ABSTRACCIÓN

Cuando consideramos las características y funciones principales de un objeto estamos realizando la tarea de abstraer, por ejemplo si queremos considerar el objeto balón empezaremos pensando en que es esférico por lo que una característica puede ser el radio, otra el material, otra el color, peso, si está o no inflado y/o dañado, el propietario; además, consideramos que tiene la función de ser golpeado o lanzado, de rodar, de ir a una velocidad y así realizaríamos la tarea de abstraer al objeto balón.

CLASE

En una clase es donde realizamos la descripción del objeto que ya abstrajimos entonces definimos todos los atributos formalmente, esta clase puede ser parte publica y parte privada, donde la parte publica describe las operaciones que realiza cada objeto de la clase y la parte privada describe los datos y operaciones que manipulan estos datos (encapsulamiento).

OBJETO

Es un ejemplar concreto de una clase y utiliza todo lo que en ella exista por ejemplo para la clase balón se puede crear un objeto que se llame miBalon y pueda decir que miBalon tiene un radio de 20cm , es de color azul, etc.

ATRIBUTO

Son las características de una clase, puede ser int, double, String, boolean, entre otros que definen como se constituye una clase y sus instancias; pueden ser públicas, privadas o protegidas dependiendo la protección que se desee manejar.

METODO

Es una de las funciones que puede realizar un objeto o una clase ya que en un método están un conjunto de instrucciones para el momento en el que es invocado por ejemplo la clase balón tiene como método inflar o rodar y cuando se llama a alguno de estos métodos para el objeto mibalon este realiza la instrucción

ENCAPSULACION

Es una manera de proteger atributos que no queremos que se accedan fácilmente desde fuera de la clase, queremos que exista un filtro que valide si se puede o no modificar el atributo de nuestro objeto. Por ejemplo para el objeto mibalon el radio solo podemos definirlo una vez y tiene que ser un radio entre un rango definido, igual su color y otras características; por lo que es necesario que para acceder a estos atributos tengamos que llamar a un método público que los modifique y controle su acceso como pueden ser los métodos getting y setting.

HERENCIA

Es una gran herramienta que permite la transmisión de métodos y atributos de una clase a otra (madre a hija) que permite establecer una jerarquía entre clase por ejemplo nuestra clase balon puede heredar a otra clases como balonDeFutbol, balonDeVoleibol, balonDeBaloncesto, etc. sus características y funciones para que estas hagan uso de ellas y además agreguen sus propios atributos y métodos; Por ejemplo el radio que por ley debe tener un balon de futbol es entre 68 y 70 cm , uno de basquetbol masculino debe medir entre 75 y 78cm, tiene diferentes materiales y peso.

Además estas clases pueden heredar a otras como balonMicroFutbol, mientras la clase madre no tenga la definición de “final” puede heredar a otra sus características.

POLIMORFISMO

Permite definir distintos comportamientos de un método en las clases de una jerarquía por ejemplo cuando llamamos al método inflar(); para la clase balonDefutbol este debe inflarse a una presión de entre 9 y 10.5 libras mientras que si llamamos el método inflar(); para un balón de baloncesto este debe inflarse con una presión entre 7 y 9 libras. Para cambiar los métodos heredados es necesario sobre escribirlos.

Diferencia entre tipo primitivo y de referencia.

La diferencia entre los tipos primitivo y de referencia radica en que las variables primitivas almacenan un solo dato dentro de un rango de valores, int, doble, etc.

mientras las variables de tipo por referencia se refiere a objetos y como tal se almacena espacio en memoria para los datos pertenecientes a la clase.

Cuando un entero posee un valor y este se iguala a otro entero se crea un entero diferente que posee este valor en memoria, mientras que si se le asigna una referencia de un objeto a otro se le asigna esta dirección en memoria y por ello puede ser modificado modificando alguna de estos objetos.

Diferencia entre usar private, protected, public, default.

Las diferencias entre estos tipos de declaraciones radica en la accesibilidad que poseen sus atributos a otras clases. Usar private al declarar las variable de una clase hace que estos datos solo puedan ser modificados por la misma clase, ya sea por los métodos públicos setting o en la clase. Usar protected hace que las variables sean accesibles solo por la clase que las posee y las clases que hereden de esta, además de las clases dentro del package, El uso de public como su nombre lo indica hace que todos las clases de cualquier parte puedan acceder y modificar estas variables desde cualquier lugar. Cuando no se especifica nada, se usa default las variables solo pueden ser accedidas por la misma clase y las del package.

Clase abstracta

Cuando se habla de una clase abstracta nos referimos a una clase que puede contener variables que cualquier clase que herede de esta debe contener, como la clase abstracta figura, cuya representación física no se puede imaginar como una sola, ya que un circulo, triangulo o cuadrado son figuras, teniendo esto en cuenta sabemos que estas figuras poseen un área, perímetro, lados, etc, estas variables pueden ser declaradas en la clase figura e implementadas de diferentes formas por las cases que hereden de esta.

Interfaz

Es una lista de métodos que que hacen algo en concreto pero la clase decide como implementarlo, el uso de interfaces en una clase hace necesario que se implementen todos los métodos de la interfaz