

2. Nuevo Paradigma

Programación Orientada a Objetos

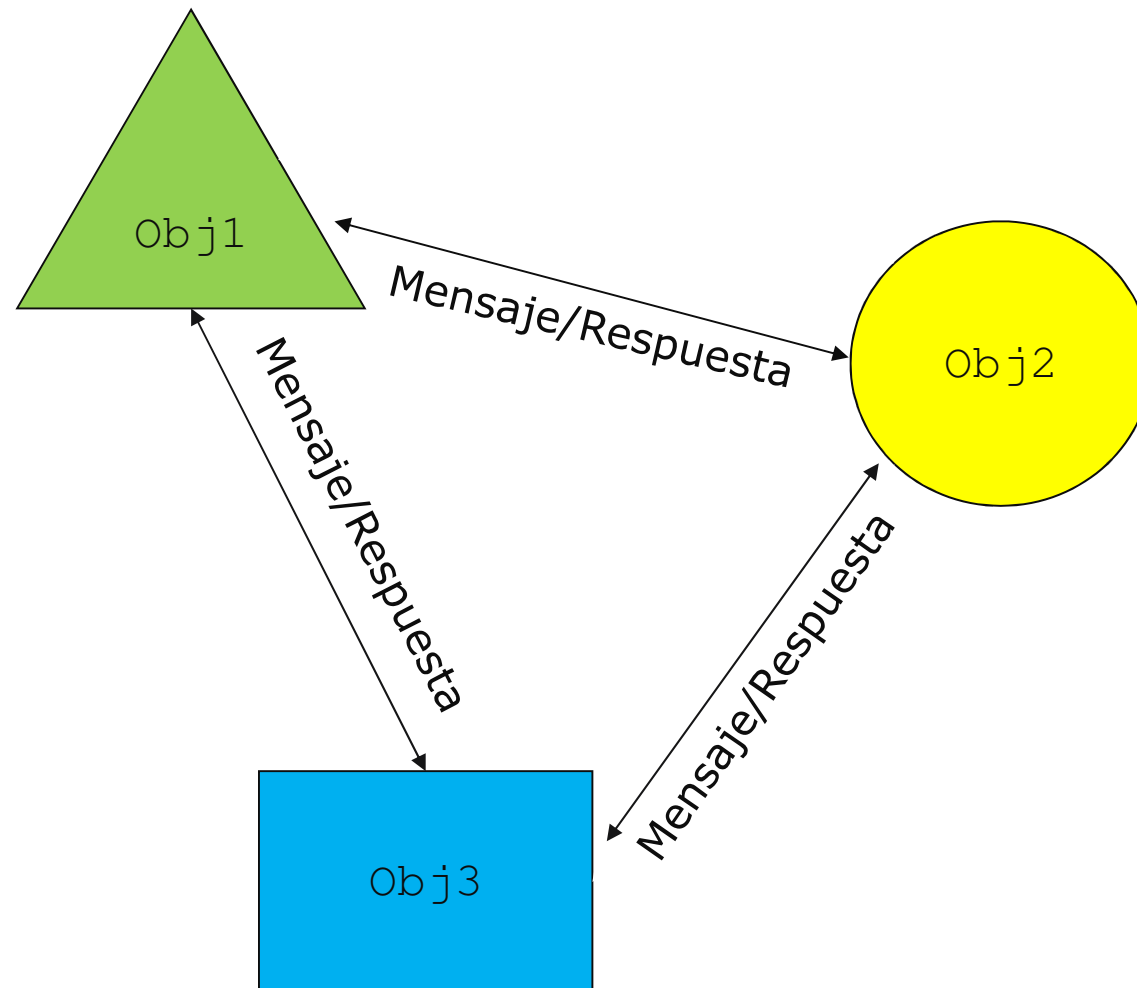
¿Qué es Programación Orientada a Objetos?

- La Programación Orientada a Objetos (POO) es un modelo de programación que utiliza objetos, unidos mediante mensajes, para la solución de problemas.

- El elemento principal es el objeto
- Los datos y los procesos forman una entidad única



¿Qué es Programación Orientada a Objetos?

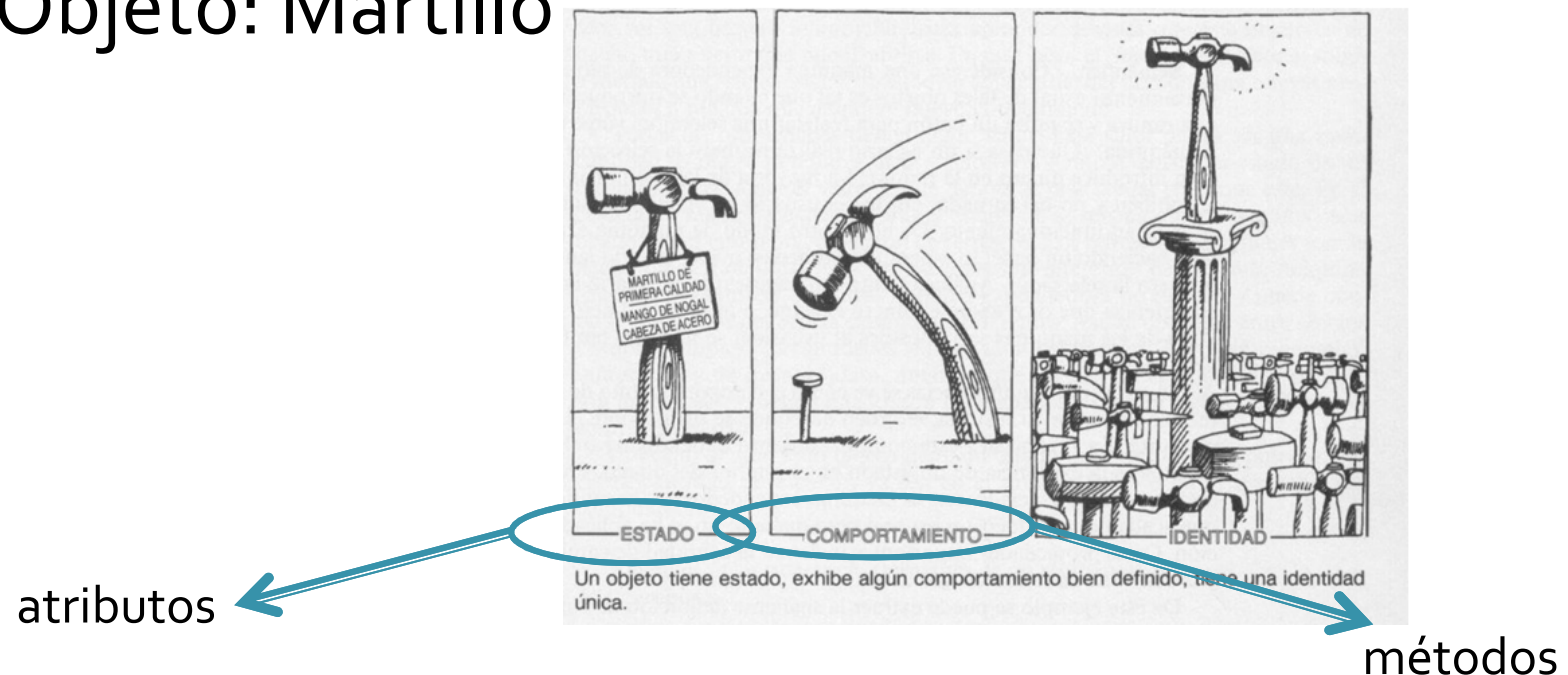


¿Qué es Programación Orientada a Objetos?

- Los objetos son entidades que combinan estado (atributo), comportamiento (método) e identidad:
 - El estado está compuesto de datos, será uno o varios atributos a los que se habrán asignado unos valores concretos (datos). Cambia en el transcurso del tiempo.
 - El comportamiento está definido por los procedimientos o métodos con que puede operar dicho objeto, es decir, qué operaciones se pueden realizar con él.
 - La identidad es una propiedad de un objeto que lo diferencia del resto, dicho con otras palabras, es su identificador.

¿Qué es Programación Orientada a Objetos?

- Los objetos en programación se usan para modelar entidades del mundo real.
- Ejemplos: Vehículo, Casa, Persona.
- Objeto: Martillo



¿Qué es Programación Orientada a Objetos?

■ Objeto: Bicicleta



Bicicleta modelada como un objeto:

Atributos:

- Velocidad
- Cadencia
- Marcha

Métodos:

- Cambiar marcha
- Frenar
- Cambiar cadencia

¿Qué es Programación Orientada a Objetos?

- Objeto: Persona
 - ¿atributos?
 - ¿métodos?



¿Qué es Programación Orientada a Objetos?

- Objeto: Celular
 - ¿atributos?
 - ¿métodos?



¿Qué es Programación Orientada a Objetos?

- Objeto: Auto
 - Cada objeto tiene identidad única, incluso si su estado es idéntico a otro objeto.

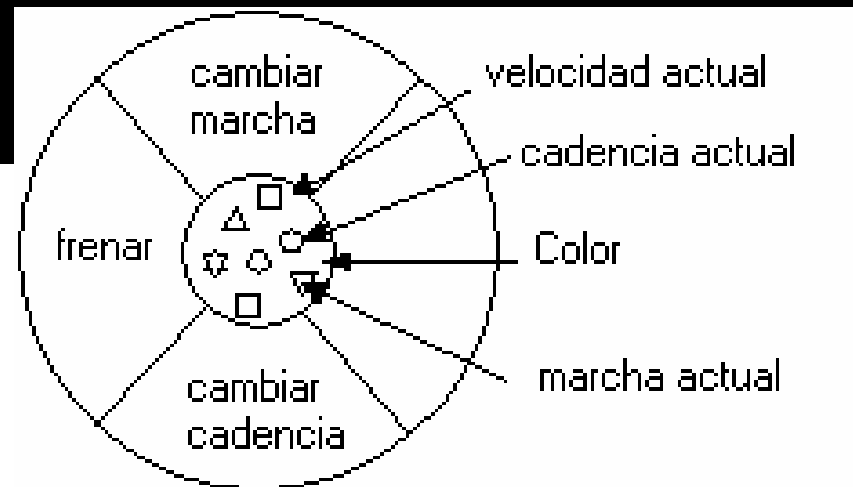


¿Qué es una Clase?

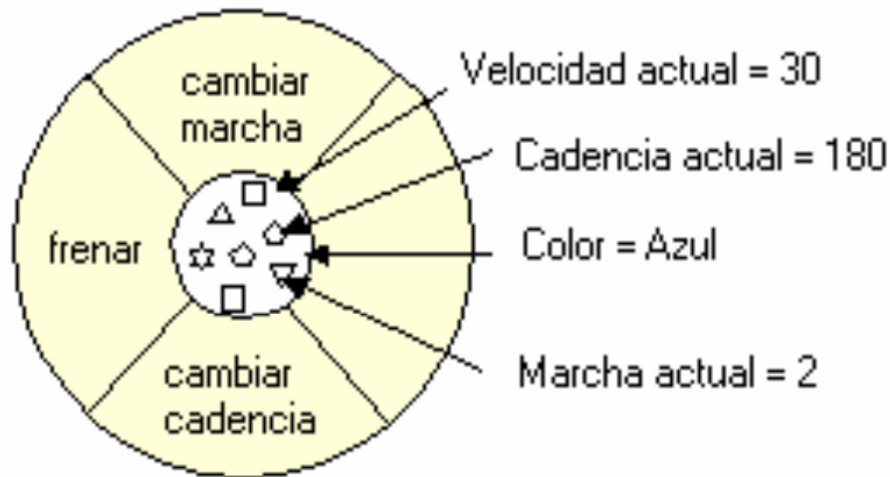
- Normalmente en el mundo real existen varios objetos de un mismo tipo o de una misma clase.
- Una clase es una plantilla que define las variables y los métodos que son comunes para todos los objetos de un cierto tipo.



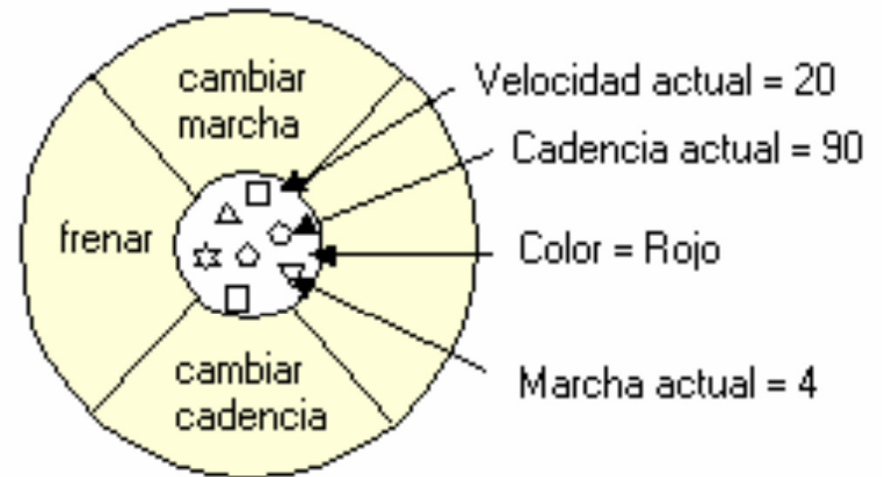
Clase Bicicleta



Objetos o instancias de la Clase Bicicleta



MiBicicleta



TuBicicleta

¿Qué es una Clase?

Clase Persona



Objetos o instancias de la Clase Persona



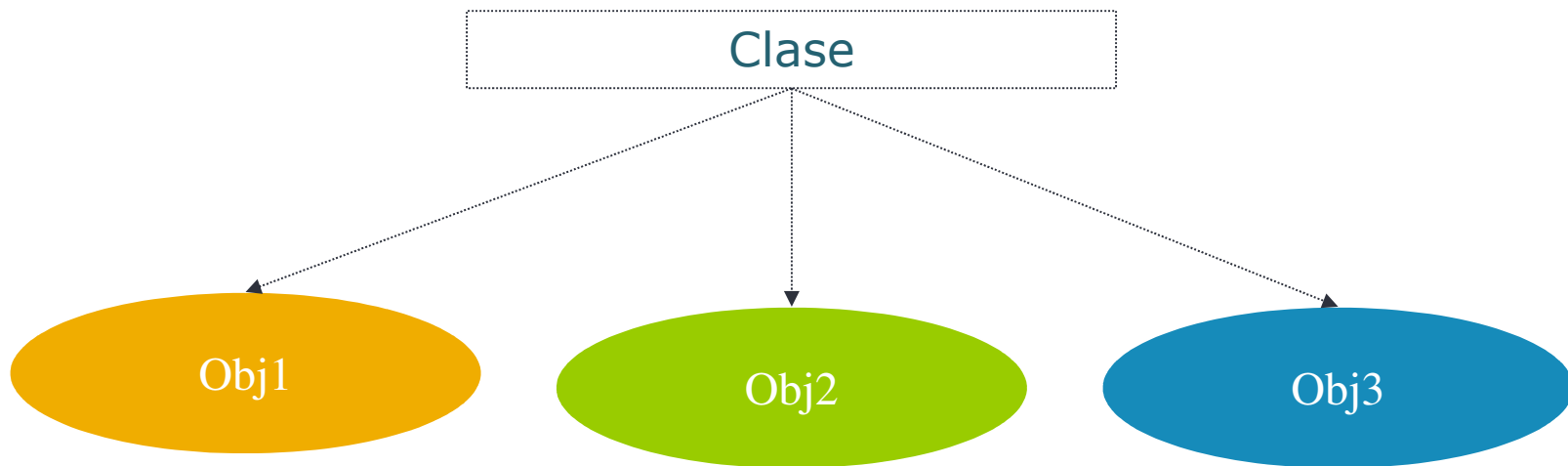
Pedro

Juan

Ana

¿Qué es una Clase?

Relación Clase-Objeto



A partir de una clase pueden generarse múltiples objetos.

Objetos vs Clases

Objetos

- Representan 'cosas' del mundo real, o de un dominio de problemas (ejemplo: "el auto rojo estacionado en el subterráneo")

Clases

- Representan a todos los objetos de un mismo tipo en conjunto (ejemplo : "auto")

Clase: Auto

Color

CantPuertas

TipoDeCombustible

Atributos

Arrancar

Frenar

Acelerar

Métodos

Objeto: Ambulancia



•Color: Celeste

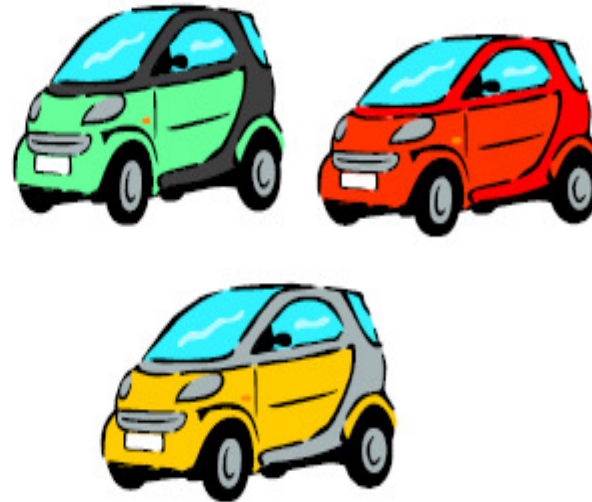
•CantPuertas: 4

•TipoDeCombustible: Diesel

Ejemplos

Clase: Auto

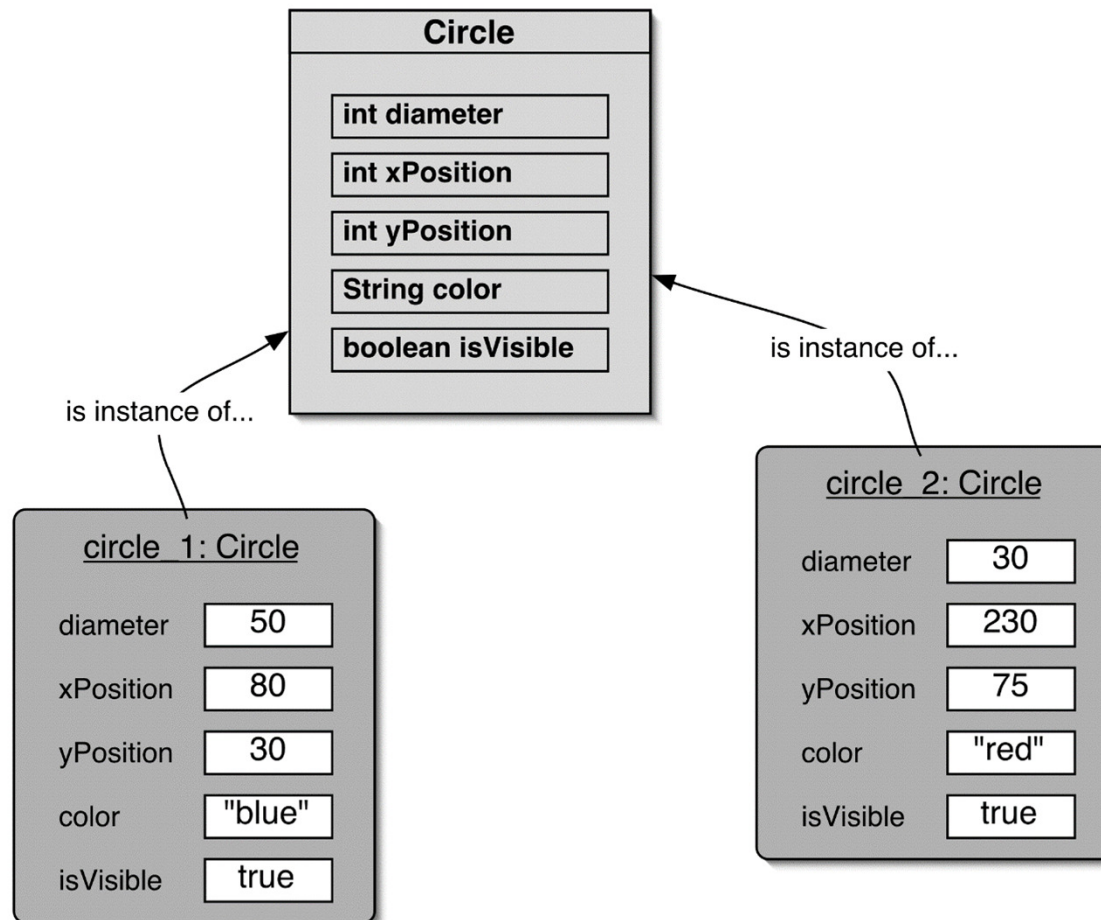
Objetos: Auto Verde
Auto Amarillo
Auto Rojo



¿A qué clase pertenece el Objeto Auto Verde?

Objetos vs Clases

- Dos objetos de la clase "Circle"



Métodos y Parámetros

- Los objetos tienen operaciones que pueden ser invocados (Java llama a esas operaciones métodos)
- Los métodos pueden tener parámetros para recibir información adicional para su ejecución

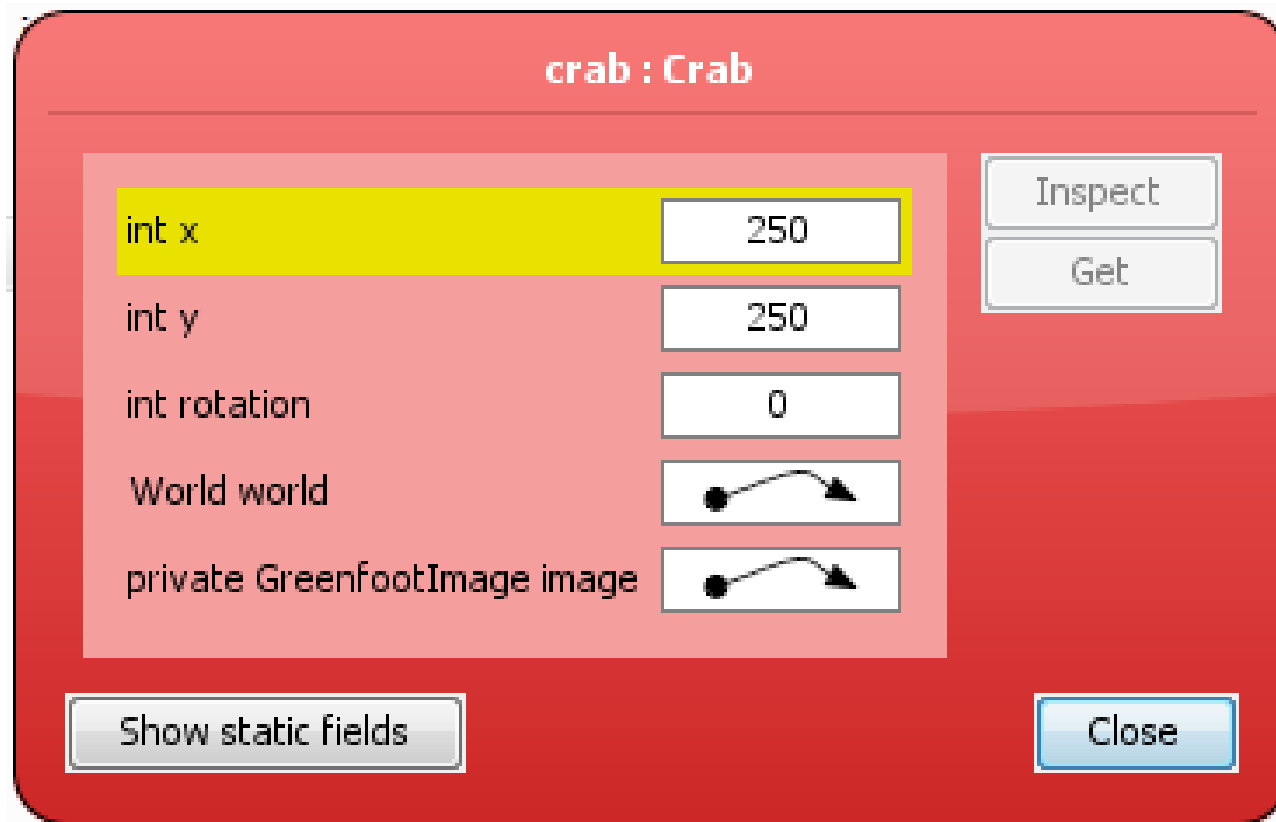
Métodos y Parámetros

- Valores de Retorno
 - Los métodos pueden retornar resultados al terminar su ejecución.

Otras observaciones

- Se pueden crear muchas instancias de una sola clase
- Un objeto tiene atributos: valores almacenados en campos.
- La clase define que campos tienen un objeto, pero cada objeto almacena su propio conjunto de valores (el estado del objeto)

Estado



Código Fuente

- Cada clase tiene código fuente (código Java) que define los detalles de esa clase (atributos y métodos).



Conceptos Fundamentales de la POO

Abstracción

Encapsulamiento

Herencia

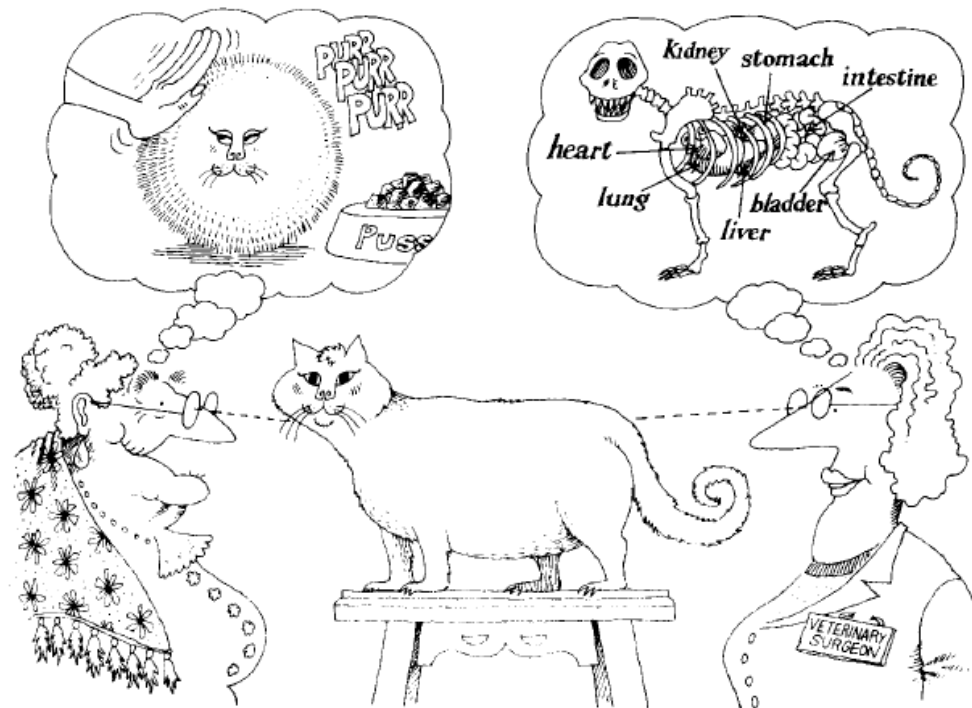
Polimorfismo

Abstracción

- Ignorancia Selectiva
 - La abstracción nos ayuda a trabajar con cosas complejas.
 - Se enfoca en lo importante.
 - Ignora lo que no es importante (simplifica).
- Una clase es una abstracción en la que:
 - Se enfatizan las características relevantes.
 - Se suprimen otras características.
- Una clase debe capturar una y sólo una abstracción clave.

Abstracción

- La abstracción enfoca las características esenciales de algún objeto, relativa a la perspectiva del observador.

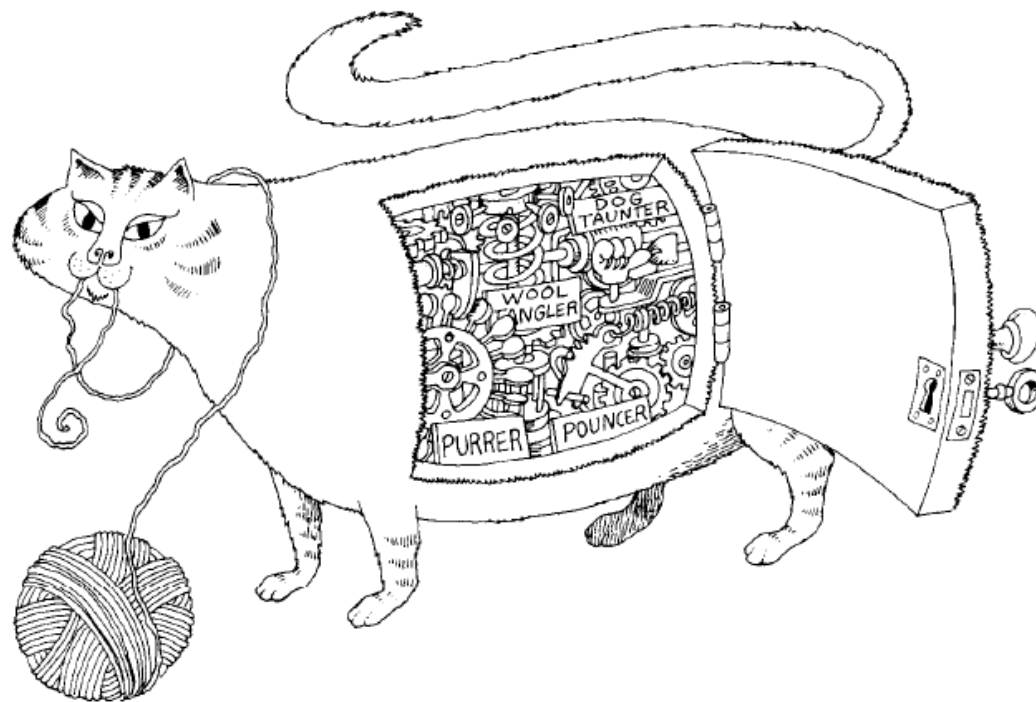


Encapsulamiento

- Principio que establece que los atributos propios de un objeto no deben ser visibles desde otros objetos.
- Deben ser declarados como privados.
- Permite abstraer al resto del mundo de la complejidad de la implementación interna.
- Permite exponer el estado del objeto sólo a través del comportamiento que le hayamos definido mediante miembros públicos.

Encapsulamiento

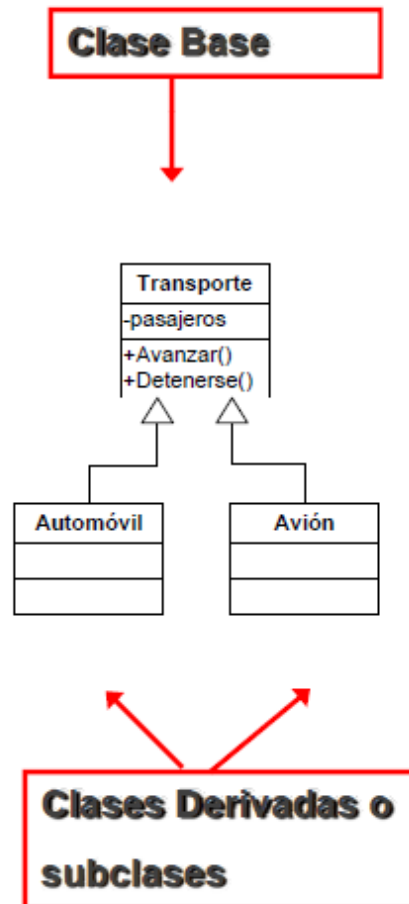
- El encapsulamiento oculta detalles de implementación de un objeto.



Herencia

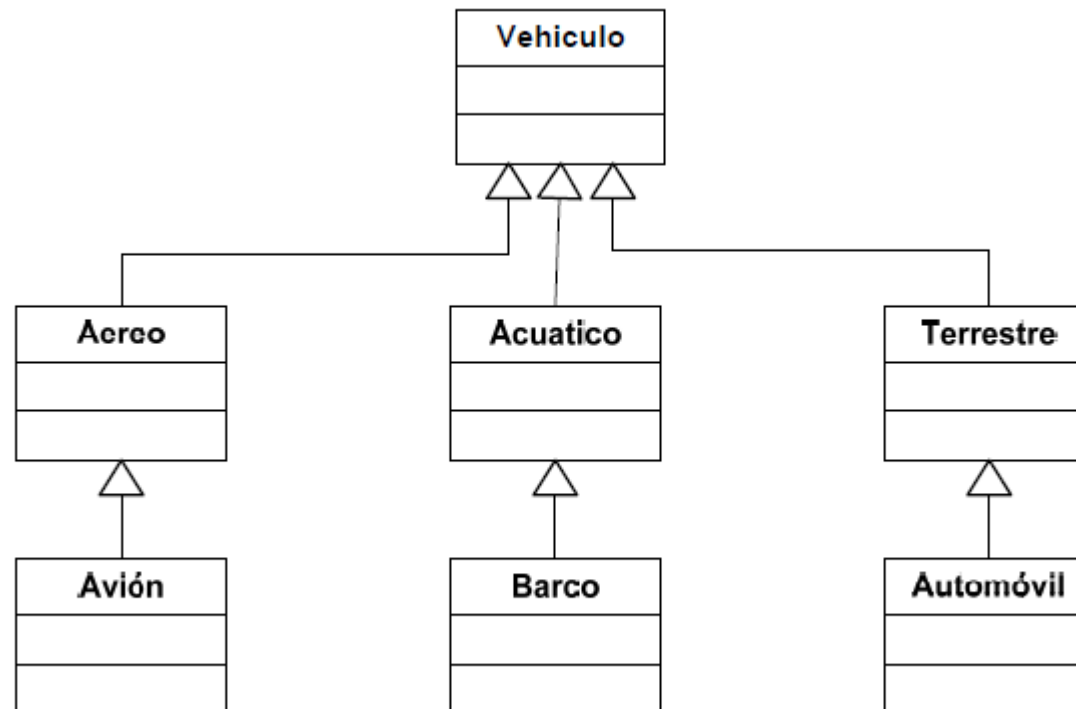
- Es una relación entre clases en la cual una clase comparte la estructura y comportamiento definido en otra clase.
- Cada clase que hereda de otra posee:
 - Los atributos de la clase base además de los propios.
 - Soporta todos o algunos de los métodos de la clase Base (puede sobrescribir métodos).
 - El árbol de herencias o jerarquía de clases puede ser tan extenso como necesitemos.
 - Una subclase hereda de una clase base.

Herencia



Herencia

- ¿De qué clase heredaría la clase Bicicleta?

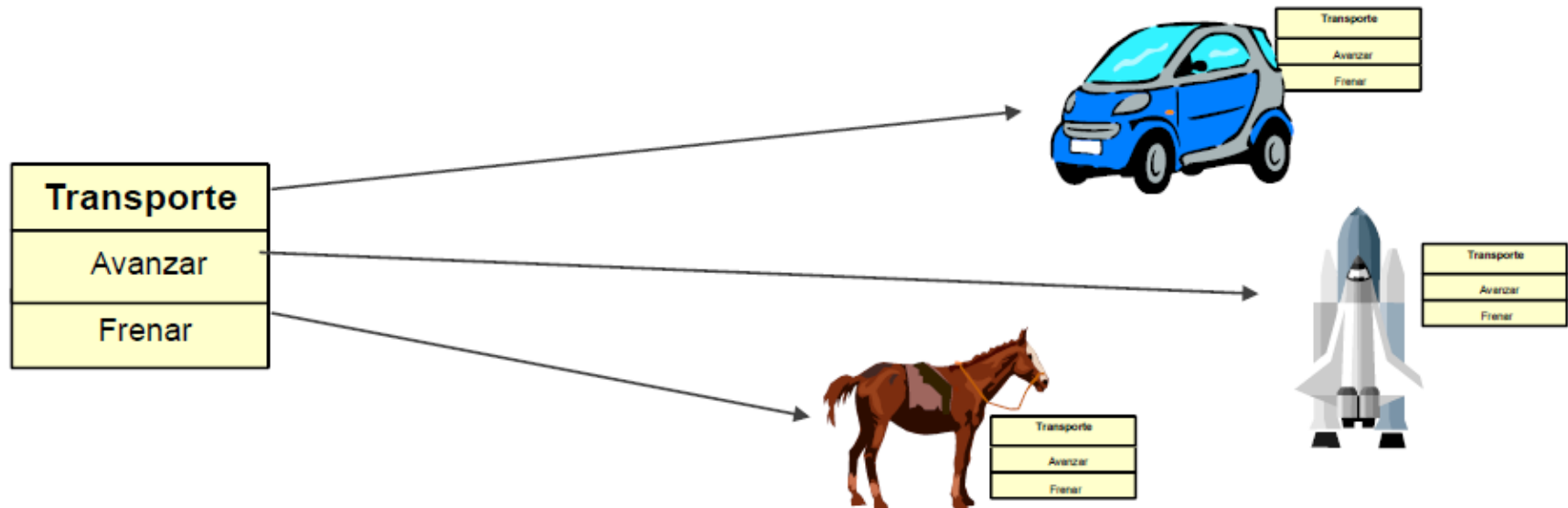


Polimorfismo

- La propiedad de construir varios métodos con el mismo nombre, pero con relación a la clase a la que pertenece cada uno, con comportamientos diferentes.
- Esto conlleva la habilidad de enviar un mismo mensaje a objetos de clases diferentes.
- Estos objetos recibirían el mismo mensaje global pero responderían a él de formas diferentes.
- Ejemplo: mensaje "+":
 - Objeto ENTERO => suma
 - Objeto STRING => concatenación

Polimorfismo

■ Ejemplo: Transporte



¿ Es igual el avanzar de Auto y Caballo?

IDE BlueJ

- BlueJ es un entorno integrado de desarrollo muy sencillo, pensado para aprender a programar en Java.
- Las principales ventajas de BlueJ son:
 - Es gratuito
 - Es fácil de usar
 - Es ligero (no requiere una máquina muy potente)



IDE BlueJ

- Ejecución
 - Para arrancar, haga doble "click" sobre el archivo bluej.exe.
- Funcionamiento
 - BlueJ organiza los programas en lo que denomina "proyectos", de forma que cada proyecto está en un directorio diferente que contiene todos los archivos de un programa.
- NOTA: lo que BlueJ llama "proyecto" es lo mismo que un "directorio" o una "carpeta" o un "folder".

IDE BlueJ

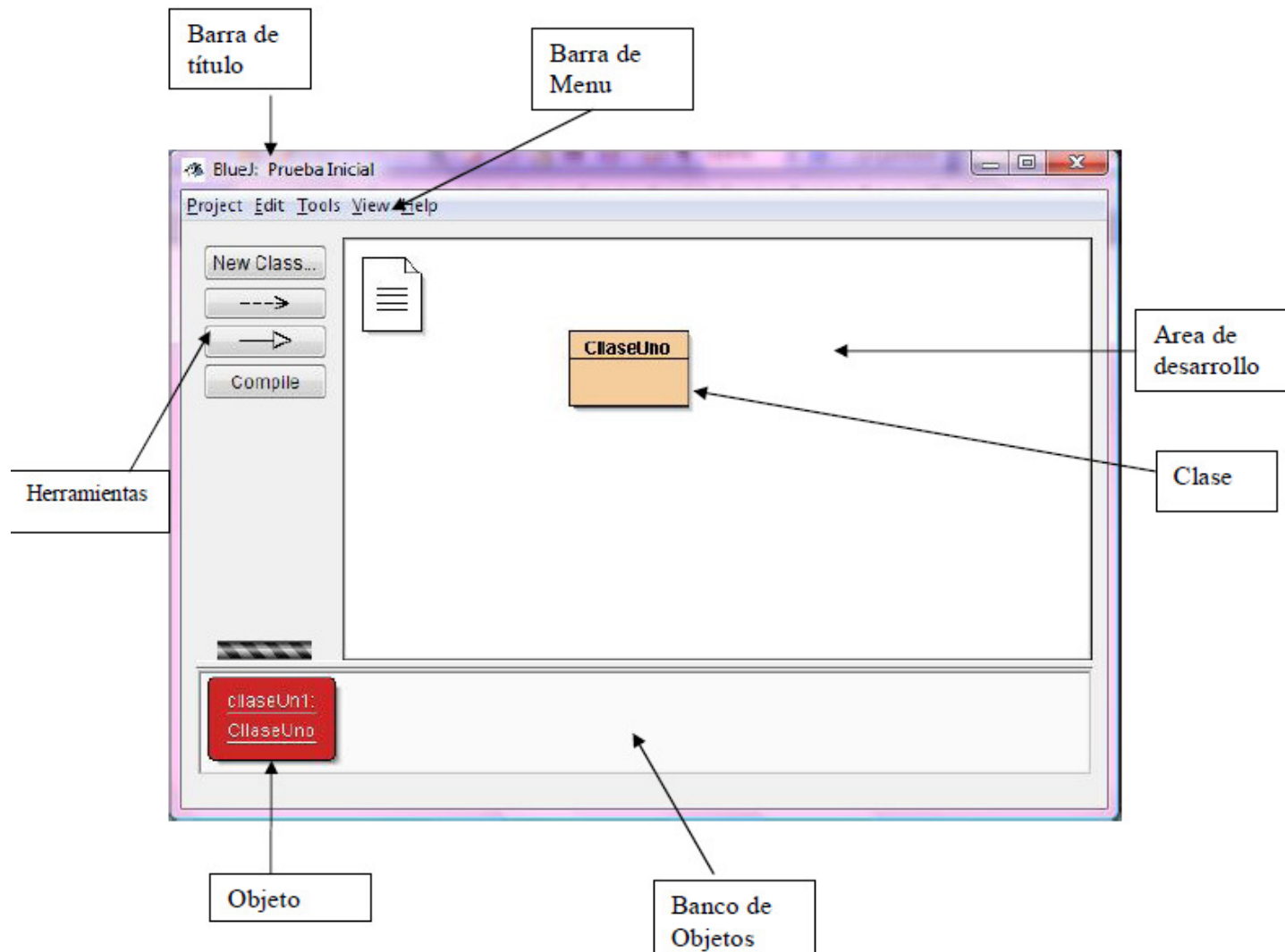


Diagrama de Clases

- Un diagrama de Clases representa las clases que serán utilizadas dentro del sistema y las relaciones que existen entre ellas.
- El propósito de este diagrama es el de representar los objetos fundamentales del sistema, es decir, los que percibe el usuario

Diagrama de Clases

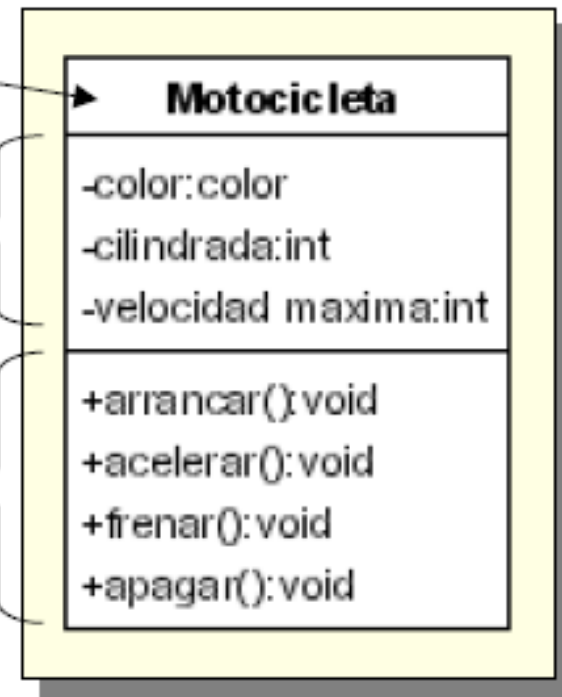
- Cada clase se representa en un rectángulo con tres compartimientos:

- Nombre de la clase

- Atributos de la clase

- Operaciones de la clase

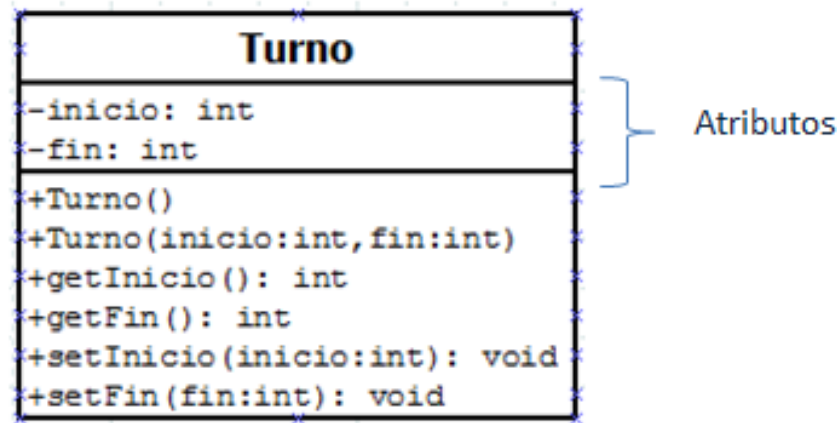
↓
Métodos



Atributos

- Tipo: tipo de dato, puede llegar a depender del lenguaje de programación a utilizar.
- Visibilidad: está relacionado con el encapsulamiento (privado o público).
- Ejemplo:

Atributos: inicio, fin.
Tipo: int (inicio)
 int (fin)
Visibilidad: privado (-)



Niveles de Encapsulamiento

- (-) Privado: es el más fuerte. Esta parte es totalmente invisible desde fuera de la clase.
- (+) Los atributos/operaciones públicos son visibles a otras clases (cuando se trata de atributos se está transgrediendo el principio de encapsulamiento).

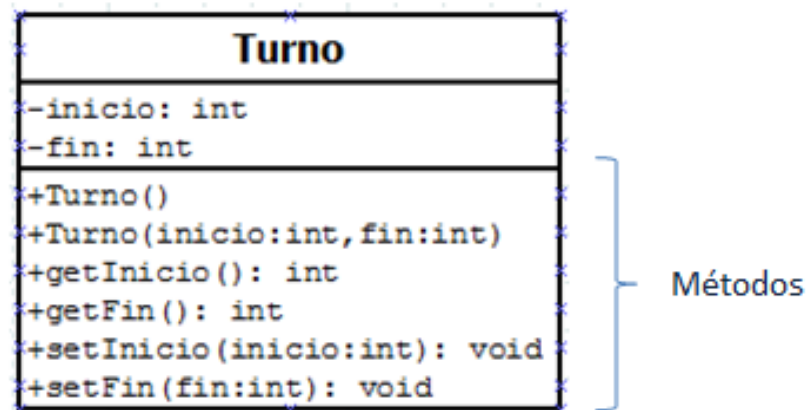
Métodos

- Una operación es un servicio que una instancia de la clase puede realizar.
 - Tipo devuelto: tipo de dato, puede llegar a depender del lenguaje de programación a utilizar.
 - Parámetros: parámetros de entrada al método.
 - Visibilidad: está relacionado con el encapsulamiento (privado o público).

Métodos: Turno, getInicio, getFin,
setInicio, setFin

Tipo devuelto: int (getInicio)
int (getFin)

Visibilidad: público (+)



Fin del módulo...