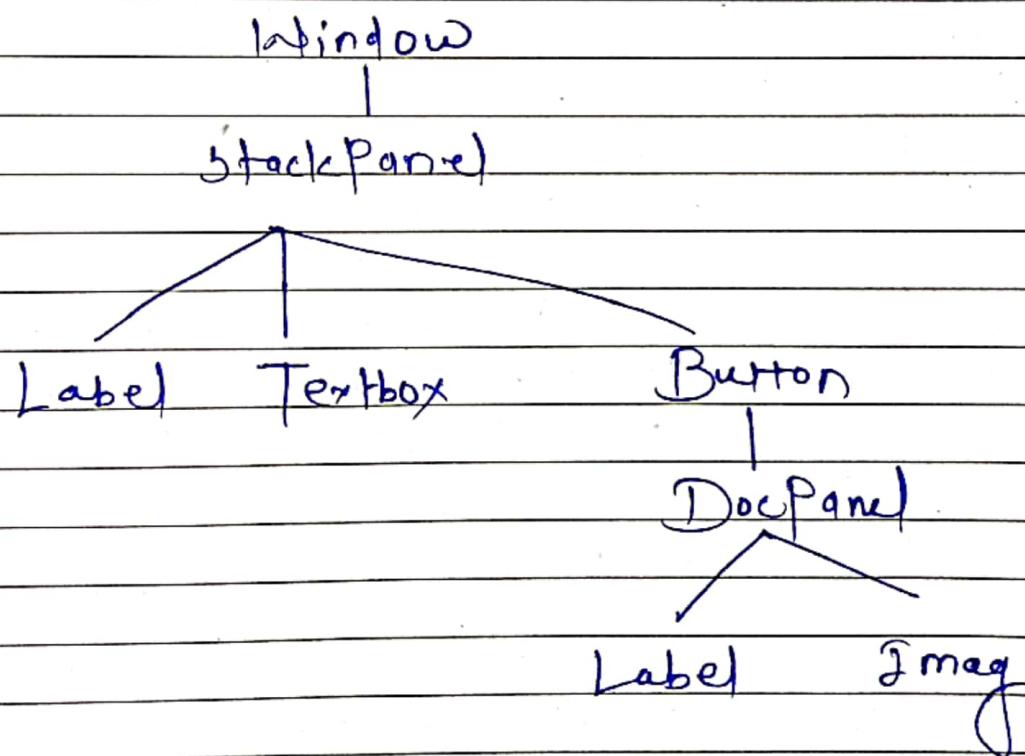
It is a dispatcher object which can expose sparsely stored properties with support for bidirectional data binding.



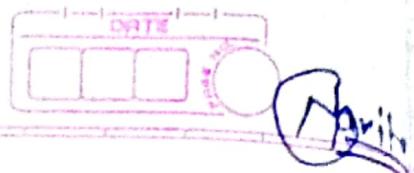
3) Visual :-

It is dependency object which can provide drawing instructions & for rendering it as node of tree of such objects.

Visual Tree



(Anita)



4) UI Element :-

It is a visual which can have layout and can receive inputs directed by the user to that layout.

Windows Forms



1A) O/P is rendered as raster graphics which is composed of pixel locations & colors.

1B) S/P is directly received by the control which is a heavy weight (window) object shown by the parent.

2) Offers event driven model which requires simple programming for handling user interaction.

3) It provides extensive support for visual design time support for layout & data binding of controls.

4) It can only present 2D graphics.

5) It does not provide any support for separating layout of UI from its interaction logic.

WPF

O/P is rendered as vector graphics which is composed of drawing metadata & instructions.

S/P is routed by the parent to the control which is light weight (windowless) object drawn on the parent.

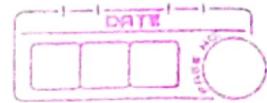
It prefers property based model for handling user interaction which requires complex programming.

It provides limited visual design time support for layout & data binding of controls.

It can present 2D/3D graphics with built in support for animation.

It provides markup language called XAML for describing layout of UI separately from its interaction logic.

Ans



Implementing a bindable object
for property based programming
in WPF

Step 1 :-

Define a class that
implements `System.ComponentModel`

`System.ComponentModel.INotifyPropertyChanged`
interface containing
`PropertyChanged` event.

Step 2 :-

Support bindable data
in above class by defining
a property to set block
guides `PropertyChanged`
event identifying itself in
the event args.

Step 3 :-

Support bindable command
in the above class by
defining a property which
returns an

implementation of,

System.Windows.Input.ICommand
interface containing

a) CanExecuteChanged event-

which is raised whenever there is change in parent's property that affects the availability of the command.

b) CanExecute method

which indicates whether parent is in the state to support the command.

c) Execute method

which manipulates parent's properties to service the command.

1/12/90



It is runtime supported
for using
off service oriented architecture to
implement managed
on top of windows communication
mechanism built within
windows system
The programming
model on WCF based on

1) Contact :-

it describes
the operations supported by
service & the data
exchanged by them

2) Binding :-

it describes how
messages are formatted
and transported through a
channel during invocations
of service operations

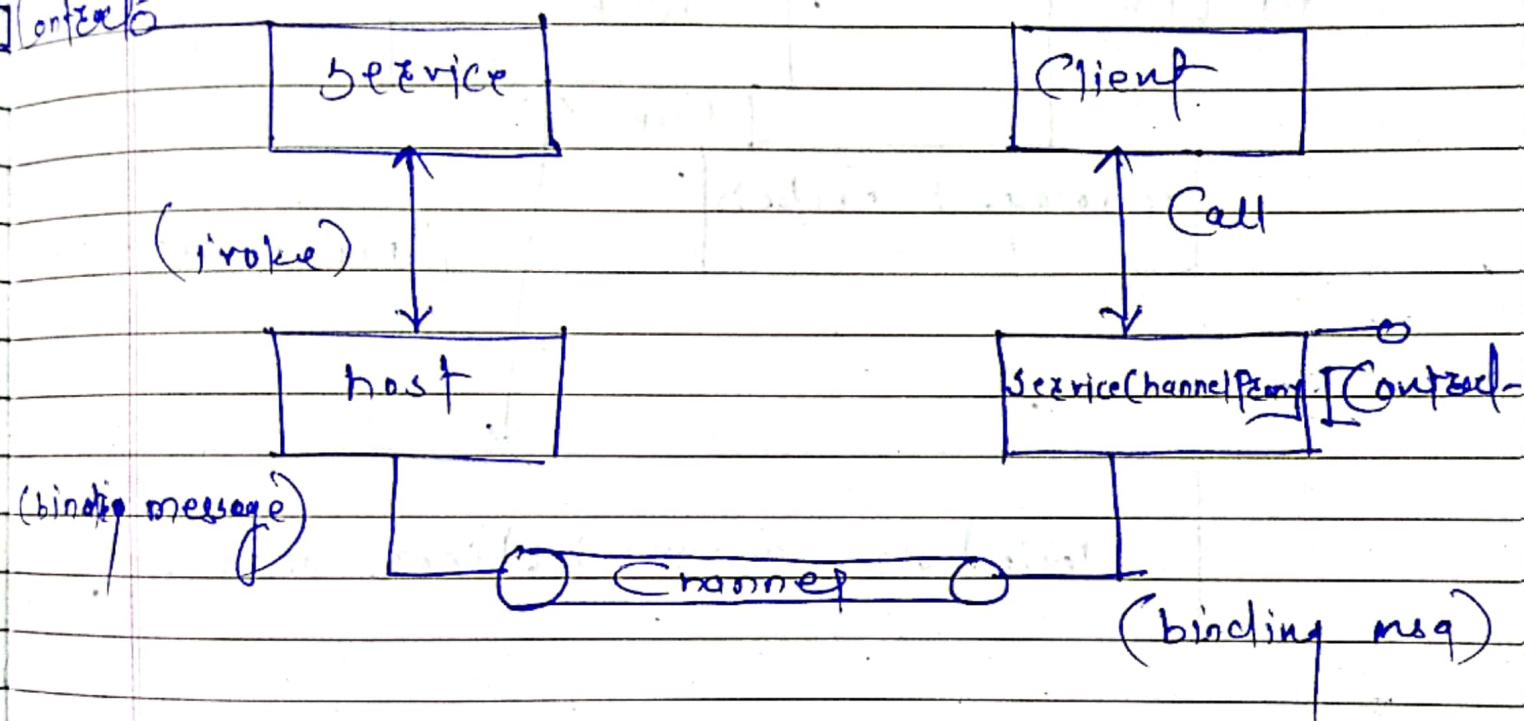
3) EndPoint :-

it describes the
well known address on
which a contract of a
service is published by
its host using a particular
binding

4) Behaviour:-

It describes common facilities required by a service for its operation at runtime from its host.

Contract

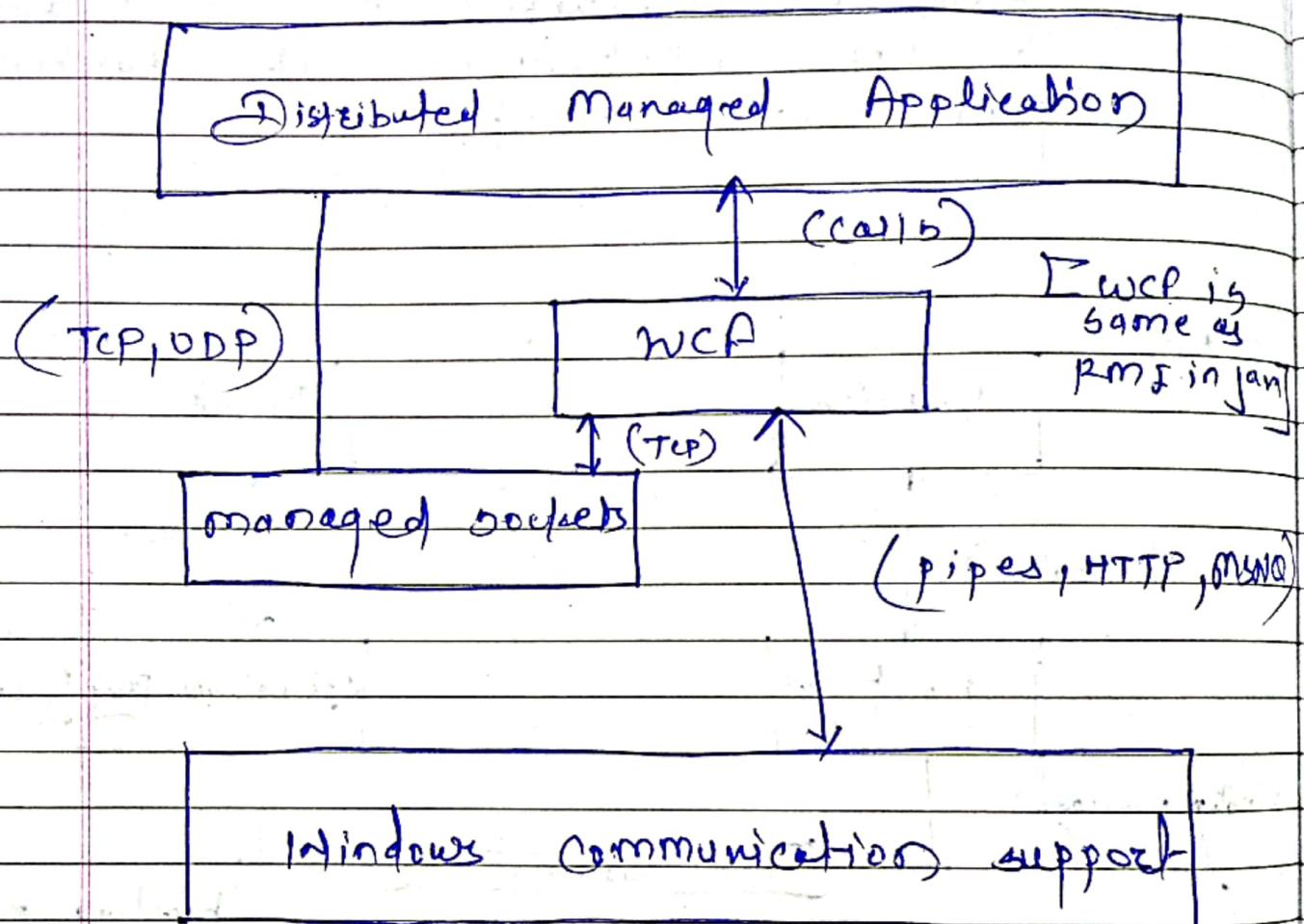


8/12/20



Nov 1

Distributed Programming in .Net :-



* Service Oriented Architecture :-

It is methodology used for designing a distributed application in which its implementation is divided into loosely coupled sets of operations called services in such way that each service :-

- ① Has an explicit boundary beyond which its operations can only be consume through exchange of message
- ② Shares only description of its operations but not their implementation
- ③ is autonomous i.e. it can be displayed independently of other services
- ④ Provides machine readable policy specifying its capabilities & requirements

(N)guru



(Agus)

A WCF :-

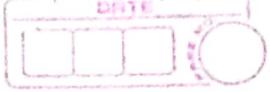
It is runtime support offered by -

Using service oriented architecture to implement managed distributed application on top of communication mechanism within windows system

A The programming model of WCF is based on :-

① Contract :-

It describes the operations supported by the data exchanged by them.



② Binding

It describes how messages are formatted and transported through a channel during invocations of service operations.

③ EndPoint :-

It describes the well known address on which a contract of a service is published by its host using a particular Binding.

④ Behaviour :-

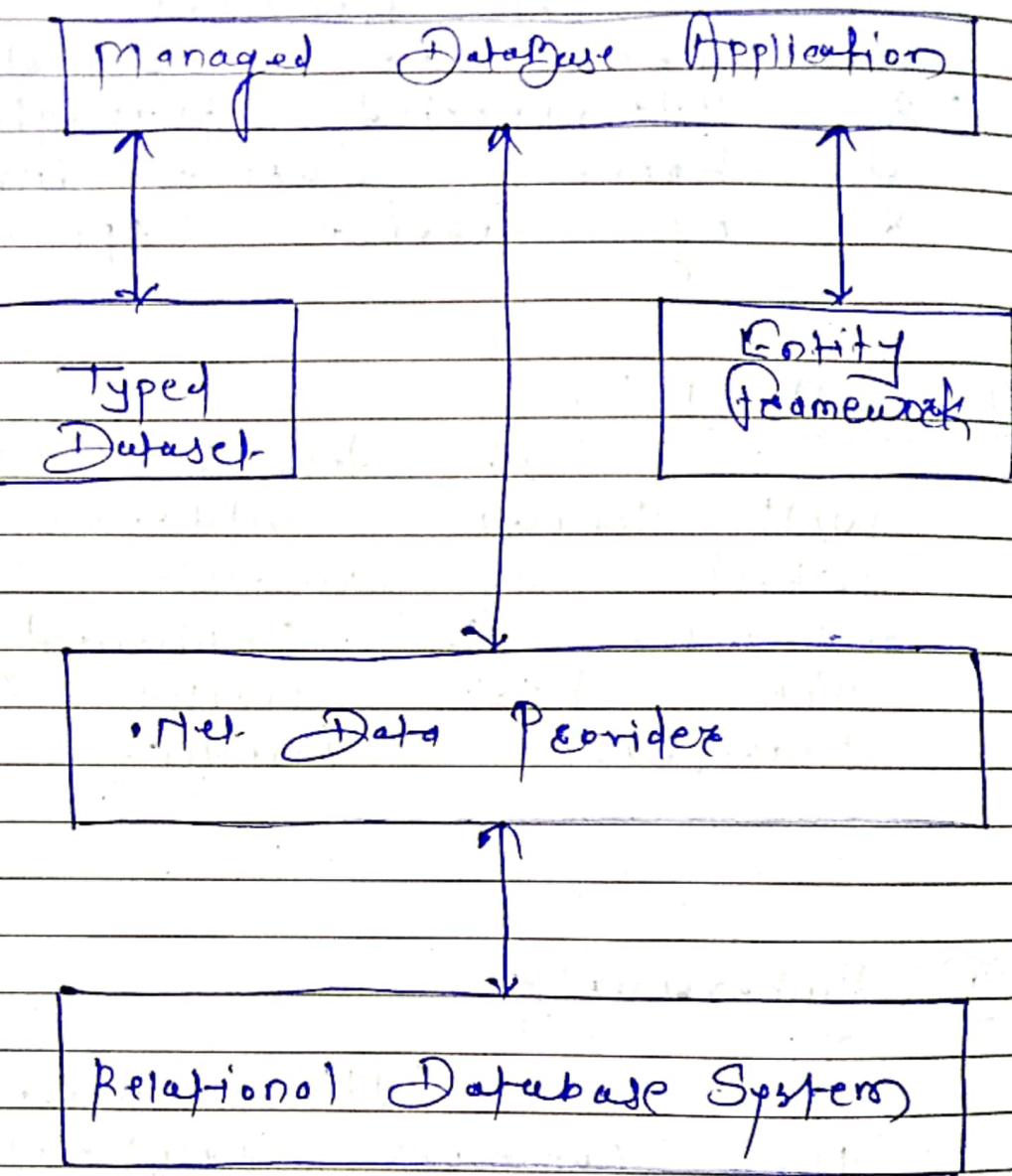
It describes the common facilities recognized by the service at runtime from its host.

Nawira

8-12-2020



A Database Programming in .Net.



ADO.NET :-

It is runtime support offered by .NET which enables a managed application to consume relational data provided by a database system before .NET became micro.

A .NET Data Provider is a database specific implementation for managed types specified by ADO.NET for accessing that database. It includes support for following objects.

1) Connection :-

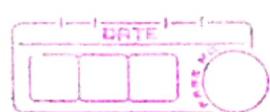
It opens a communication session with database system using set of name values called the connection string.

2) Command

It executes an SQL statement or stored procedure using connection object.

③ Datareader:

It fetches rows resulting from execution of a query statement using the command object.



* .NET data provider serves as basis for typed data services.

It provides access to relational data through an in-memory collection of data-table objects which can be filled from a database system

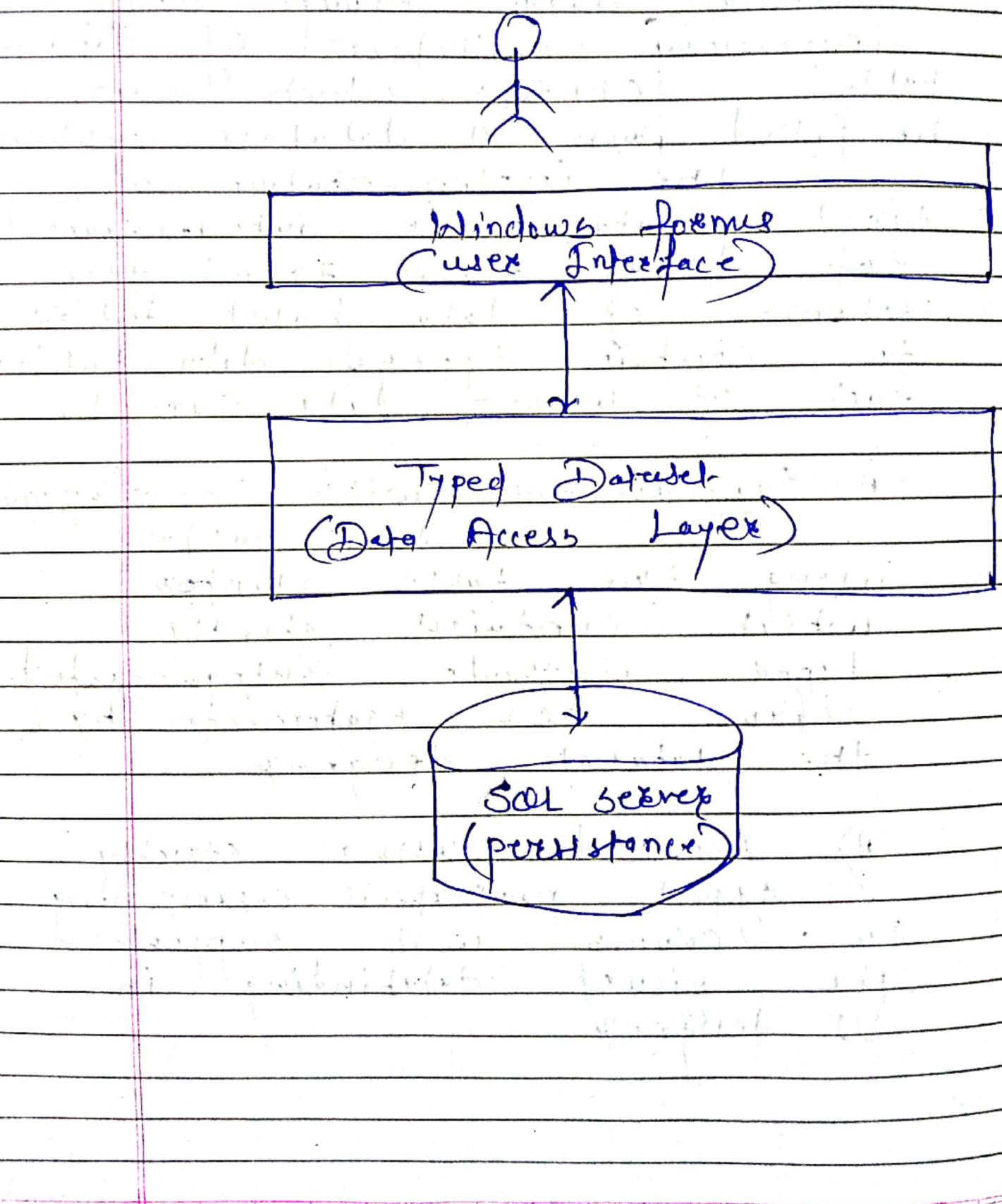
The implementation of typed dataset is autogenerated at design time from the schema of data based tables to contain typesafe data tables each of whose data can be

A) Synchronized :- with corresponding data table

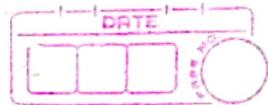
using the table adaptor which contains strictly typed methods autogenerated from SQL statements by the dataset designer

B) Accessed using strictly typed properties corresponding to columns with support for visual databinding in UI designer

A typed dataset is used to implement data access layer for database client application.



2-12-20



Entity Framework:-

Entity Framework provides access to relational data through link queryable sets of data context unaware objects link which can be loaded on demand from the database system.

Entity Framework provides a standard convention for mapping database table to poco (plain old CLR object) classes which can be customized using

A) Fluent API :-

The conventional mapping is overidden by passing the metadata of class to the poco class & its members through a chain of calls to methods of model builder objects.

B) Data Annotation :-

The conventional mapping is overidden by applying schema attributes.

Attribute

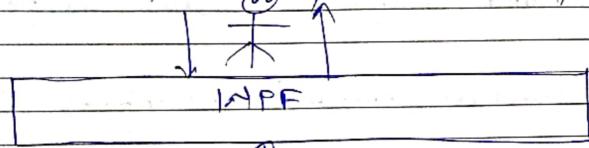
to the poco class & its
members.

Poco	Database	Rule.
1 Entity Class	Table	Table name is plural of class name.
2 Entity Identity	Table column.	Column name is same as property name.
3 Id EntityId Property	Primary key column	Database generated for integer type.
4 ParentEntityId Property	Foreign key column	References primary key column of table mapped to primary entity mapped to parent entity

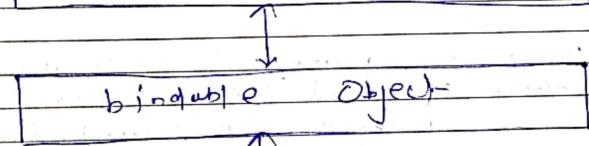
* Use Cases for Entity Framework

Entity framework is used
to implement data access
layer for

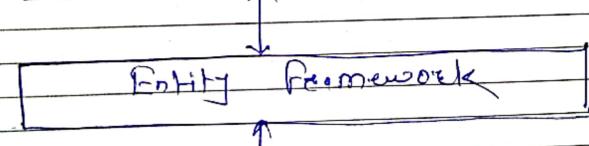
1) Database design Client application
using mvc pattern



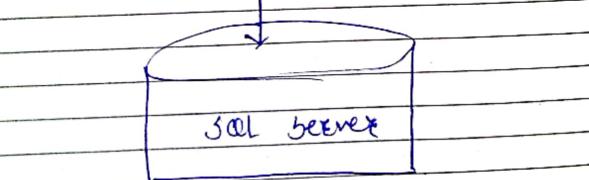
view provides user interface to support input/output of application data.



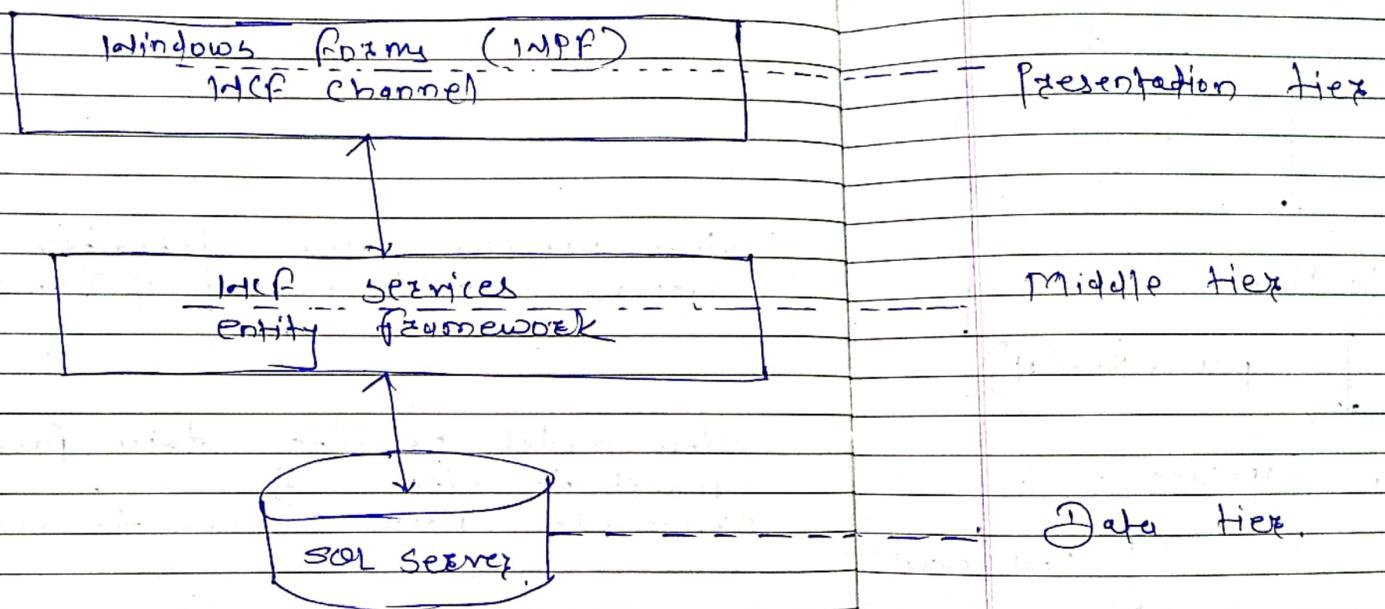
view-model provides data from model to view & Commands from view to model.



model provides logical interface for reading/writing application data.

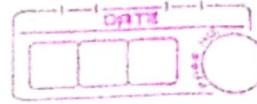


2) Middle tier for enterprise Application



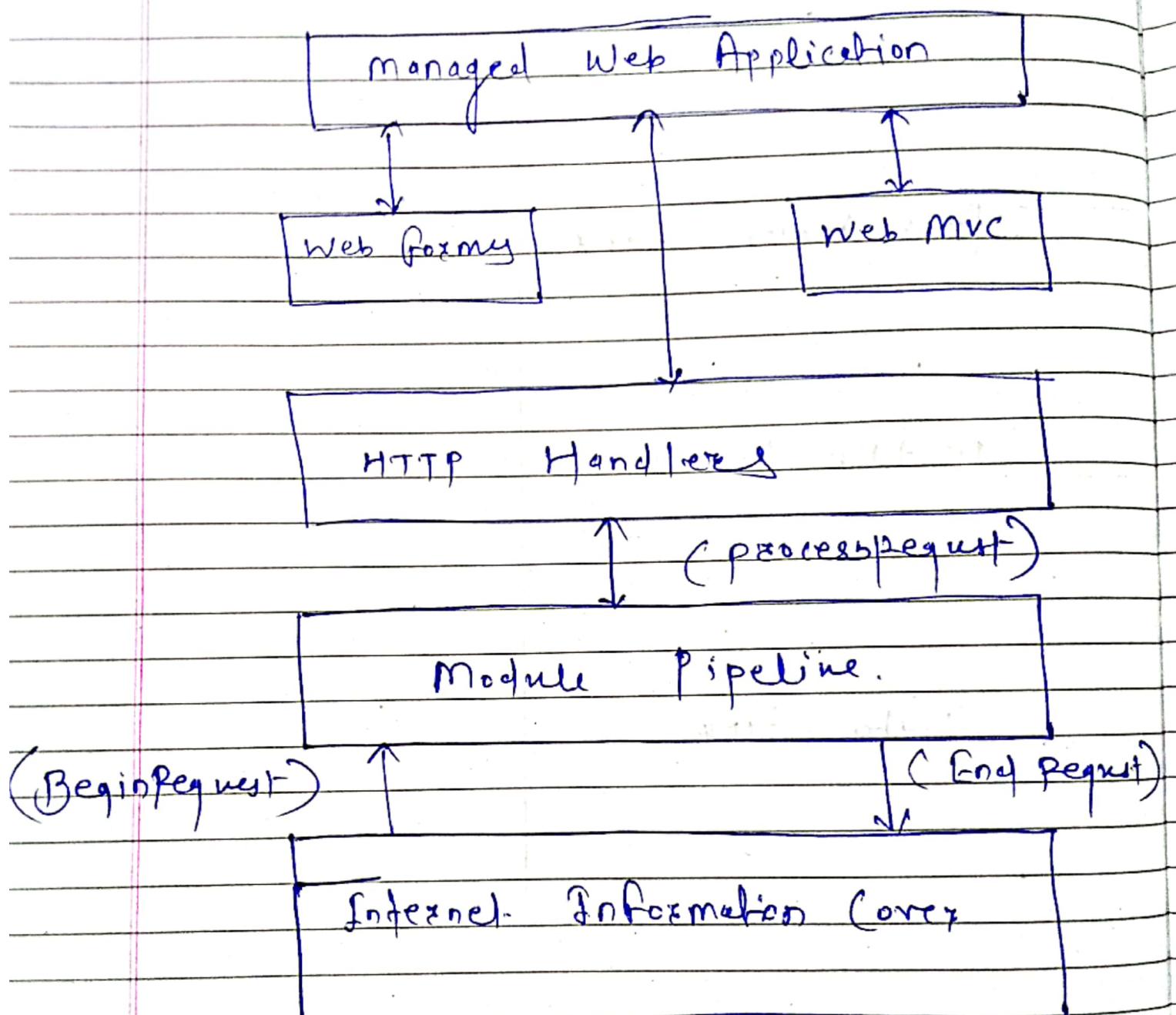
NTR Architecture in .NET

10-12-20



@Nani

Web Application Programming in .Net



ASP.NET

It is runtime support offered by .NET Framework for hosting a managed web application on a CLR aware web server (such as IIS) and for routing an HTTP request received by server through a pipeline of modules to the appropriate handler implemented by that application.

An ASP.NET Handler is managed object whose class implements `System.Web.IHttpHandler` interface to process any request with particular path and work for its application.

ASP.NET services each request received by web server using following steps

Navita



Chaitanya

Step 1 →

Initialise the Http Context to include the current request & response object & raise BeginRequest event of each module in the pipeline.

Step 2 →

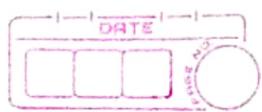
Get the handler object from the handler factory mapped (by configuration) to requested map of verb path.

Step 3 →

Invoke ProcessRequest method of the above handler object passing it the current Http context & then pass this handler object to ReleaseV method of its factory handler.

Step 4 →

Raise & request event of each module in the pipeline & send the content of current response object to the web server.



path	Handler Factory	Handler Types
.ashx	Simple	identified in required file
.aspx	page	generated from generated file
others	<default>	specified in configuration file.

* ASP.NET handlers also stores as basic file:-

① Web Forms

It is a framework -
for composing a web page containing a form based user interface by combining client side & server side mark up elements

The path (*.aspx) is of web form page is mapped to page built in page handler factory which Reusability

generates the handles from the content of the requested page to process a request based on

1) Web Control :-

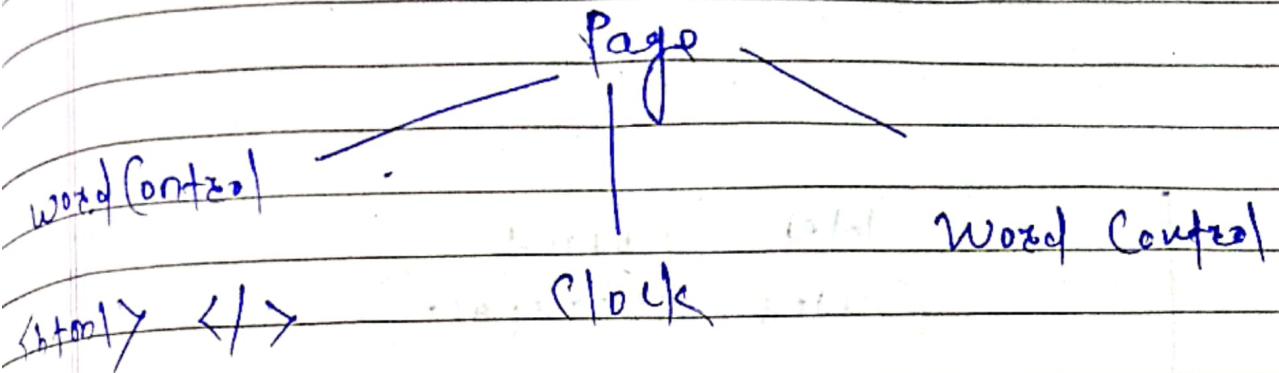
It is a server side component which appears in a web form page as markup element with run-at attribute set to server. It supports Render method which outputs its corresponding client-side markup using the specified writer.

2) Page :-

It is an instance of System.Web.UI.Page class which contains a tree of web controls of a webform page.

When a webform page is requested the corresponding page of initialise its load event

raise & its web control
are rendered.

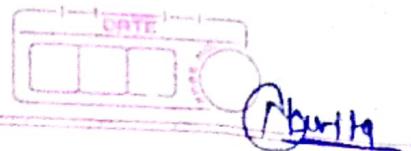


3) Post Map:-

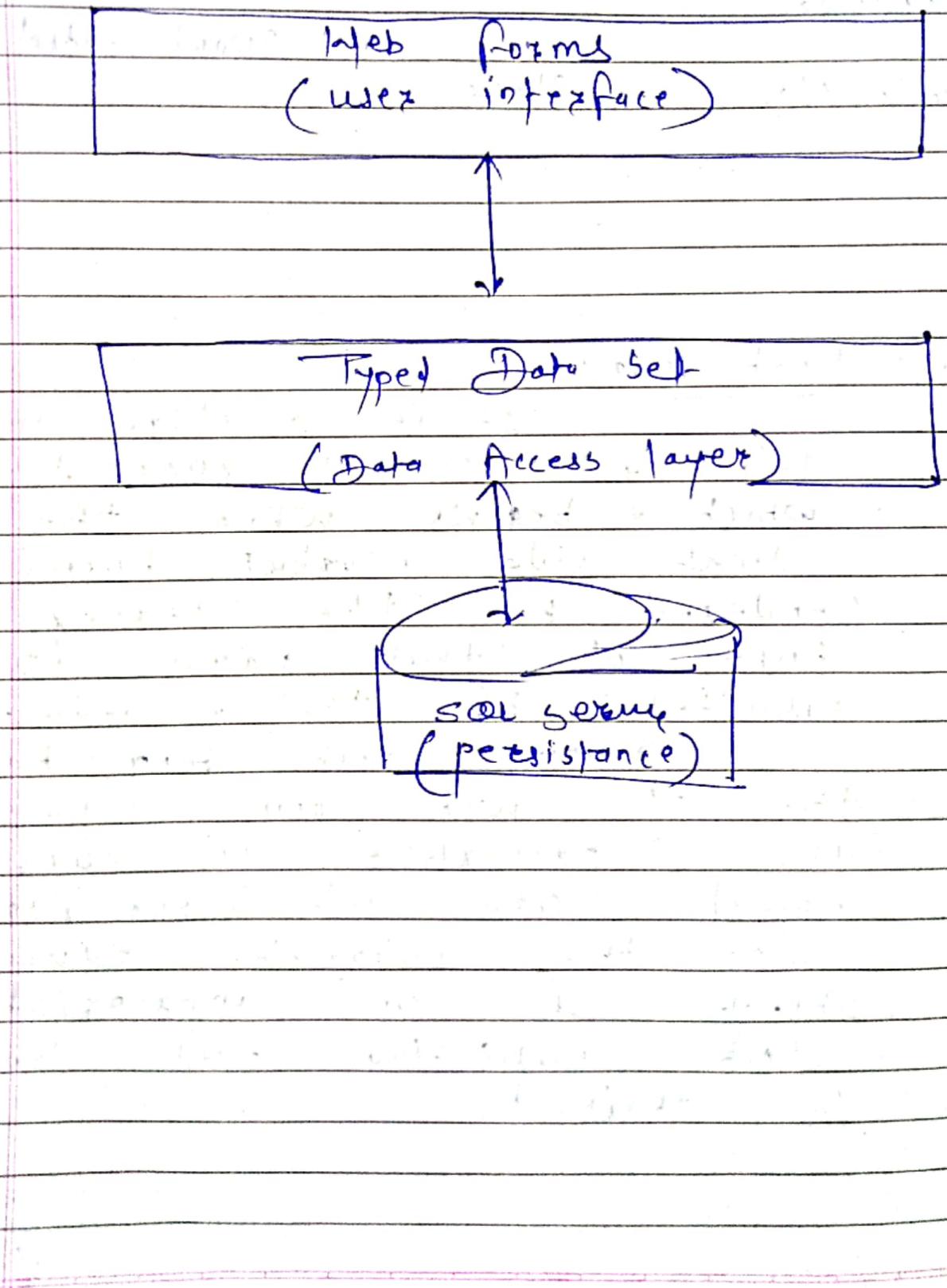
It is a stage in life cycle of web form page which begins when the client side Markup Form rendered by the server side form of that page is submitted by the client.

During post back the properties of client object Page set of Web Form object. i.e. the current request or appropriate web control is raised.

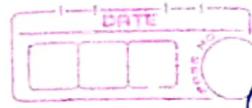
Partha



Using Web Form in Data Driven Application:-



11-12-20



Navin

ASP .NET WEB MVC :-

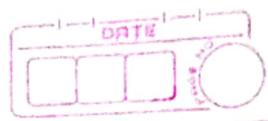
If it is a framework for implementing a large data driven web application using MVC.

MVC is design pattern for implementing MVC for dividing of an application into Separately programmable parts with following responsibilities

1) Model :-

It implements logical interface for reading & writing application data in terms of persistence unaware data objects.

ASP .NET MVC provides data annotation attributes for specifying the presentation & the validation of data objects imposed by the model



2) View :-

It implements the web interface for supporting input & output of application data.

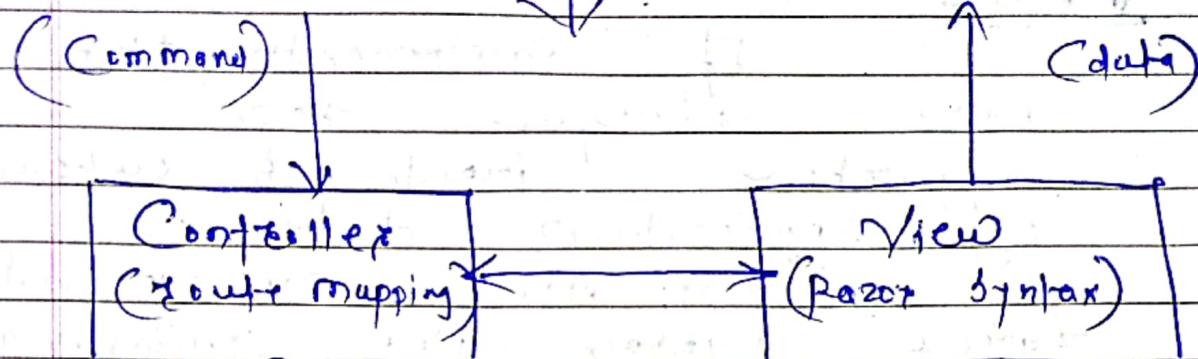
ASP .NET MVC provides razor syntax for combining server side (C++) code with client side (HTML) markup to compose a web page for the view.

3) Controllers

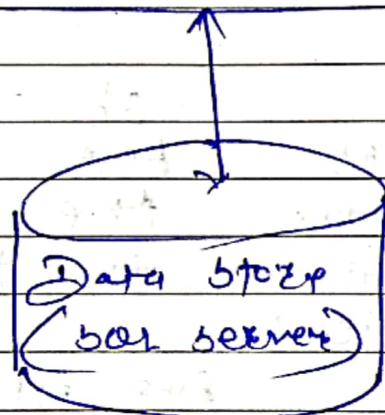
It implements handlers for user commands to pass the corresponding instructions to the model & the view.

ASP .NET MVC provides route-mapping for invoking action method based on path which also identifies the controller.

(Priority)



Model
(Entity Framework with data annotation)





REST

ASP .NET MVC also enables a web application to publish RESTful API client methods called WEB API which a client side code can call to Create, read, update or delete a data object managed by that application.

REST is an architectural style commonly apply when the negotiable Representation state of some resource (data object) is to be transferred between HTTP Endpoints using an operation which is

Parmita

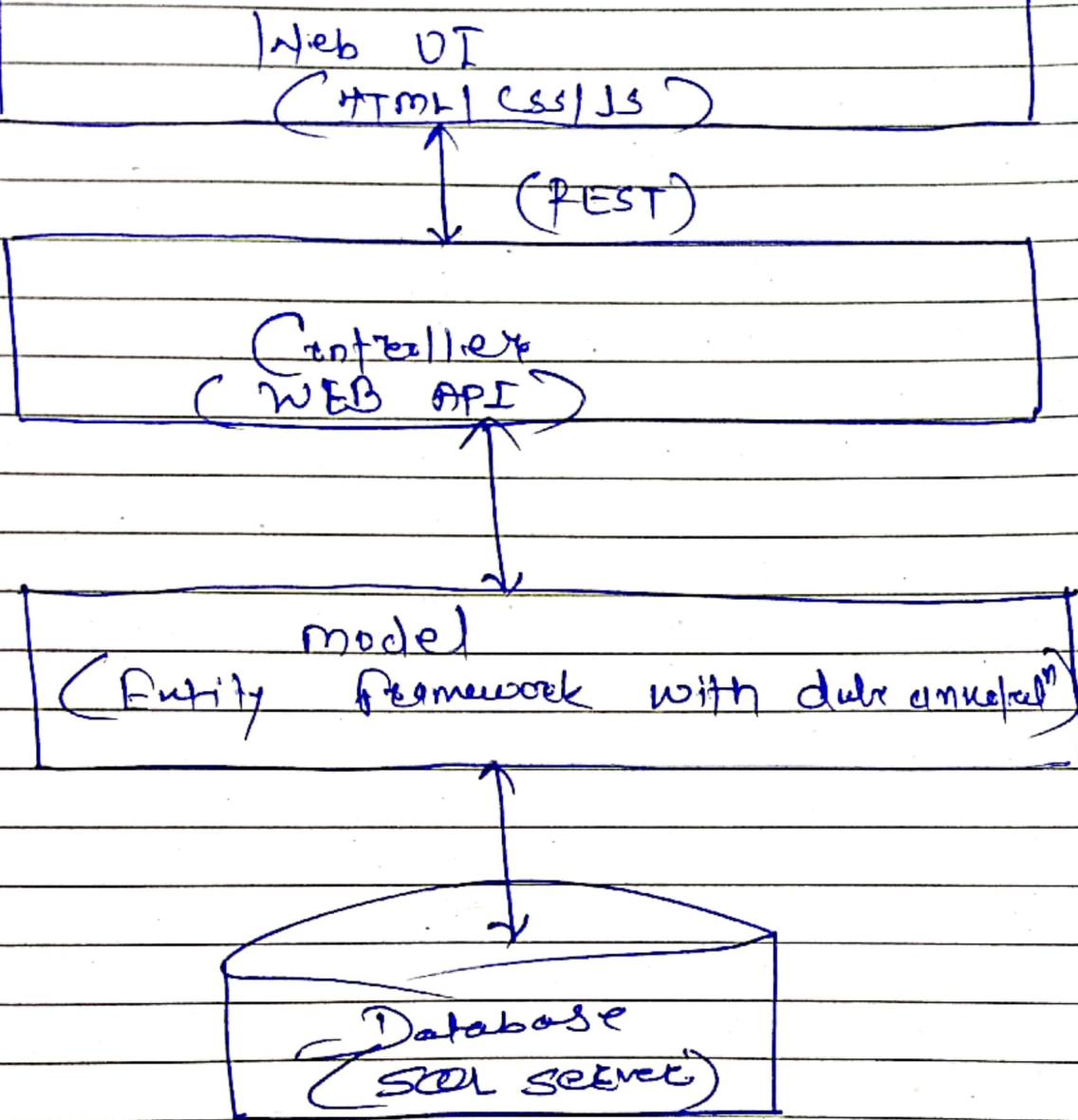


Notes

3) Performs a standard action
(Create, Read, Update, Delete)
Corresponding to the requested
verb (post, get, put, delete)
on resource identified in
the requested path with
its state in the request
body

2) Returns the state of
resulting resource in the
response body while
indicating success or
failure using standard
response status code

3) Uses standard Web Media
types such as JSON and
XML to exchange a
resource whose state can
be expressed in form
of structured data



Nerity