**Abstracción -**

**Clase –**

**Objeto -** Objeto es un elemento que representa una entidad abstracta o concreta, un objeto contiene sus atributos y métodos.

**Ej.** Un objeto puede ser una bicicleta, que contiene de atributos llantas, frenos, material entre otros atributos propios, y el método puede ser informar el tipo de freno, el tipo de llantas que también puede ser un objeto o el material:

**Class** Bicicleta{

**private** ***string*** freno;

**private** ***string*** tipoLlantas;

**private** ***int*** nLlantas;

**pubic** Bicicleta() {

freno=“xxx”;

tipoLlantas=“sss”;

nLLantas=“2”;

}

}

**Atributo -**

**Método -**

**Encapsulamiento -** El encapsulamiento la subdivisión de una entidad en varias partes menores que contienen sus propios atributos y métodos que no se pueden acceder directamente. De modo a proteger sus características para que no se pueda acceder directamente, así solamente a través de métodos se puede realizar alteraciones.

**Ej.** el modo como se protege una variable de la entidad, es utilizando el prívate, de modo que solo se puede realizar cambios sobre esta variable dentro de la clase a través de métodos propios, un ejemplo es cuando se tiene un sistema de control de finanzas, que no son todos que pueden tener acceso a los valores:

**private** ***int*** totalMoney;

**public** ***int*** get valueTotalMoney(){

**return** totalMoney;

}

**public** ***int*** set valueTotalMoney(***int*** value){

this.totalMoney = value;

}

**Herencia -**

**Polimorfismo -**

**diferencia entre tipo primitivo y referencia -** Una variable tipo primitivo, puede guardar solamente un valor de su tipo especificado, cuando se asigna otro valor a esta misma variable, el valor en su interior será sustituido, generalmente son inicializadas con “0”, las variables de tipo referencia son utilizadas haciendo referencia a la localización en la memória de un objeto, estos objetos pueden contener varias variables primitivas y también métodos, generalmente son inicializadas con “null”.

***int*** cantidadManzanas = 0;

***String*** tipoManzana = null;

cantidadManzana = 50; //Su valor es sustituido por 50

tipoManzana = “manzana orgánica”; //se asigna un conjunto de caracteres para describir su tipo

**Diferencia entre usar private, protected, public, default -**

**Clase abstracta -**

**Interfaz -** La interfaz es un conjunto de acciones que deben ser ejecutadas, cada clase puede ejecutarlas de manera distinta, la interfaz contiene valores constantes o valores asignados a través de los métodos, que deben ser implementados dentro de la clase, generalmente son clases complementares abstractas, define lo que la clase deberá hacer.

**interface** Interfaz {

**public *void*** metodoAbstracto();

**public *String*** otroMetodoAbstracto();

}**public *class*** Principal **implements** InterfacePrincipal{

**public *void*** metodoAbstracto() {

}**public *String*** otroMetodoAbstracto() {

**return** “retorno";

}