**Herencia**

La herencia en la programación funciona del mismo modo que en la vida real, un padre puede dejar en herencia ciertas características y atributos a sus hijos.

Aprendamos a utilizar la herencia en nuestro código y veamos como una clase hijo puede heredar ciertas características o funciones de la clase padre, también veremos que estas funciones o métodos heredados se pueden sobrescribir.

Al momento de programar nuestras clases padre podemos usar la palabra clave **abstract** para que dicha clase solamente pueda ser heredada, pero nunca instanciada. Por el contrario, tenemos la palabra clave **sealed** permite generar instancias de la clase, pero no permite heredarla.

**Abstracto**. Ejemplo que esta clase es solo una idea

* Una clase abstracta (asbtract) puede ser heredada pero no instanciada
* Una **clase sellada (sealed) no puede ser heredada pero si instaciada**

**Polimorfismo**

En polimorfismo un objeto hijo que hereda de una clase padre puede ser tratado como un objeto padre, pero al ser convertido en objeto padre ya no se podrá acceder a los atributos del objeto hijo. Por otro lado, un objeto padre no puede tratarse como un objeto hijo a menos que el objeto padre estuviera guardando un objeto hijo.

**Lista de Objetos polimórfica**

En C# podemos realizar validaciones sobre el tipo de objeto que estamos manejando, usando las siguientes palabras clave:

* **is**, para verificar si un objeto es de un tipo en específico.
* **as**, para tratar un objeto como un tipo específico, en caso de no poder convertir el objeto entonces va a asignar un valor null.

**Implementación de interfaces**

En C# no existe la herencia múltiple, sin embargo, esto no es una limitante para hacer que un objeto se vea como otros objetos diferentes ya que podemos recurrir al uso de interfaces.  
Una interfaz es la definición de la estructura de un objeto. Las interfaces las podemos tomar como contratos en las que obligamos a que los objetos cumplan determinadas características.

**Parámetros de salida**

Los parámetros de salida son una manera de retornar valores en un método, para que esto se lleve a cabo se debe ante poner la palabra “out” seguido del tipo de dato en la firma del método.

**Diccionario Polimórfico**

El objeto tipo Dictionary necesita dos parámetros, el primero es la llave y el segundo es el contenido. Haciendo que la instancia de un Dictionary quede de la siguiente forma:

Dictionary<**int**, string> diccionario = **new** Dictionary<**int**, string>();

**Eventos**

¿Qué es un evento?  
Los eventos están en el núcleo del desarrollo de aplicaciones con C#. Permiten basar la lógica de la aplicación sobre una serie de procedimientos y de funciones que se ejecutan cuando alguno de sus componentes solicita la ejecución. Es el caso, por ejemplo, de los componentes gráficos: estos pueden desencadenar eventos cuando el usuario realiza alguna acción como, por ejemplo, la selección de un elemento en una lista desplegable o hacer clic sobre un botón.

En .Net Core se debe agregar el método que se quiere que se ejecute al evento usando el +=, ejemplo:

**AppDomain.CurrentDomain.ProcessExit += AccionDelEvento;**

**private** **static** void **AccionDelEvento**(object sender, EventArgs e) {

Printer.WriteTitle("Saliendo"); Printer.Beep(3000, 1000, 3);

}

En el **AppDomain** se refiere al contenedor de eventos de todo lo que está pasando en la App, el **CurrentDomain** se refiere a lo que está pasando en la actualidad y el **ProcessExit**, se refiere al momento cuando se está cerrando la aplicación.

El += es el utilizado para agregar el método “AccionDelEvento” el mismo se va a ejecutar al momento de cerrar la aplicación.

Los eventos tienen una particularidad y es la sobrecarga de los eventos, se puede volver a utilizar el mismo manejador del evento para agregar un nuevo evento.

**Linq o Language Integrated Query**

**try** { // Abrir conexion a la BD // Conectarse a la BD y hacer cosas }

**catch** (E) { // Error por X motivo }

**finally** { // Cerrar conexion a la BD // No importa lo que pase, SIEMPRE hay que cerrar la conexión a la BD. }