**POLIMORFISMO.**

Para que haya un polimorfismo tiene que haber herencia.

Poli= “muchos”

Morfismo=”forma”

El polimorfismo es las muchas formas que puede tomar un objeto.

Las clases hijas heredan los atributos y los métodos de la clase base.

Las clases hijas pueden sobrescribir los métodos de la clase base(override).

**CLASES Y METODOS ABSTRACTOS.**

Una clase abstracta solo se utiliza como clase base y no se puede instanciar objetos de esta clase (no se implementan métodos, solamente se definen).

Una clase abstracta es muy genérica.

Sirve para dar una idea de lo que tendrán las clases hijas (ej :métodos)

Los métodos abstractos obligan si o si a que la clase sea abstracta.

Los métodos de la clase base abstracta obliga si o si a que se implementen sus métodos en las clases hijas.

Una clase abstracta puede contener atributos variables.

**INTERFACES**

Las interfaces fueron creadas por el problema de que una clase solamente podía heredar de otra clase.

Las clases pueden heredar múltiples interfaces.

Las interfaces no se pueden instanciar.

Pueden contener atributos pero solamente estáticos y a su vez se pueden definir como final (no admitirá cambios después de su declaración/ no puede ser sobrescrito o redefinido)

**ENCAPSULAMIENTO**

Es el hecho de restringir el acceso a las funciones internas de una clase.

Con esto logramos :

controlar la manera en que los datos son accedidos o modificados.

El código es más flexible y fácil de cambiar a partir de nuevos requerimientos.

Poder modificar una parte del código sin afectar a otras partes del mismo.

Tipo de datos por:

Valor (se almacenan en una parte de la memoria estatica llamada stack).

Ej : los valores de las variables.

Referencia (se almacenan en una parte de la memoria dinámica llamada heap).

Ej : posición de memoria en donde esta almacenado la variable.

Constructores.

Tienen el mismo nombre que su clase, son públicos y no tienen ningún valor de retorno.

Los constructores pueden ser sobrecargados utilizando un numero diferente de parametos.

Los constructores no se heredan.

SQL

SELECT

- Mostrar todos los datos de los Clientes de nuestra empresa:

SELECT \* FROM clientes

INSERT

-Agrega a la tabla clientes (1,Emiliano,17,caballito)

INSERT INTO clientes (ID,nombre,edad,barrio) VALUES (1,Emiliano,17,caballito)

UPDATE

-Modifica con estos datos(parrotta, 25,PCH) la tabla clientes en el ID 1.

UPDATE clientes SET nombre = parrotta , edad = 25 , barrio = PCH WHERE ID = 1;

DELETE

-Elimina el cliente con id 1 en la tabla clientes

DELETE FROM clientes WHERE ID = 1;

Mostrar los clientes que no sabemos a qué región pertenecen (o sea, que no tienen asociada ninguna región) :

SELECT \* FROM Customers WHERE Region IS NULL

Mostrar los clientes que no sabemos a qué región pertenecen (o sea, que no tienen asociada ninguna región) :

SELECT \* FROM Customers WHERE Region IS NULL

Mostrar los clientes cuyo nombre empieza por la letra “W”:

SELECT \* FROM Customers WHERE CompanyName LIKE ‘W%’

Mostrar los empleados cuyo código está entre el 2 y el 9:

SELECT \* FROM Employees WHERE EmployeeID BETWEEN 2 AND 9

Mostrar los clientes cuya dirección contenga “ki”:

SELECT \* FROM Customers WHERE Address LIKE ‘%ki%’