POO/OOP

Programación Orientado a Objetos

Object-Oriented Programming

Vocabulario

- Clase, objeto
- Ejemplar de clase, instancia de clase, ejemplarizar una clase, instanciar una clase
- Modularización
- Encapsulamiento / encapsulación
- Herencia
- Poliformismo

Conceptos

- **Paradigma OO** (Orientado a Objetos)
  El modelo se basa en el **objeto**
- Un **objeto** es aquello que tiene:
  - Estado (acciones y reacciones a mensajes)
  - Identidad (propiedad que la diferencia de los demás objetos)
- La **estructura** y **comportamiento** de objetos similares están definidos en una **clase** común
- Los términos **objeto** y **instancia** son intercambiables. Un **objeto** es más que la plasmación, manifestación, **instanciación** de un modelo, plantilla, **clase**
- Una **clase** es un conjunto de **objetos** que comparten una **estructura** y **comportamiento** común

Clases y objetos

- Clase

Una **Clase** es un modelo donde se redactan las características y comportamientos comunes de un conjunto de **Objetos**

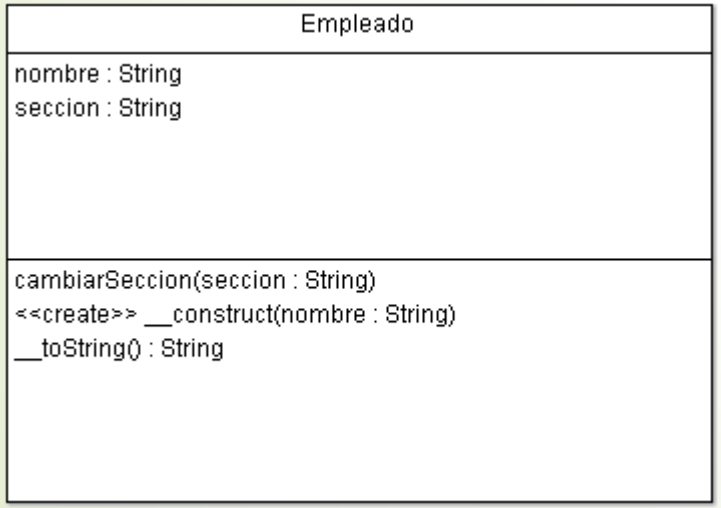
Clase Mamifero

Objetos

Clases vs objetos

Clases y objetos parecen conceptos similares pero existe una clara diferencia conceptual entre los mismos.

Las **clases** son un **concepto estático** definido en el **programa fuente**, son una **abstracción** de la esencia del objeto



CLASE

Concepto estático
Programa fuente
Abstracción

Clases vs objetos

Los **objetos** son **entes dinámicos** que existen en **tiempo** y **espacio**, que ocupa memoria en **memoria** en la ejecución de un programa

OBJETO

Concepto dinámico
Existen en tiempo/espacio
Ocupan memoria
Materialización

Objetos

- Accediendo a propiedades y comportamientos (pseudocódigo)
- Nomenclatura del punto

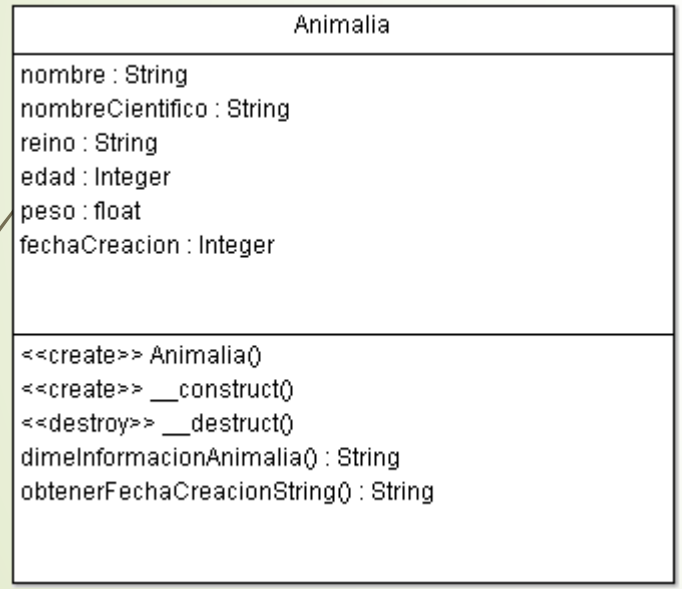
Objetos

Accediendo a propiedades
Perro.pelo = “largo”;
Perro.peso = 5;
Perro.edad = 3;

Accediendo a comportamientos
Perro.come();
Perro.pasea();
Perro.limpia();

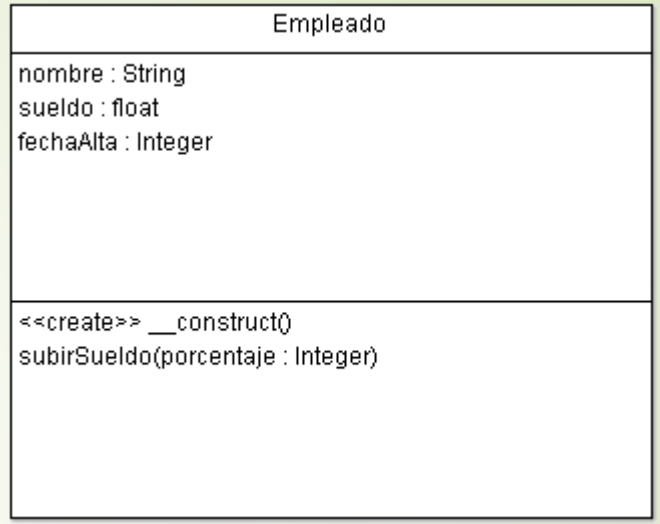
Diagrama de clases

Hay herramientas como ArgoUML que nos permite un diseño de forma visual de las clases y sus relaciones



Esto nos permite una mejor comprensión de nuestro proyecto, sobre todo cuando el modelo se complica.
Seguidamente pasaremos el modelo a código (Directorio poo_1)

Clase Empleado



Se necesita modelizar en una empresa los empleados de la misma por tanto:

- Crear la clase (plantilla/modelo) de la clase Empleado
- Tendrá varios campos de acceso público (**public**) que son el nombre, sueldo y la fecha de alta
- Crear un método constructor para establecer el estado inicial de los objetos
- Hay empleados que se les puede subir un porcentaje de su sueldo. Este porcentaje se pasará como parámetro al método subirSueldo