Un objeto es una instancia de una clase, pero la forma previa de saber esto seria, es cualquier cosa del mundo real que pueda ser programada. Un objeto posee identidad (serían las características que definen de una manera especial a un objeto, la idea general seria que cada objeto por simple hecho de existir será diferente a los demás), comportamiento (el comportamiento es todo lo que puede hacer el objeto) y estado (seria el valor de todos los atributos en un determinado momento)

Un objeto puede ser

Físico: auto

Abstracto: cuenta bancaria

Algo del dominio (es el escenario que deseamos resolver) del problema: (seria los problemas que buscamos resolver) como un préstamo

Algo del dominio de la solución: un botón, lista despegable

Un mensaje es el medio por el cual los objetos se comunican entre sí, en todo mensaje existe un emisor y un receptor y su medio,

Cubo.get (obtener un valor)

Cubo.set (establecer valor)

Cubo.girar (hacer algo)

Una colaboración se produce cuando dos o más objetos interactúan entre sí para realizar una tarea

Un poo es una red colaborativa de objeto o una red de objeto que colabora entre si

La responsabilidad, incluye tanto conocimiento y acción, incluye un aspecto estático y dinámico, es importa a la hora de definir un poo, para definir las clases, y cuando definimos los objetos deben tener una responsabilidad almenas, la responsabilidad define la capacidad que tendrá el objeto de poder hacer algo

Tarjeta CRC (SON LAS TARTEJETAS QUE SON SIRVE PARA SABER QUE RESPONSABILIDAD TENDRAN NUESTRAS CLASES Y CON QUE OTRAS CLASESS VAN A INTERACTUAR)

CLASE es una matriz para crear objetos del mismo tipo (las clases nos servirán para manejar la complejidad de los objetos de una manera más simple)

Compuestas por atributos

Métodos

Constructor (inicializa los valores que tendrá de manera predeterminada que tendrá ese objeto)

Eventos

Relaciones

Una relación en uml refleja un vínculo existente entre elementos:

- Asociación (es la conexión de un objeto con otro)
- Agregación / composición (son tipos de asociaciones, se consideren como un tipo de, normalmente son muy similares a la hora de codificar, es decir que, cuando se habla de composición existen una relación de dependencia, es decir que cuando una clase que está asociada con otro, y se elimina la principal, la secundaria no funciona, en la agregación se pierde la exclusividad, es decir que un objeto puede depender de más de una)
- Generalización / herencia
- Dependencia
- Realización (es un subtipo de la dependencia)

Pilares poo

- Abstracción (es lo que nos permite definir lo importante dentro del objeto y descartar los secundarios)
- Herencia (sirve para reutilizar código, es una relación entre de clases,)
- Encapsulamiento (es el mecanismo que nos permite ocultar los detalles de la implementación de un objeto, siempre se recomienda que los atributos de los objetos sean privados, lo que se logra con el encapsulamiento, es proteger los datos, separación de la función,), seria los límites que le damos a los objetos para ver que disponemos al exterior, y además para que sea más fácil de mantener el código
- Polimorfismo

Prehistoria de .net

c/Windows api

- desarrollo tradicional

ventajas y desventajas: no está muy abstraído de lo que es un lenguaje de máquina, y ventajas es que se pueden manejar punteros y direcciones de memoria lo que hace mi flexible que un lenguaje de alto nivel, no es obsoleto, ya que sirve para programar otro tipo de aplicaciones, como sistema indebidos o app que requiera un manejo detallado de los recursos, dada a la complejidad este leguaje es propenso a errores por cantidad extensa de código,

c++/mfc

- orientado a objetos

es que posee mfc (microsoff fundetion clases), funciona como embolato a la Windows api es lo que nos permite a un leguaje un poquito más de alto nivel y aparte es un lenguaje orientad u objetos, pero sigue poseyendo los mismos errores que c porque sigue siendo un lenguaje de bajo nivel

VR6

- RAD mínimo esfuerzo

Es y un paso más arriba al nivel de abstracción, porque sirve para crear aplicaciones de en Windows de una manera más fácil, es un lenguaje que está enfocado en eventos porque sirve para facilitar la programación, no está orientado a objetos, está más orientado a eventos, Rad (desarrollo rápido de aplicaciones) permite prototipo una base de lo que se busca programar ni caso si hay que mostrárselo al cliente

Componente object model(COM): es una de la tecnología prehistóricas de .net, permitía crear objetos que sean reutilizables en distintas aplicaciones, necesitaban interactuar entre sí, com servia com podía ser el código común, conocido como servidor com(es aquel que pueda brindar cualquier servicio), para eso se necesitaba implentar ceiertar reglas se podía utilizar un programa com para que se puedan aderir a el, pero como tenia una gran complejidad al programar, poseía acceso a la blireria de atl, que estaba vinculada a mfc.

La diferencia entre plataforma y framework es que, plataforma incluye lo que es el hardwar y software (sistema oiperativo , entorno de dsarrollo, etc), mientras que los framwork

Framwork es un conjunto de librerías poara un determinado trabajo, además de eso incluye lo que son las herramientas y tutoriales que posee, pero lo principal para que sea un framwork es por la forma que estamos trabajando, el programa debe tener una parte dinámica, es decir un entorno de ejecución para utilizar estas bibliotecas

Objetivosde .net framework

Proveer un ambiente (consstente, minimice problemas, segura, performance, integrable)

Desarrollo web en .net ascp net

Es consistente porque no importa para que medio este trabajando las bases de la programación son la misma, iguala que tener en cuenta las particularidades de lops diversos medios

Seguridad, tratar evitar el uso de punteros, por eso ahora para compilación de un programa debe ser ejecutado

Componente de .net

Clr (se acemeja a la maquina virtual , funciona para cargar las aplicaciones cargadas en .net y poder ejecutarlas y podes administrar esos procesos que están cargados y además entre los compilados de .net intentar manejar el asociaminetos de esos aseemblys, en resumen es el resultado que obtenemos al ejecutar nuetro codigo) y bcl (es una coleccione de tipos resusables, para la utilización en una variedad de aplicación puede servir para una base de datos, o en la parte de entrada y salida, el manejo de cadenas,)

Cts (sirve para que por debajo del programa, esta funciona para que en todos los lenguajes las variables se vean de la misma manera es decir que un entero va a ser lo mismo e ntodos los lenguajes. Ayuda a que diferentes lenguajes puedan compartir información de una manera mas fácil y aunque sea un tipo de lenguaje comunes, lo que hace es definir todos los tipos posibles que podrain implemtar en framwork y eso no quiere decir que todos lenguajes implementados o que trabajen sobre el frame deban implementar todos los datos comunes, es decir que pueden ahcer partes, y eso se ve en cls(especificación del lenguaje común, es un conjunto de reglasque sdicen que es conjunto minito del cts que deben implementados por todos los lenguajes))

Cls (son las reglas que debe implemntar el lenguaje, es decir un conjunto minimo,el uso de tipos que no sean soporte de la cls hace que no sea conforme a la misa)

Cql(tiene una sintaxis simila sql)

*framwork 4

Multilenguaje. Porque?

Preferencia, inclusión y fortalezas y debilidades

que es un ensamblado? es un bloque de construcción básico del .net frame, el esamblado puede ser un ejecutable (.exe) o una extencion dll que es una biblioteca en clase dinámica o biblioteca de base, se encuentra ensamblada en un lenguaje il, o cil lenguaje intermedio común, el resultado de la ejecución es este lenguaje intermedio y en trascurso de la misma esta se va convirtiendo en lenguje de maquina

c#:

basado en c++ y vb6

propiedades y arreglo de parámetros

sobrecarga enumeraciones y callbaks (via delegates)

la programación con atributos (es la anotación que se puede poner encima de un método por ejemplo en casos que se vuelva obsoleto)

normalmente el manejo de manero se hace de manera automática esta en manejo de la clr, además tiene caracteicas de lengaujes funcionales, es una lenguaje especial como f#

ginaeric(es una colección generarica de lista) es mas flexible en c# encomparacion de java

se puede usar un método anónimo es decir que no tiene nombre es la base de las funciones lamda

la utilización de parcial se puede declarar una clase como parcial, es decir una aprte de la clase va a estar en un archivo y otro en otro, pero al compilar estaría funcionando a la normalidad

esto sirve en casos para que se puedan esconder los miembros auto generados en el entorno de desarrollo, esto lo hace en una clase parcial, para una mejorar el ordenamiento

la incializacio nde objetos inline, tipos anonmos, métodos de extencion y lambda es para quefuncion linq,

var deja al compilador para que interprete que es lo que se intentar buscar

de que se trata visual studio? Es una ide (es un enterono de desarrollo integrado), es un conjunto de aplicacions sobre el .net framework, permite constriur software desee un entorno desarrollo simple,

características: mulktilengujae, entorno de desarrollo integrado, diferentes tipos de proyectos (para hacer todo tipo de aplicaciones gracias a la bibliotecas o plantillas), ayuda y noticias en línea, intellisense, acceso a base de datos, diseño de clases, etc

estructura básica,

solución(puede acumular varios tipos de proyectos)

solución: dos tipos de archivos

sln(ahí tiene toda la configurarion de la soluciony de los proyectos contenido) (si se llega a eliminar este lo que se tendría que hacer es volver a vincularlo a una misma solucion)

suo (metadatos de personalizacion)(son las opciones que tendrá el usuario, como las ventanas que tendrá abierta en ese momento) (si se llega a eliminar este tipo de solución, no va importar muhco porque como son opciones del usuario no afectan en el proyecto)

proyecto, es lo que permite predefinir plantillas y es la parte mas importante. Es un contenedor que reside dentro de una solución y alberga uno o mas elementos de la aplicación y define el tipo básico de ella (exe,dkk,etc)

elementos hay dos tipos

elemento de la solución (pueden ser las carpetas, no son los comunes a la hora de crear el proyecto)

elementos de proyecto (son los mas utilizacidos como los .cs)

caja de herramientas:

despliega en forma agrupada los controles disponibles, arrastrar y soltar, personalizable, varios tipos * .net * com * html, codgi c&p

explorador de servidores (todas las maquinas que se encuentren conectada a la red, si se poseen los permisos suficientes, se puede a acceder a ella para ver los que se están ejecutando,)

- Administrar conexiones a base de datos
- Explorar contenido de servidores por ejemplo los servidores disponibles

Variables c# (es un espacio de memero que tiene la característica, que cambia su valor a lo largo de la ejecución, en c# hay que tener en ceunta el tipo, el nombre y los modificadores de acceso)

Importante inicializar las variables locales, es una calase no hace falta, pero si localmente si o si hay que hacerlo

var: es un lenguaje fuertemente tipado en c#,

dinamic: ese si puede cambiar su tipo a lo largo de la secuencia

c# en escencia es un lenguaje fuertemente tipado

tipos de datos

el object es el tipo base de estos tipos de datos porque se dice que de todas heredan de object, es decir que se puede asignar a cualquier tipo a object

aunque hay excepciones debedido en cuanto al menjo pro quu hay dos tipos de valores, por referencia y por valor ()

string es por referencia, como las clases

tipos decimales doublé, float, decimal

numero de . flotante (puede llegar a ser un numero muy grande o muy chico)

ventajas presentar valores muy grandes o muy chicas, sirve para las cuestiones científicas,

desventajas, puede fallar la excactitud (es la aproximidad que tendrá un resultado al valor real)

el tipo decimal es punto flotante, pero es tipo flotante decimal, puede guardar hasta números con 28 0, encomparacion del doublé que puede guardar 308, pero además de eso ocupa el doble de memoria que el doublé, 16, cuando es para manejar dinero, lo mejor es ocupar dinero, porque los errores de exactitud son menos regulares

ámbito de las vairables, es el alcance que tendrá esta dentro del código, durante la cual la vairable es validd y se la puede referenciar, puede a nivel de bloque (es una que esta limitada por las llaves{} una estructura repetitiva o condicional) y a nivel de procedimiento (los métodos, funciones serian las variables que se pueden acceder dentro de ese metodo) a nivel de clase (los campos) a nivel de espacio de nombre ()

se puede declarar varaibles en nivel de procedimiento que si bien no pueden ser referenciada fuerta de ese método, puede conservar ese valor si vuelve a ingresar ese emtodo, por eso se hace la isquisicion entre ámbito y duración de la variable, en este caso la duración no coincide con el ambtio, porque se puede mantener ese variable

c# es sensible a las mayúscula int a,A; a!=A;

comentarios los ocmentacion de una línea comenzacon con //

comentacion de una línea comenzando con /**/

cada ves que muestro un numero este se convierto en string, para ello se utilizan formateadores de números y hay fomateqadores por defecto, al cologar el formato :G17 muestra el numero completo en el caso de un decimal como 0,000000000001 y asi no aparece 0,01

el símbolo que es como la potencia es el or exlusivo en c#

&& y || se ejecutan por corto circuito es cuando cumple o no cumple con una propiedad la 2da no lo controla, en caso del y cuando la primera condición es falsa se sabe de antemano que todo será falso, y en un or si la priemra es verdad ya se sabe que el resultado será verdadero.

En caso que tenga una función dentro del mi and o or, y que estas tengas acciones que necesitan ser ejecutadas voy a necesitar que el resultado de esta función se vulque dentro de otra variable para que se pueda ejecutar por corto circuito. Por ello existe las situaicon que no son por corto circuioto andalso y orelse con las versiones sin corto cicuito

If ocmo función?

Switch se utilizan valores discretos y son para problemas simples y los casos que se utilzan es porque hace una mejor legelibidad

En c# 7 se agrego apareamiento, igualación de patrones, extendieron la capacidad de comprecion del swich, permite hacer evalucion de tipo de varaibles y agregar una sentecia when en cada case

Arreglos: un cojunto de variables que está relacionada entre si, cada elemento de lamtrziz puede ser accedido atraves de un índice, puede tener de 1 a n dimensiones(vector,matriz,tabla,cubo,etc)

Estrucutar repetitivas c#

Tratar de nunca tocar la variable que itera el for, hay que intentar no tecarla dentro de la iteración

Repite se ejecueta una ves hasta que sea falso, no se como el do while

El ensamblado es un contendor de código msil, pe, punto de entrada

Es un limite de seguridad,tipos, ámbito de referencia y versión.

Además es una unidad de implementación y ejecución similtanea

Cuando se habla una referencia externa se habla de un ensamblado, el ensamblado es la minima unidad de versionado que xiste dentro de.net por eso se puede poner un numero de versión, es la unidad de implmeentacion que sirve para cargar un ensablado. Ya que se puede hacer cargo simultanea de esamblados del mismo ensamblado pero en distintas versiones

Un ensablado puede tener una sola cosa o puede tener varios archivos a la ves

Manifestó: es un convenio de metadatos que describe a prio el ensamblad oque esta por recorrer

Los metadatos de tipos, es que el ensamblado tmb define cuales son los tipos que tmb están adentro

Un metadato es un dato que define otro dato. Es una descricipcio de los datos que tendrá ese ensamblado, tendrá el nombre la versión y la lista de archivos que este apuntando, en resumen seria una descripion de datos que muestra como se relaciónan los elementos de lensalbado entre si

Los recursos pueden ser un archivo de texto, un archivo xml, un audio una imagen

El código intenermedio (msil)

Dll son librerías dinámicas:

Metadatos de tipo

Tipos: ckases, estructuras, enumeració

Miembros : propiedades,métodos,eventos

Usado por : compilador, intelicense

Estos son los mas usados, en especial las clases tendrán atributos

Los ensamblados se pueden clasificar entre privados y compartidos

Es decir que abra una serie de ensablados que serán utilizados por el resto de las aplicaciones de .net y tmb esta los ensamblados de 3ro, que pueen estar en la caropeta de assembly, en esta carpeta solo debe ir los ensamblados que realemte serán útiles en diferentes tipos de aplicaciones para su uso, si no son necesarios hay que dejarlos en su carpetas nomas.

Si se desea guardar un ensamblado en gac se deberá generar una firmar ese ensamblado en una firma distitna al resto,con u nombre fuerte y se lo puede guardar con las herramientas deisntalacion, gac útil y con el mismo aplicación

La gac es un Reservorio de ensamblados, no se debe tentar en colocar todos los esanblados en esta, pero colocarlos se debe utilizar la herramienta de instalación

Ejecución del codgio administrado

Selección de lcompilador (lenguaje)-> compilación a msil -> compilación a código nativo -> ejecución del código

Selección del compilador:

Soporte muntilenguaje, compiladores provistos (c#,vb.net y c++) y compiladores de terceros

Compilación a msil (al compilar convetimos el código fuente en msil y metadatos)

En este paso es donde conseguimos el ensamblado, mas especidifcamente cuando compilamos conseguimos el ensamblado, puede ser un ejecutable o una dll, que tendra código de lenguaje intermedio intermedio, que será independiente al lenguaje de alto nivel que estemos ocupando para acceder a las bibliotecas, etc, y como se puede ejecutar ese código, en widonws existe un comando llamado comond object file format este tipo de archivo se considera un portable ejecutable, es decir que tiene una cabecera que muestra a quien debe llamar para que se peuda ejecutar, entoces el ensablado que en se momento debe ejecutar a la clr para ser ejecutado, eso lo ahce el sistema operativo.

Compilación a código nativo

Antes de poder usar msil dene convertise a código de maquina, aquí el compiñador jit adecuado

Las compilaciones sucesivas ya utilzian esta compilaciones ya guardadas

Ejecución de código, la clr maneja la ejecución de la plicacion (administrada)

Proceso de compilación jit y ejecución posterior continuo

Además de recolección de elementos no utilizados, seguridad, interoperabilidad con código no administrado, etc

Es el proceso que se basa en la administración de memoria y la compilación del codgio faltante, y cuando se ejecute todo este hace la la ejecución del código de memoria en, la clr se va a encarga de la recollecion de basura

Ejecución del código: el corazón del clr esta físicamente representado por mscoree.dll(common object reutime execution engine)

El class loader se encargacargar las bibliotecas necesarion para la ejecución del programa y platfomspecidgic, se encarga de la traducción del código o compilación y lo hace miembro a miebro

Developer command promp vs poner ildasm

El mscorlib es el nucleo del framework, el nucle de la biblioteca de clase base, es una dll que esta apuntando

Ussing solamente sirve para acortar la escritura en caso cuando utilizamos por referencia algún programa

Los espacios con como carpetas lógicas, que sirve para agrupar clases,

Organizar grandes cantidades de clases , construcción de bibliotecas de clases, para invocar a utilizar una clase entro de una jerarquía namespace, se utilizan puntos

- Miespacio.smisubespacio.miclase

Una subconjunto de esto se denomina biblioteca de clase

Un subconjuto de estas librerías que se proveen con cada implementación de . net se conoce como las libreria de clases base(]Bcl) esta expuesta para cualquier implmentacion.

Es un subconjunto que ebe estar entodas las implementaciones de la bcl estándar es la que viene del framwork estándar

Según el estándar ECMA 335, la librería de clases base es parte del eprfil kernel es los que se necesita si o si, entonces para que debería contar cada implmentacion del framwor para proveer funcionalidad a esa implementación. Provee tipos incircos, archivos,seguridad, manejo de strings.

Hay que diferencia de lo que se conoce como .net standard que es compatible con todas las implementaciones y otra cosa es el bcl , en relación ecma 335, que seria lo que se debería hacer y lo que se deben basar los que quieran hacer otra implementación del framwork

.net estándar es una especificación formal de la api, que pretende estar dispo en todas las aplicaciones de .net, se puede compilar directamente al .net estándar

No se puede apuntar un programa que tiene una implmentacion diferente a otro por ej frame y core, pero si se puede hacerlo desde versiones distitnas

Las versión estándar es recién compatible con el framwork 4.6 hacia arriba

Privado:

Protegido: es para la herencia, va aser visible de una clase heredada desea, a noser que alguna de las clases privadas, si una clase hace el llamado de otra clase, solo puede llamar a los campos públicos, internay y protectediunternal. Es accesible cuando es heredado

Publico

Internal: los internar se puede utilizar en cualquier parte del ensamblado, no se podría utilizar este campo en caso que se haga un llamado en caso que llame a una clase que llamo un integer anteriormente

protectedInternal los campos internal además de la herencia también se los pueden ocupar dentro de clases que llaman a otra que tiene este tipo de campo

privateProtected no funciona fuera de assembly, aunque vea heremncia, asi que solmaente funcióna dentro de una herencia que esta dentro de lassembly

el namaspace no determina delimitar accesibilidad solmanete sirve para ordenar

tipos de .net

clases (son tipos por referencia, estos valores se guardan en el monton o en el heat)

estructuras (la diferencia que hay que entre las clases y estas, es que son definidas por el usuario, son un tipo por valor, es similar a los lenguajes estructurados, este tipo de valor se guardan en el stack osea la pila)

enumeraciones son un tipo particular que tmb depende del tipo por valor,

interfaces son tipos por referencias

delegados

una clase permite definir los atributos (campos. Una clase permite definir los atributos y métodos comunes de varios objetos

que diferencias sintácticas existen entre funciones y procedimientos en c#

las funciones retornan un valor mientras que los procedimiento serian los meotdos, que no retornar ningún valor al cuerpo

crear un objeto usamos la palabra new para reservar memoria.

Constructores: se utilizan normalmente para instanciar objetos de una clase, el cosntructor es invocado al crear una instacia, puede no estar presente,

Modificadores de acceso:

Establecen el alcance de un tipo y sus miembros

Public : irrestricto, cualquier código puede ver este elemento

Protected código dentro de la clase y sus clases derivdadas gerarquia de herencia

Internal código dentro del ensamblado

Protected internal unión de protected e internal

Private protected protected per osolo dentro del ensamblado

Private código dentro del tipo donde se declara el elemento

Tipos de pasajes de parámetros

Hay dos, por valor y por refencia

El alcance de un miemro (clase, meotdo, etc) es el ámbito en donde puede ser accedido, es establecido por los modificadores

Modificador : caracteristica

Niinguo : copia valor (referencia por valor)

Out pasa variable por referencia en dereccion de salida (se espera que salgan valores)

Ref pasa varabiel por referencia e n ambas direcciones (referencia)

In pasa variable por referencia en dereccion de ENTRADA (se espera que venga una referencia desde afuera, pero se evita su modificación dentro del metodo)

PARAMS COLEECION DE PARAMETROS OPCIONales del mismo tipo

Los parámetros mas comunes son por valores o por referencia

Pasaje Por valor (se puede decir que se hace una copia)

Pasaje Por referencia (es como que estamos apuntando a esa posición por memoria, por lo cual no se hace la copia del valor que estoy pasando)

Pasaje y variables son diferentes :

Campos/ acc/mut y propiedades

Campo:

Accesores y mutadores: es algo que podamos ver

La idea es juntar lógicamente el get y set, en una unidad lógica que seria la propiedad

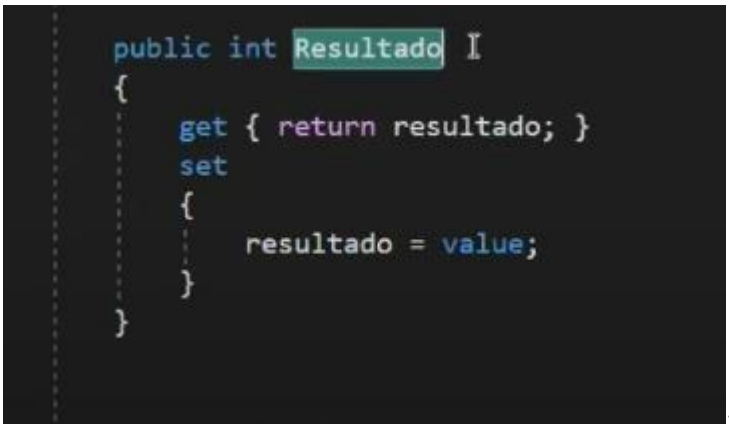
Getters y setters (son métodos comunes)

En c# tiene algunos datos mas z

Se necesita estos accesorios y mutadores porque se requiere , para cumplir los pilares de OO, la herencia, atributos priv y métodos públicos

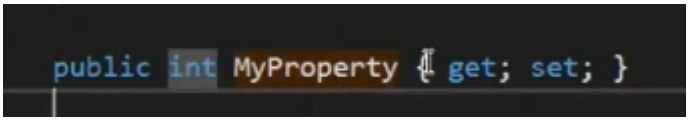
Propiedad

Propiedadades completas:



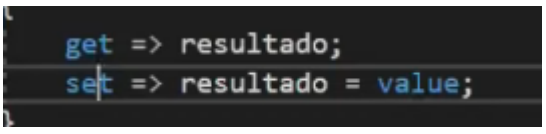
+

Propiedades autoimplementadas:



El campo no existe a nivel del codigo fuente, esto solamente se hace para realizar el encapsulamiento

Propiedades cuyo cuerpo es una expresión : son loscaso en donde su get y set se pueden expresar en una sola línea



La propiedades se utiliza cuando se necesita devolver o optener atributos o para los atributos, por eso se espera que no sean complejos, sino que permitan encapsular esos atributos. Seria una versión especializada de los métodos

Generalmente:

Campo -> privado

Propiedad -> publico

El sintatic suggar es una terminología que se utiliza, cuando hacen el código mas simple para el programador

Las constantes son un una acción que hacen que una variable que no pueden ser modficadas durante la ejecución de la aplicación

Las constantes no necesitan una instanciación de la clase

```
public class ConstTest
{
        class EjConst
        {
                public int x;
                public int y;

                public const int c1 = 5;
                public const int c2 = c1 + 5;

                public EjConst(int p1, int p2)
                {
                        x = p1;
                        y = p2;
                }
        }
        static void Main()
        {
                EjConst mC = new EjConst(11, 22);
                Console.WriteLine("x = {0}, y = {1}", mC.x, mC.y);
                Console.WriteLine("c1 = {0}, c2 = {1}",
                EjConst.c1, EjConst.c2 );
        }
}
```

Enumeraciones : una enumeración es un conjunto de constantes fijo al cual se le asigna un nombre, pueden ser solo de la famiia de enteros

Arreglos en c# t

Todo arreglos hereda de la clase base array y son un tipo por referencia

Permite utilizar funciopnalidad como:

Lenght (devulve la longitud)

Clear (borrar)

Copyto(copiar el arreglo a otro)

Reverse (revertir el arreglo)

Sort (ordena el arreglo)

Los arreglos pueden ser de cualquier tipo

Puede ser dos arrgelos

Arreglo muntidimencional rectangular hay que oiber yba coma en el medio [,]

Arreglos multidimensional jagged

Hay dos arreglos anidados

Array.sort

Manejo de fechas:

System.datetime no se lo puede considerar un tipo intrisenco, sino que seria un tipo formado por otros tipo a la ves, este representa un instante de tiempo en particular,

Puede ir del año 1 al 9999

Mas usados

Para sumar es add

O add year, month, day , etc

Now, fecha hora y acutial

Dayofweek dia de la semana

Sistema.timespan para hacer operaciones con fecha y hora se recorre a time span

Cuando se maneja dos fechas y se buscar ver el periodo de tiempo entre estas dos fechas

Depuración y errores

La depuración (debugging) se refiere al proceso que hacemos para intentar encontrar y corregir errores en una app,

Modos de compilación: perfiles de compilación

Debug : sirve para hacer esa depuración, la cual genera info para hacer la depuracion, paso a paso el cambio de variables, como resultado de esto, e l código no esta optimizado,

Reléase: es un ensamblado que optimizado para ser ejecutado de una forma mas eficiente

Tipos de errores

Compilación

Son los errores que nos impiden que nuestro programa comience a ejecutarse de entrada

De ejecución

Son aquello en el compilador funciona, pero en el momento de la ejecución encontró un incoveniente, son los errores que arrojan una excepción

Tratando de obtener el limite superior de un arreglo con una dimensión inexistente debería ser 0 y no 1

Y lógicos

Son los errores que se producen cuando hay un mal funcionamientro en el código, por lo cual va a producir errores

Herramientas disponibel

Punto de interrupción (brakpoints) interrupir la ejecución en el momento exacto que lo dije z|

Ejecución paso a paso

Ayudan a solucionar errores

Editar mientras se ejecuta

Manejo de excepciones se considera una manejo estructuras, solamente se utiliza cuando noi se puede utilizar otra lógica de codificación otros errores, debido a que el trycatch consume mucho los recursos y además para mejorar la legibilidad del codigo, ejemplo cuando se intenta crear un archivo de texto , cuando no se puede conectar a una base de dato, en estos casos se usa un try catch para intentar capturar el error

Manejo de decisión estructuradas : se lo considera estruturada porque con trycatsh, esos errores se mantienen ailado dentro de una estructura al habitar el error , siempre y cuando sea capturados

Lo no estructurado se considerar entrar y ejecutar un código por un lado que no sean las interfaces que no sean las adecuados debido a la legibilidad

Partes

Try : donde puede ocurrir el error

Nullreference arrogar un error de manera artificial con throw

Catch que este realciondo con el nullrefence hace que todos los errores que pasen el try pasaran aca, sino se iria al otro catch

Puede haber un catch amplio el cual maneja todos los errores sin importar su naturaleza

Y si hay una error con un manejo especifico se utiliza el manejo de errores especidifos,

El try catch debe ir de los especifico a los mas general

Finally vea o no error siempre se va a ejecutar finally por eso es el lugar adecuado para hacer generalmente limpieza, osea liberar recursos