“P.O.O”

Programación orientada a objetos

Por otra parte podemos encontrar que estas clases cuentan con modificadores de acceso los cuales se asignan a métodos y variables como los podemos ver en el anterior diagrama, aparte de esto hay varios tipos:

(+) Publico: Permite al acceso a este desde cualquier otra clase

(-) Privado: En este caso solo tenemos acceso a este únicamente usando la clase en la que se instancio

(#) Protegido: Nos da un acceso público para clases derivadas o heredadas y uno privado para el resto de clases

( ) Sin modificador: Tenemos acceso al elemento desde cualquier clase que se encuentre dentro del archivo

El paradigma de la programación orientada a objetos (POO) básicamente está enfocado en simular o modelar todos los objetos que encontramos en la vida real, un ejemplo de esto podría hacer los zapatos que podemos encontrar en una tienda de ropa

|  |
| --- |
|  |
|  |
|  |

Los métodos y las variables tiene que ser comunes en todos los zapatos y si llegásemos a usar otra cosa tendríamos que usar variables y métodos que fácilmente encontraríamos en cada objeto como un animal acuático como el pez las variables básicas serian tamaño, edad, género, color y raza por otro lado los métodos básicos serian nadar, comer, defecar, reproducirse. De esta manera la abstracción de pez enmarca todos los tipos de peces que se pueden definir con esas características

Según el anterior diagrama podemos ver que las clases se pueden representar gráficamente a través del lenguaje unificado de modelado (UML)

Por otro lado podemos decir que un objeto es la instanciación de una clase para dar a entender mejor esto podríamos poner otro ejemplo: Decimos que las clases son el modelo del zapato estándar con apenas lo básico para poder ser un zapato como si habláramos de una plantilla que nos ayuda a crear zapatos cada uno con cualidades idénticas a las de nuestra clase (molde o plantilla), estos zapatos creados de nuestra plantilla llevarían el nombre de objetos y todos los objetos creados con la misa clase tendrán variables iguales y métodos iguales

+Tipomateriales = cuero y caucho

+Color = verde

+Talla = 40

Definen cada una de las acciones que puede hacer el objeto

“verbo en infinitivo”

Definen o brindan una descripción de las cualidades del objeto

“sustantivos”

Método

Variable

+cuidarLospies ()

+Abrigarlospies ()

+Correr ()

+Caminar ()

+Saltar ()

+Escalar ()

ZAPATO

Para la creación de las clases podemos encontrar dos métodos uno que es el constructor este está encargado de inicializar el nombre de la clase la cual tendrá todos los valores para inicializar las variables, este se ejecuta automáticamente al momento de instanciar una clase, aparte de ello existe otro método con el nombre de destructor este es opcional e igualmente se ejecuta automáticamente es usado para destruir una clase, liberar memoria, cerrar archivos, desconectarse de base de datos o despedirse del usuario:

Podemos ver un ejemplo simple de cómo sería el método constructor:

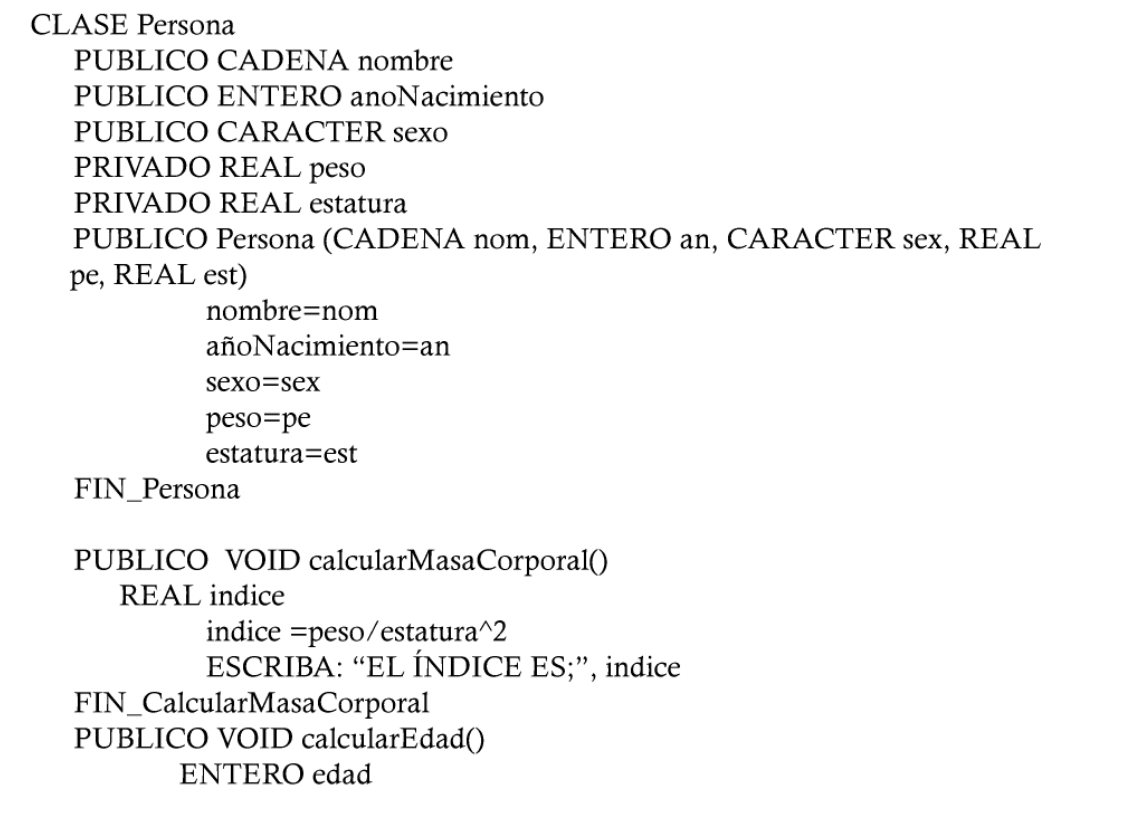
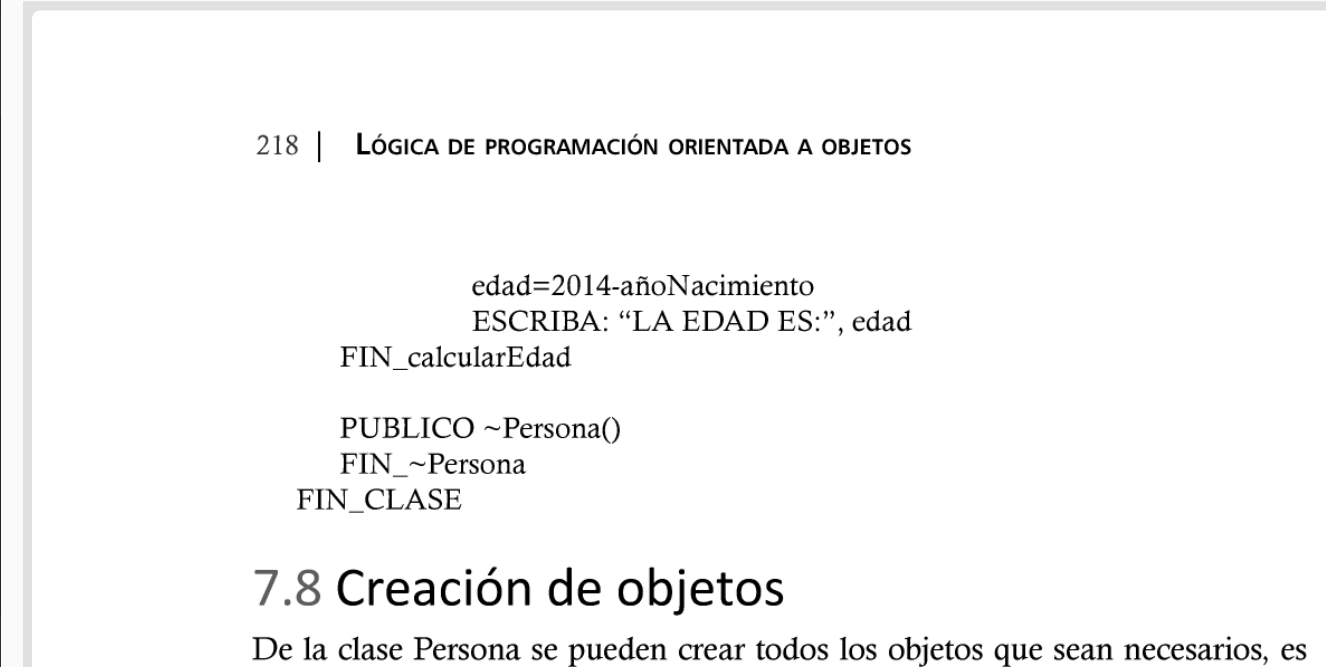
PUBLICO Persona ()  
 ESCRIBA: “creando un objeto de la clase persona”  
Fin\_persona

Y por otra parte tenemos al método destructor el cual se usa anteponiendo “~“al nombre de la clase

PUBLICO Persona ~ ()  
 ESCRIBA: “creando un objeto de la clase persona”  
Fin\_ ~ persona

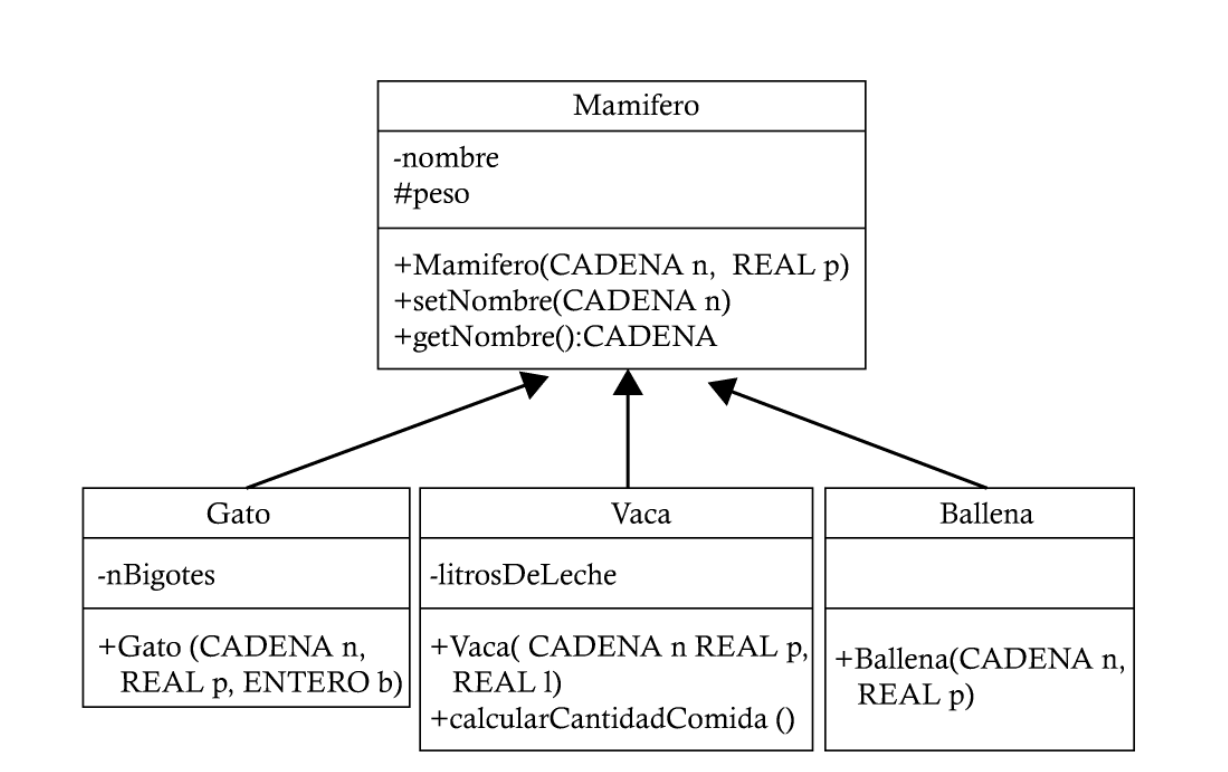
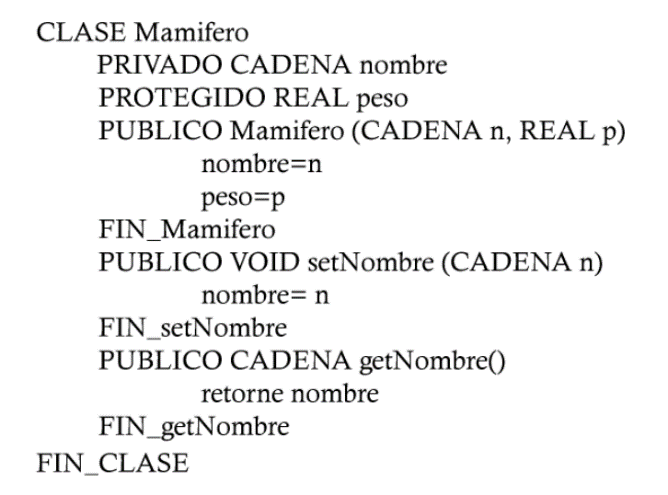
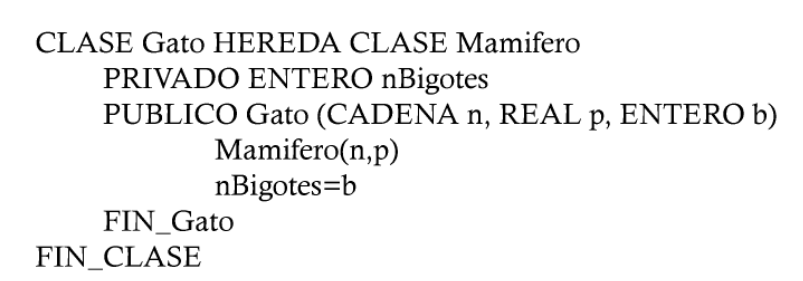
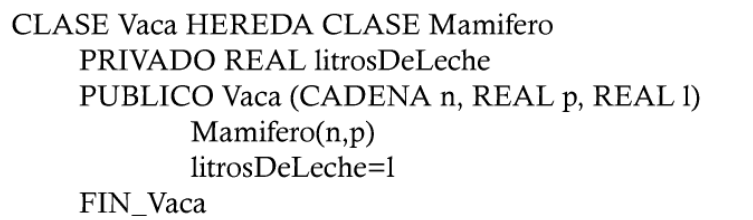
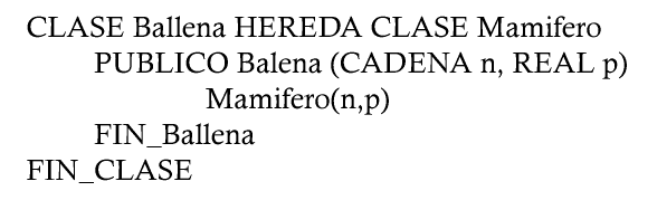
PUBLICO Persona ()  
 ESCRIBA: “creando un objeto de la clase persona”  
Fin\_persona

A continuación daremos un ejemplo de cómo es una clase persona expresada directamente desde la estructura estándar del código de la programación orientada a objetos



Una vez completemos el desarrollo de nuestra clase en la que de cierta manera le explicamos lo que es el objeto de una persona a nuestra maquina informática podemos proceder con la creación de objetos provenientes de nuestro molde de clases lo cual tiene el nombre de instanciación



De igual manera podemos encontrar lo que sería el código de este diagrama primeramente desarrollamos el código de la superclase

Por otro lado podemos entontar tambien la herencia lo cual básicamente es interpretado literal mente como la palabra lo indica la herencia poniendo el ejemplo de personas habla de cómo la clase padre hereda variables y métodos a sus hijos.

De tal forma que evitamos rescribir código dando asi paso a la reutilización de ciertas partes de código, esta es una de los grandes beneficios que podemos encontrar en la programación orientada a objetos

En la herencia contamos con una clase denominada superclase o clase base esto para la clase padre y en la clase hija podemos encontrar los nombres de subclase o clase heredada

Esta tambien puede ser representada por medio de los diagramas los cuales muestran la herencia por medio de una flecha negra que va de la subclase o clase hija apuntando a la clase base o clase padre, para tener mayor claridad de esto mostramos el siguiente diagrama con un ejemplo de poo:

Una vez este esté terminado podemos elaborar las herencias de esta superclase y para inicializar las variables heredadas de la superclase es necesario invocar al constructor de la clase padre

## 

Pagina: 219 de 444

P.O.O : 219 – 292 (73)pg

Link: https://sena-primo.hosted.exlibrisgroup.com/permalink/f/1i756fj/TN\_cdi\_elibro\_books\_ELB70431

Título: Lógica de programación orientada a objetos