Introducción a la Poo

Definiciones

- Variables, métodos:
  - Públicos (accesibles)
  - Privados (inaccesibles)
- Herencia: Permite a una subclase derivar de una superclase adquiriendo sus métodos y atributos.
- Polimorfismo: Técnica que permite que objetos de diferentes clases respondan a un método de manera específica.

CLASE:

- Tipo de datos y operaciones que describen el comportamiento de un conjunto de elementos.

OBJETO

- Ejemplar concreto de una clase, es definido por las operaciones de la clase a la que pertenece.

ATRIBUTO

- Cada uno de los datos de una clase, presente en todos los objetos de esa clase. Una clase puede tener o no tener atributos.
- Los atributos de una clase se consideran como atributos globales.
- Estado: Conjunto de valores de los atributos de un objeto en un momento dado

MÉTODO

- Definición de una operación o comportamiento de una clase

MENSAJE

- Invocación de un método sobre un objeto *.algo();*

Ejemplo

- Clase: Persona
- Objeto: ClienteGimnasio.
- Atributos: Edad, sexo, nombre.
- Métodos: Saltar, correr, descansar
  - ClienteGimnasio.saltar(2);

Definición de objetos

- Un Programa es un conjunto de objetos que interactúan entre si.
- Un objeto se puede componer de otros objetos más simples.
- Cada objeto pertenece a un tipo en concreto.
- Los objetos del mismo tipo tienen comportamientos idénticos.
- Una operación es una función que se puede aplicar a cualquier objeto de una clase.
- Los objetos utilizan referencias de memoria.
  - Antes de construir un objeto hay que declarar una variable del tipo de su clase.
  - Para crear un objeto usamos el operador *new*
  - New busca un espacio en memoria y construye el objeto devolviendo la referencia del objeto.

CONSTRUCTORES

- Métodos especiales invocados para crear e inicializar un objeto recién creado.

NULL

- Referencia nula a ningún bloque de memoria.
- Una variable se inicializa por defecto en null.
- Acceder a una referencia nula da un error.
    - Persona per1 = new Persona(); //Esto hace referencia al objeto
    - per1 = null //per1 no hace referencia a nada.

SOBRECARGA de métodos

- Permite tener diferentes métodos con el mismo nombre, para ello el número de parameros de entrada han de ser diferentes.
    - Parámetros de entrada: van entre paréntesis, compuestos por un valor primitivo (y un nombre de variable en los constructores) separados por " , "

MODIFICADORES DE VISIBILIDAD

- Una clase será visible por otra dependiendo de si se encuentran en el mismo paquete y de los modificadores de acceso que utilice.
    - Estos modificadores cambian su visibilidad entre clases mostrándola u ocultándola.
    - Podemos modificar la visibilidad entre miembros de distintas clases, que atributos y que métodos son visibles.

<u>Modificadores de acceso para clases</u>

- Clases vecinas: Las del mismo paquete.
- Clases externas: Las de distintos paquetes.

VISIBILIDAD por defecto

- Clase definida sin utilizar modificadores de acceso
- Solo visible para clases vecinas.

VISIBILIDAD total

- Clase definida por el operador de acceso *public*
    - Visible para cualquier otra clase.
    - Necesita ser importada.

<u>Modificadores de acceso para miembros</u> //hace referencia atributos y métodos

- Que un *atributo* sea visible permite modificarlo.
- Que un *método* sea visible permite invocarlo
- Para que un miembro sea visible lo ha de ser también su clase.
- Un miembro siempre es visible para su clase.

TIPOS DE VISIBILIDAD (de miembros)

- Por defecto: Cuando definimos un miembro sin valor de acceso.
  - Solo visible para clases vecinas.
- Pública (*public*): Visible para cualquier clase.
- Privada (*private*): Solo visible para su clase.
- Protegida (*protected*): Solo visible para sus clases vecinas y subclases aunque sean externas.

MÉTODOS GET Y SET

- Creados por convención entre programadores para ocultar los atributos.
- SET permite asignar un valor a un tributo pudiendo validar o manipular ese valor antes de asignarlo al atributo.
  - set*nombre*();
- GET devuelve el valor del atributo o bien alguna transformación de este.
  - Get*nombre*();

<u>Más definiciones</u>

- Dos variables anidadas no pueden tener el mismo identificador, excepto si una de ellas es una variable local de un método ya que tiene prioridad.
  Se dice que la variable está oculta.

ELEMENTOS ESTÁTICOS

- Son métodos y variables que pertenecen a la clase en vez de al objeto.
  - Se declaran insertando la palabra "*static*" después del indicador de visibilidad.
  - Para invocar métodos y atributos estáticos usamos el nombre de la clase.
- Dentro de un método estático no podemos referenciar (emplear this) un atributo ni un método que no sea estático.
  - Dentro de un método estático si podemos instanciar (crear) objetos.
- Cualquier atributo de la clase puede ser declarado como estático añadiendo la palabra "*static*" antes de la declaración de su tipo.
  - Los atributos estáticos son útiles para situaciones donde los datos deben de ser compartidos por múltiples objetos de la misma clase.

Palabra reservada THIS

- Para la manipulación de atributos
- Variable de referencia que se refiere al objeto actual.
  - Se emplea dentro de un método de instancia o de un constructor para acceder a los miembros del objeto actual tales como variables de instancia, métodos y constructores.

ASOCIACIÓN

- Agregación: Indica que una clase es parte de otra clase.
  - Los componentes pueden ser compartidos por varios compuestos.
  - La eliminación de un componente no elimina el objeto.
- Composición: La clase contenida existe gracias a la clase contenedora.
  - Los componentes forman una parte del objeto compuesto.
  - La eliminación del objeto conlleva la eliminación de los componentes.