



**POLO
TECNOLÓGICO**
MUNICIPAL - MINA CLAVERO

CURSO INTRODUCCIÓN NET 6.0

CLASE 14 - HERENCIA



Ministerio de
Desarrollo Productivo
Argentina



Municipalidad de
MINA CLAVERO
VALLE DE TRASLADERA - CORDOBA - ARGENTINA

Clases e Instancias

¿Que es un Objeto?

- Es un **agregado** de datos y procedimientos que permiten manipular dichos datos. También conocido como instancia.

¿Que es una Clase?

- Es la **definición** de las características concretas de un determinado tipo de objetos.

Miembros de una clase

Son los datos y funciones de los que van a disponer todos los objetos de la misma.

Campos

- Es un **campo común** a todos los objetos de una determinada clase.

Métodos

- Es un **conjunto de instrucciones** a las que se les asocia un nombre dentro del objeto.

Propiedades

- Es una **mezcla** entre el concepto de Campo y Método, fuera del objeto se ve como “un campo”, pero tiene código asociado a él.

Eventos

- Es un **mensaje** que dispara un objeto, en determinada circunstancia. Este mensaje, puede ser capturado por un método de otro objeto.

Constructor / Destructor

No es más que un **método** definido en la clase que tiene el mismo nombre de la clase pero sin ningún valor de retorno ni tampoco **void**.

Este método, puede recibir los parámetros que sean necesarios y es el encargado de “**inicializar**” el objeto.

En realidad devuelve una **referencia** a la dirección de memoria donde se creo el objeto.

Se puede escribir otro método que se llama igual que la clase precedido por el símbolo **~**, el cual es llamado cuando se destruye el objeto. Este método se conoce como **Destructor**.

Sobrecarga de métodos

Es la posibilidad de **definir**, en la misma clase, varios métodos con el **mismo nombre**, siempre y cuando tomen diferente número o tipo de parámetros.

Esto es posible ya que cuando se les llame el compilador sabrá a cual llamar **a partir** de los *<parámetros>* pasados en la llamada.

No se permite definir varios métodos que sólo se diferencien en su valor de **retorno**.

Herencia

Es un mecanismo que permite definir nuevas clases a **partir de otras** ya definidas, de modo que, si en la definición de una clase indicamos que ésta hereda (o deriva) de otra, entonces a dicha clase “a la que se le suele llamar **clase hija**” será tratada por el compilador automáticamente como si en su definición incluyese la definición de la cual hereda “a la que se le suele llamar **clase padre** o **base**”.

A los miembros definidos en una clase **hija** se le **añadirán** los que hubiésemos definido en la clase **padre**.

En .NET, una clase, **no puede** heredar de mas de una clase, (herencia múltiple), pero **si** puede implementar múltiples interfaces.

Herencia *cont...*

Modificadores de clases

- Usados en la definición de la **clase**
- **:** – Define que la clase en cuestión hereda de otra clase.
- **abstract** – Permite la creación de “clases abstractas”, ya que nunca se instancian y siempre deben heredarse.
- **sealed** – Permite la creación de una clase “sellada”, ya que no puede heredarse.
- Acceso calificado con: **base**

Herencia *cont...*

Modificadores de miembros

- Usados en la definición del **miembro**
- **static** – Permite compartir un miembro en todas las instancias de la clase.
- A los miembros estáticos se los suele llamar como miembros de clase.
- A los que no son estáticos, se les suele llamar miembros de instancia.

Herencia *cont...*

Modificadores de acceso

- Cada miembro define su propia accesibilidad.
- **public** – Accesible por todo el mundo.
- **private** – Accesible dentro del tipo. Ni las instancias ni los tipos derivados lo pueden acceder.
- **protected** – Accesible dentro del tipo y en las clases derivadas, pero no desde otras clases.
- **internal** – Accesible dentro del tipo y del ensamble. Las demás miembros dentro del ensamble pueden accederlo.
- **protected internal** – Accesible dentro del tipo y en las clases derivadas, y accesible por los demás miembros de la clase dentro del ensamble, pero inaccesible fuera del mismo.

Herencia *cont...*

Sobreescritura de métodos

- Métodos que **sobrescriben** a otros con la misma declaración. Se definen en cada clase derivada.
- **override** – Usado al declarar nuevamente el método o propiedad.
- **virtual** – Define que ese miembro puede sobrescribirse.
- Si no se especifica **virtual** el miembro **no** se puede sobrescribir.
- **abstract** – Define que ese miembro no se implementa en esa clase, sino en las clases derivadas
- **new** – Oculta un miembro de la clase padre. Nota: Al usar **new**, estamos haciendo un “shadow” del miembro que estamos sobrescribiendo.

Herencia *cont...*

Todas las clases heredan implícitamente de **System.Object** que esta definida en la BCL por lo que dispondrán de todos los miembros de esta. (Si no son ocultados)

Algunos de sus miembros son:

- Equals()
- GetHashCode()
- ToString()
- GetType()
- Finalize()

Interfaces

Una interfaz es la definición de un conjunto de métodos para los que **no se da implementación**, sino que se les define de manera similar a como se definen los métodos abstractos. Es más, una interfaz puede verse como una forma especial de definir clases que sólo cuenten con **miembros abstractos**.

Una clase **puede** heredar, **múltiples** interfaces.

Todo tipo que derive de una interfaz ha de dar una implementación de **todos** los miembros que hereda de esta, por esto se dice que es como un “Contrato”.

Usar Interfaces para definir comportamiento, y usar Herencia para reutilizar el código.

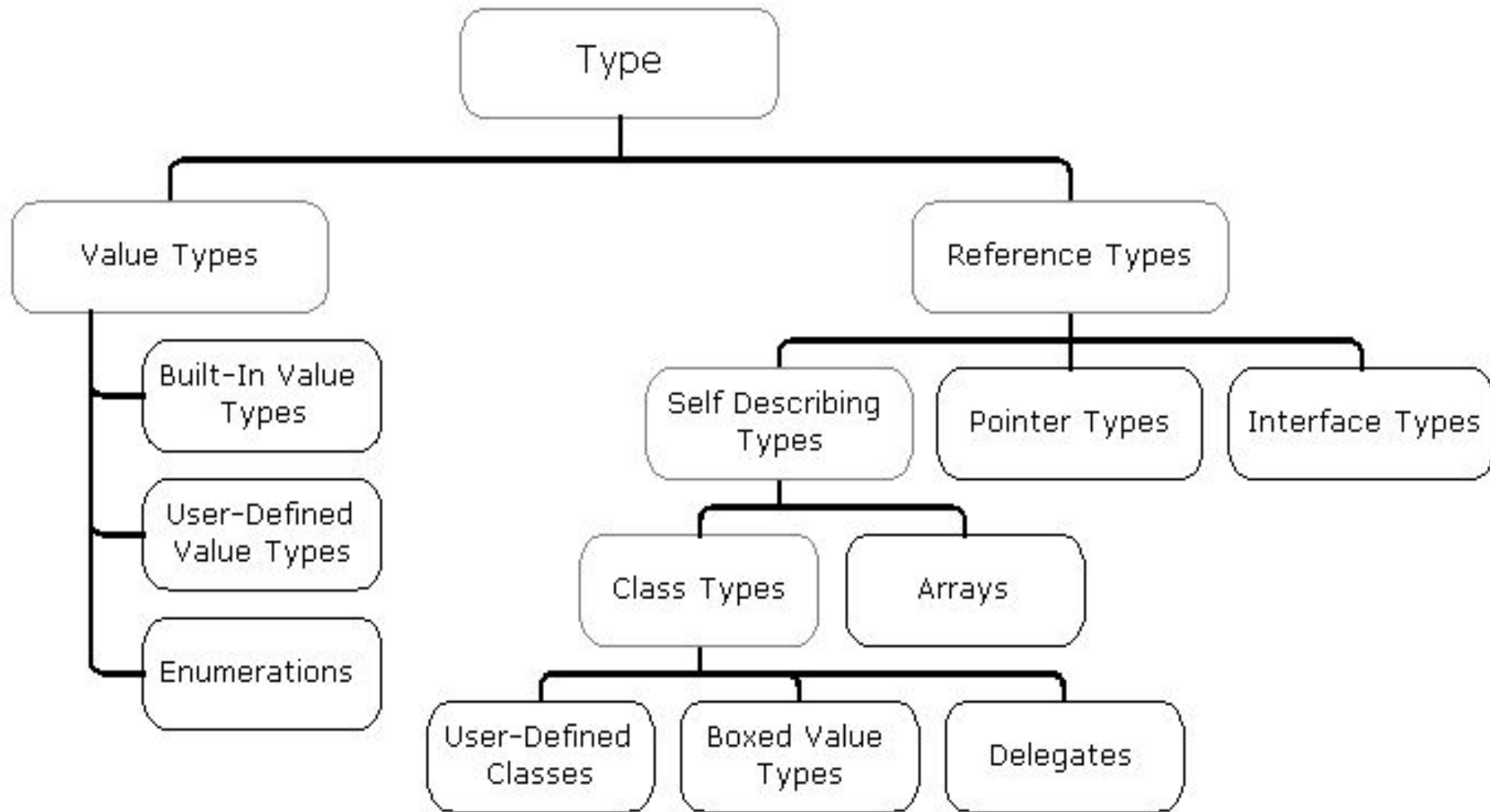
Estructuras (Structure)

Tipos definidos por el usuario.

Como “clases livianas”

- Son tipos de valor, en cambio las clases son por referencia.
- Pueden implementar Interfaces.
- No pueden heredarse.

Tipos por valor o referencia



Delegados

Punteros a funciones orientados a objetos.

Pueden apuntar a un **método en particular** de un objeto específico.

Se utilizan, por ejemplo, para **capturar** los eventos disparados por los objetos.

Polimorfismo

Es la capacidad de almacenar objetos de un determinado tipo en variables de tipos antecesores del primero, a costa claro está, de sólo poderse acceder a través de dicha variable a los miembros comunes a ambos tipos.

Casting: Relacionado íntimamente con el polimorfismo. Es el acto de “**convertir**” un tipo padre, al **verdadero** tipo del objeto.



**POLO
TECNOLÓGICO**
MUNICIPAL - MINA CLAVERO

¡MUCHAS GRACIAS!



Ministerio de
Desarrollo Productivo
Argentina



INCLUIT



Municipalidad de
MINA CLAVERO
VALLE DE TRASLADERA - CORDOBA - ARGENTINA