

Programación orientada a objetos

Conceptos generales

Programación orientada a objetos

- Organización de los programas de manera que representen un modelo de la interacción de las cosas en el mundo real
 - Un programa consta de un conjunto de objetos que representan elementos del mundo real
 - Interesa *qué* hacen los objetos más que *cómo* se hace

Programación orientada a objetos

- La programación procedural (PASCAL, C) se basa en módulos de funciones
- Los datos y las funciones se consideran de forma separada.
- POO combina los datos y sus funciones

Programación orientada a objetos

- La POO proporciona una forma de pensar más natural sobre el problema, no sobre funciones y procedimientos
- Permite reusabilidad de código y soporta modificaciones de requerimientos o tecnología.

Programación orientada a objetos

- Cada objeto es responsable de unas tareas
- Los objetos interactúan entre sí por medio de mensajes
- Cada objeto pertenece a una clase (es un ejemplar de)
- Las clases se pueden organizar en una jerarquía con herencia

Objetos

- Los objetos son cosas
- Los objetos pueden ser simples o complejos
- Los objetos pueden ser reales o imaginarios



Métodos

- Un objeto puede realizar una serie de acciones
 - Definen la funcionalidad y comportamiento de un objeto
 - Son los mensajes para realizar una acción en un objeto
 - Equivalen a las funciones en otros lenguajes de programación



- Arrancar motor
- Parar motor
- Acelerar
- Frenar
- Girar a la derecha (grados)
- Girar a la izquierda (grados)
- Cambiar de marcha (nueva marcha)

método argumentos

Java POO 7

Datos o Atributos

- Valores o características de los objetos
- Permiten definir el estado del objeto u otras cualidades



- Marca
- Color
- Potencia
- Velocidad máxima
- Carburante
- Velocidad
- Aceleración
- Capacidad de combustible

variables

↑
constantes

Java POO 8

Clases

- Los objetos similares se agrupan en clases con estados similares y mismo comportamiento
- Las clases son "plantillas" que especifican el comportamiento y los atributos de los ejemplares (objetos) de la clase



Objetos de la clase
automóvil

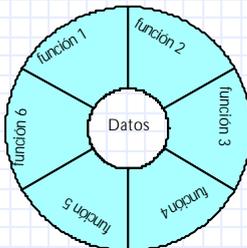
Java POO 9

Orientación a objetos

- Proporciona los siguientes mecanismos
 - Encapsulación
 - Herencia
 - Abstracción
 - Polimorfismo

Java POO 10

Encapsulación



Java POO 11

Encapsulación

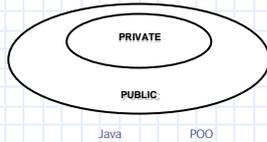
- Los objetos agrupan su estado (los atributos o datos) y su comportamiento (los métodos)
 - Relación clara entre el código y los datos
- Ocultación de información
 - Las partes necesarias para utilizar un objeto son visibles (**interfaz pública**): métodos
 - Las demás partes son ocultas (privadas)

Java POO 12

Encapsulación

Ejemplo

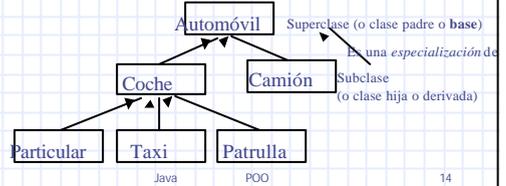
- El volante, el cuentakilómetros, los pedales y la palanca de cambios representan una interfaz pública hacia los mecanismos de funcionamiento del automóvil
- Para conducir no es necesario conocer la mecánica del automóvil (su implementación)



13

Herencia: jerarquía de clases

- Permite definir una clase especializando otra ya existente
 - Se extiende un tipo de datos, heredando las características comunes y especificando las diferencias
 - Permite la reutilización de código
 - En Java sólo existe herencia simple



14

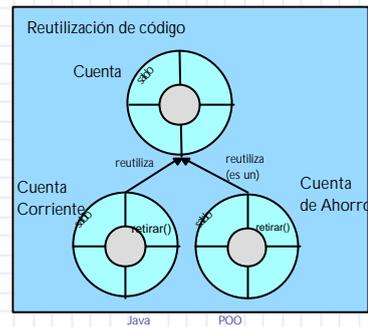
Herencia de clases

- ✦ Los métodos y atributos de la superclase son heredados por las subclases
- ✦ pero, en la especialización:
 - Se pueden añadir nuevos métodos y atributos
 - La clase Taxi tiene taxímetro y las operaciones poner en marcha taxímetro y para taxímetro
 - Se pueden redefinir los métodos
 - La clase Taxi redefine el método arrancar: si es recién subido un nuevo cliente, poner en marcha taxímetro
 - Se pueden ocultar métodos heredados
 - La subclase CocheAutomático oculta el método cambiar marcha

Java POO

15

Herencia de clases



16

Abstracción

- ✦ Capacidad de especificar las características comunes a un conjunto de clases
 - Definición parcial del estado y del comportamiento
 - Declaración del comportamiento (interfaz)
- ✦ Clases abstractas
 - Especificación de datos y comportamiento común a un conjunto de clases
 - Forzar a que las subclases proporcionen un comportamiento específico
- ✦ Interfaces
 - Declaración de métodos a incorporar en las clases que implementen la interfaz
 - Definición de constantes

Java POO

17

Polimorfismo

- ✦ Capacidad de solicitar la realización de una misma operación (mensaje) sobre distintos tipos de elementos
 - La realización concreta de la operación depende del objeto que reciba la petición
- ✦ Procesamiento genérico de objetos que:
 - Pertenecen a clases en una misma jerarquía
 - Pertenecen a clases que implementan un mismo interfaz
- ✦ Sobrecarga de método
 - Un mismo nombre de método y varias implementaciones en el mismo (en la misma o distintas clases)
- ✦ Sobrecarga de operadores
 - En Java no existe sobrecarga de operadores, excepto + para concatenar cadenas de caracteres

Java POO

18

Enlace dinámico

✦ Concepto relacionado con el procesamiento genérico de un conjunto de objetos

- Cuando una variable de un tipo Coche posee una referencia a un objeto de clase Taxi (siendo Taxi descendiente de Coche) y se invoca a un método por medio de la variable, ¿cuál se ejecuta si hay una implementación en Coche y otra en Taxi (p.ej. Arranca)?
- Los lenguajes con enlace dinámico tratan de asociar la ejecución al tipo más específico, por tanto a Taxi
- El mecanismo general de búsqueda es empezar por el tipo dinámico de la variable receptora y remontarse por la jerarquía de herencia hasta encontrar una implementación válida

Java

POO

19