

La Orientación a Objetos



Métodos Estructurados y Métodos Orientados a Objetos

■ Métodos estructurados

- Origen en la programación estructurada (secuencia, ramificación, iteración, función).
- Pensar en términos de **estructuras de datos** que son manipuladas mediante **algoritmos**.
- Dificultades para adaptarse a requisitos cambiantes.

■ Métodos orientados a objetos

- Origen en la programación orientada a objetos.
- Más cercano al mundo real porque imita (supuestamente) su estructura: **objetos** que responden a **mensajes** enviados por otros objetos.
- Más estable frente a cambios en los requisitos:
 - responsabilidades de clases.
 - mayor cohesión entre datos y funciones.

Principios de la Orientación a Objetos

■ Abstracción

- Simplificación de la realidad.
- Retener los aspectos relevantes y prescindir del resto.
- El proceso de abstracción es relativo a un punto de vista particular.

— Clasificación

- Relación básica entre clases, y objetos clasificados en clases.
- Un objeto es instancia de una clase, la clase es su tipo.

— Encapsulamiento

- Separación de interfaz e implementación en una clase.

— Generalización

- La subclase hereda estructura y comportamiento de la superclase.

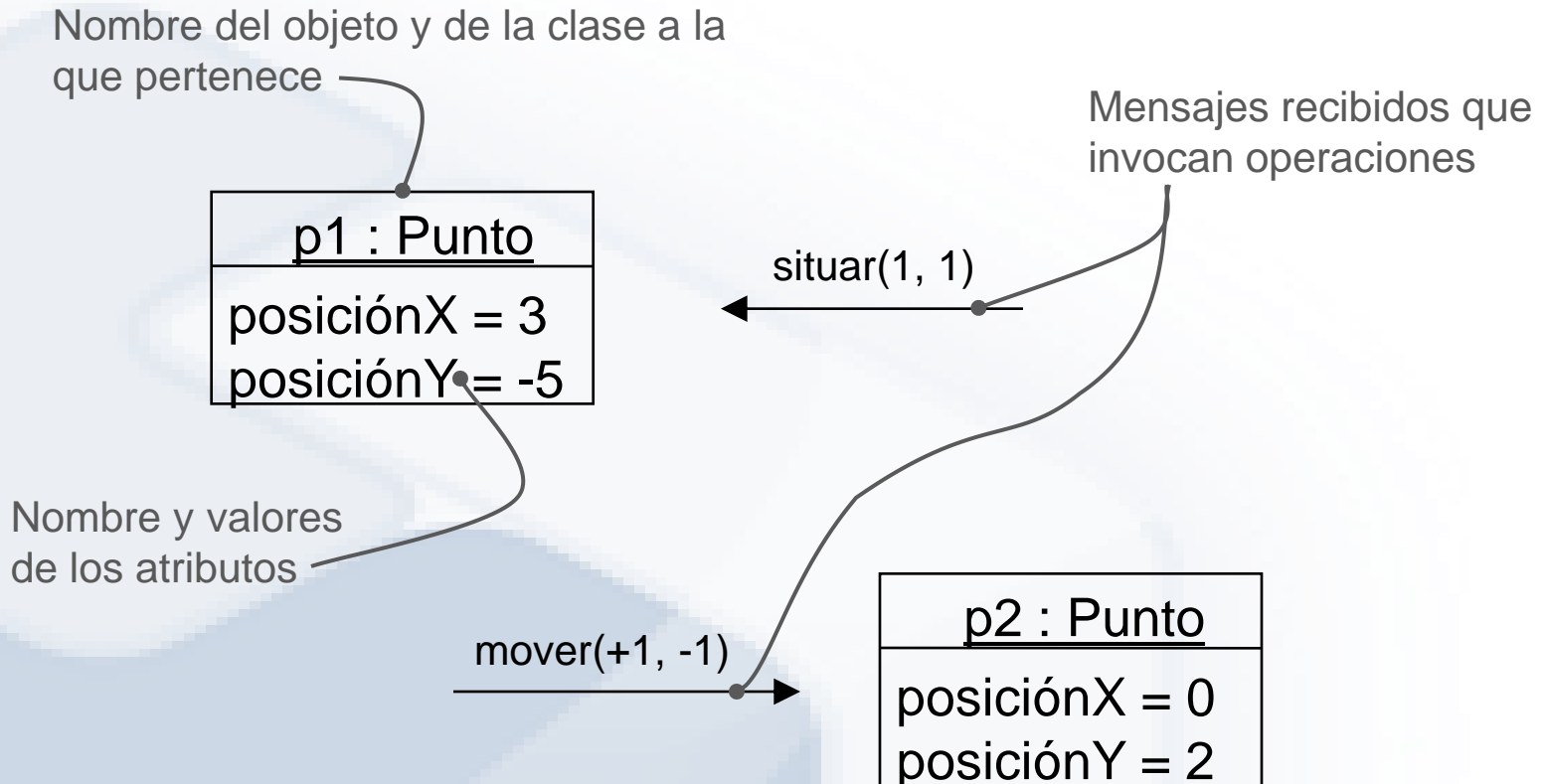
— Polimorfismo

- Capacidad de ejecutar distintas operaciones en respuesta al mismo mensaje.

Abstracción: ¿Qué es un Objeto?

- Representación, o signo, de una **entidad concreta**.
- Características de un objeto:
 - **Comportamiento**: operaciones o servicios que puede realizar, en respuesta a mensajes recibidos de otros objetos.
 - **Estado**: valores almacenados en el objeto como atributos, que son resultado de las operaciones pasadas.
 - **Identidad**: todo objeto tiene una identidad propia que lo distingue de otros objetos aunque tengan el mismo estado (no es un “valor” identificador).
- Entrelazamiento de estado y comportamiento.
- Abstracción: simplificación de la realidad.
 - Retener los rasgos relevantes y prescindir del resto.
 - Operaciones relevantes, valores relevantes.

Notación Básica de los Objetos

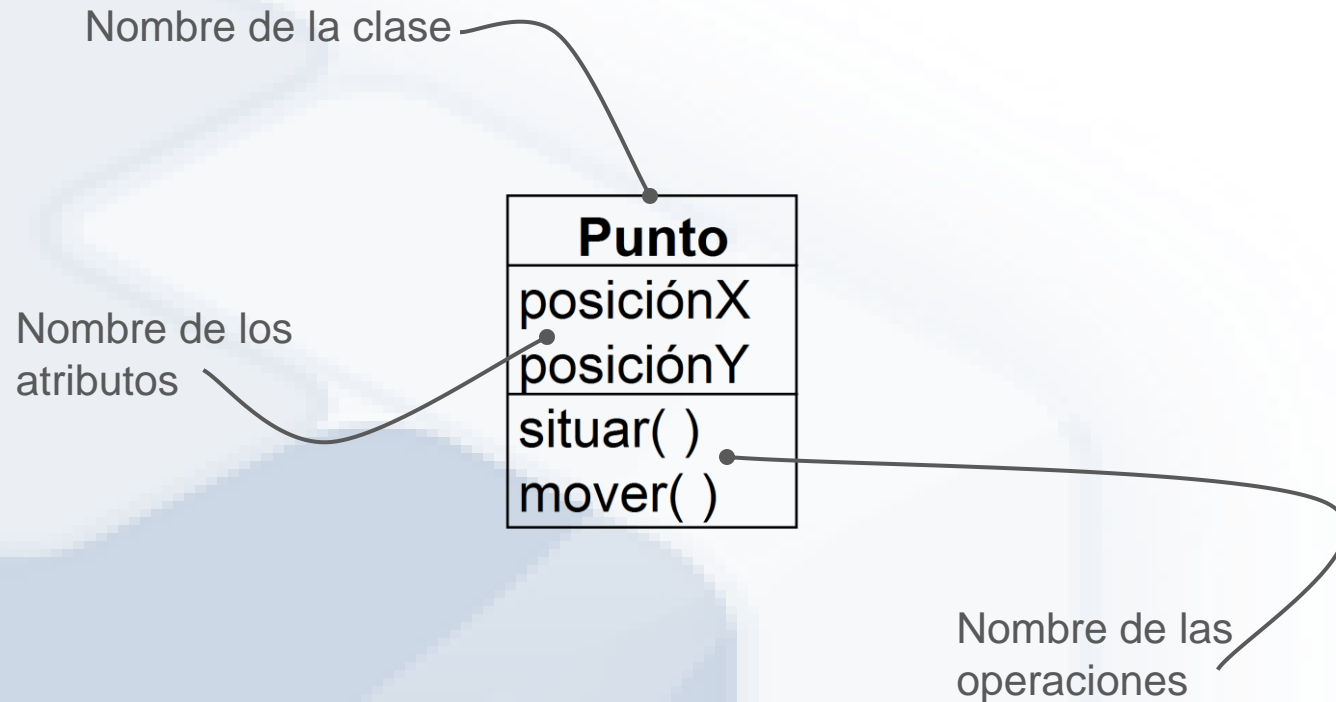


¿Son distinguibles estos objetos después de responder a los mensajes recibidos?

Abstracción: ¿Qué es una Clase?

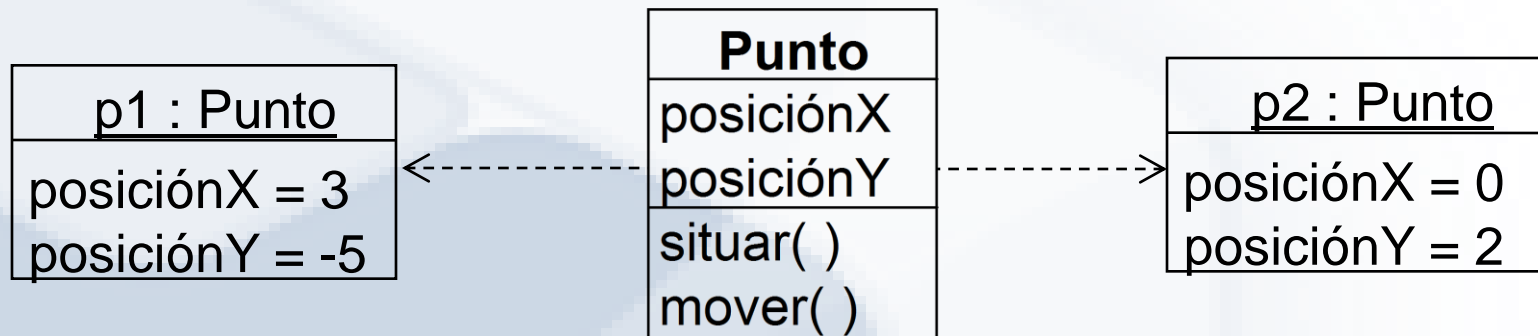
- Representación de un **conjunto de entidades**.
 - Extensión: los objetos.
 - Intensión: definición o especificación.
- Niveles de abstracción.
 - **Primer nivel** (objetos): representación de entidades concretas.
 - Sólo con sus rasgos relevantes de comportamiento y estado.
 - Prescindir de rasgos no relevantes de comportamiento y estado.
 - “Entidad concreta” no significa necesariamente entidad física o tangible.
 - **Segundo nivel** (clases): representación de conjuntos de entidades.
 - Definición o especificación de los rasgos relevantes.
 - Prescindir de los objetos y valores concretos.
- Una clase es la especificación de un conjunto de objetos con **estructura y comportamiento** comunes.
- En el modelo de análisis, las clases, atributos y operaciones corresponden a conceptos del **vocabulario del dominio**.

Notación Básica de las Clases



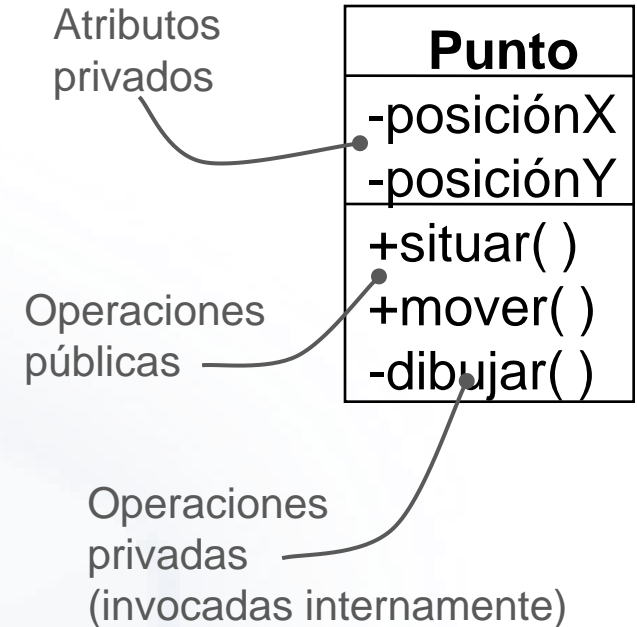
Clasificación

- Relación básica entre clases, y objetos clasificados en clases.
 - Un objeto es **instancia** de una clase, la clase es su **tipo**.
 - La clase se usa como **plantilla** para construir (instanciar) objetos.



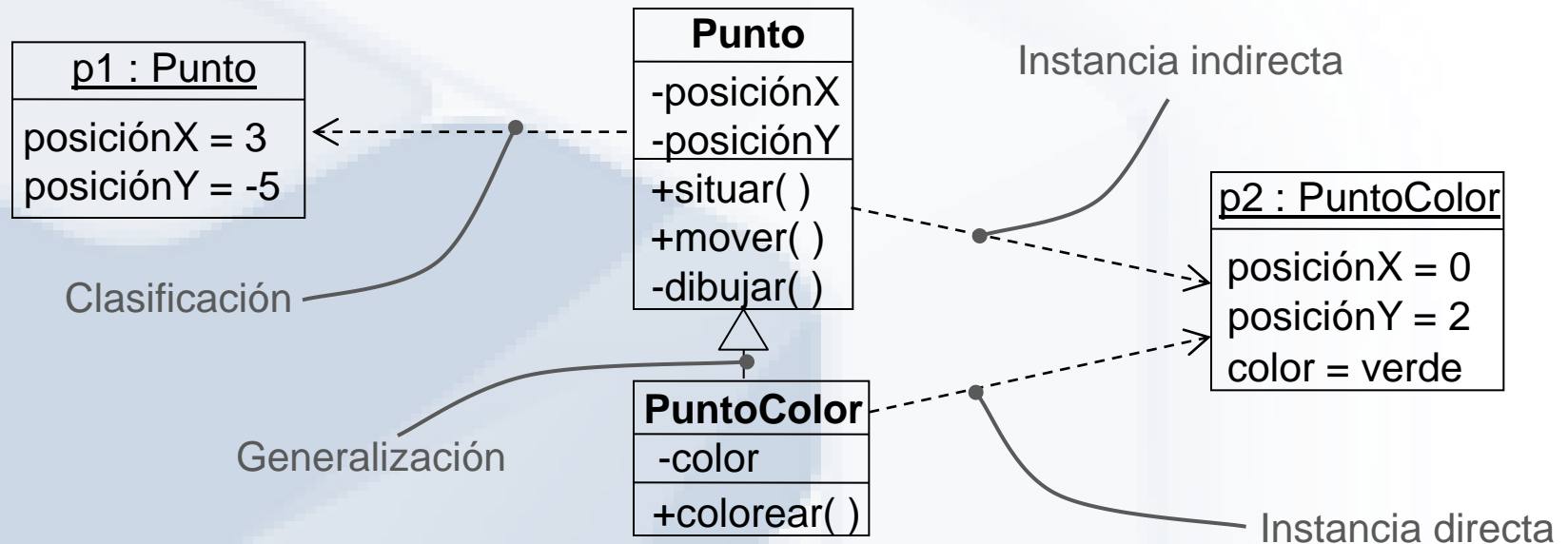
Encapsulamiento

- Separación de interfaz e implementación en una clase.
 - Interfaz: **vista externa** desde otras clases.
 - Implementación: **realización interna** de las estructuras de datos y funciones.
- **Objetivo:** reducir el acoplamiento entre clases:
 - Esconder la implementación, dependencia sólo de la interfaz.
 - Diseñar teniendo en cuenta sólo la interfaz, no la implementación.
- **Visibilidad:** típicamente todos los **atributos** son privados, y las **operaciones** son públicas si pertenecen a la **interfaz** de una clase, o privadas si son para uso interno: sólo se permite la **comunicación por mensajes**.



Generalización

- Relación básica entre una **superclase** y una **subclase**
 - Hereda la **estructura** y el **comportamiento** (atributos y operaciones), y
 - Puede **añadir** atributos y operaciones.
- **Principio de sustitución:** donde se espera una instancia de la superclase se puede poner una instancia de la subclase.



Polimorfismo

- Capacidad de ejecutar distintas operaciones en respuesta al mismo mensaje.
- Una **operación polimórfica** es aquella que tiene muchas implementaciones.
 - La subclase puede **modificar** la implementación de las operaciones heredadas.
 - El mismo **nombre de operación** puede designar implementaciones distintas.
 - **Objetos de distintas clases** en el mismo árbol jerárquico tienen operaciones con el mismo nombre pero con diferente implementación, por tanto pueden responder al mismo mensaje cada uno a su manera.

