Programación de sistemas

Orientación a Objetos en Java

I. Programación ***Basada*** en objetos

II. Programación ***orientada*** a objetos

Ingeniería Telemática

M. Carmen Fernández Panadero

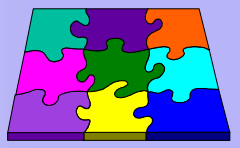
<mcfp@it.uc3m.es>

Escenario IV:
Declarar e implementar una clase

- Ahora que ya sabes interpretar código e implementar tus propios métodos te encargan el diseño de una clase completa para crear un nuevo tipo de datos con sus características y comportamiento.

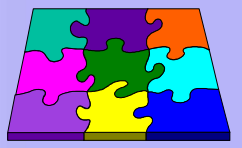
- Objetivo:
  - Ser capaz de **declarar una clase** con un conjunto de características (**atributos**) y comportamientos (**métodos**)
  - Ser capaz de **crear objetos** de una clase dada y modificar o restringir el acceso a su estado y su comportamiento
- Plan de trabajo:
  - Memorizar la **nomenclatura** básica de la programación orientada a objetos
  - Practicar el **modelado** de objetos con ejemplos sencillos para distinguir entre una clase, un objeto, su estado y su comportamiento
  - Repasar la **sintaxis** java para declarar **clases atributos**, **constructores** y **métodos**
  - Recordar el mecanismo y la sintaxis para **paso de mensajes** entre objetos

Contenidos



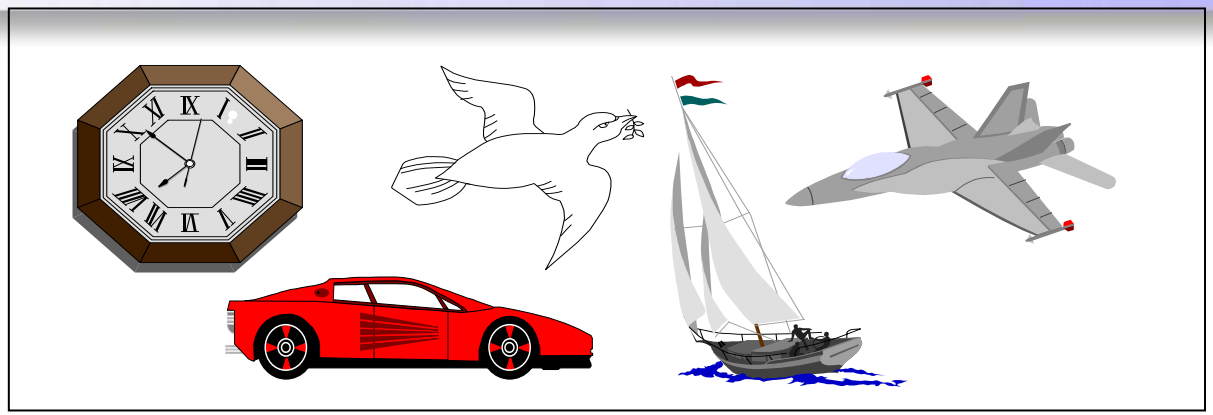
- Clases y objetos
- Encapsulación de objetos
  - Abstracción funcional
  - Abstracción de datos
- Miembros de una clase (atributos y métodos)
- Paso de mensajes
- Constructores
- Sobrecarga

Objetivos



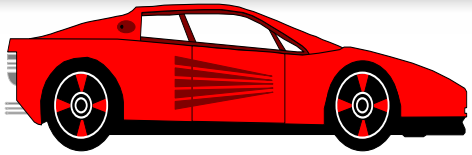
- Definir los ***conceptos básicos*** de la programación ***basada*** en objetos
  - Clases, objetos
  - Miembros (variables, métodos)
  - Abstracción y ocultación de información
- Describir ***relación*** entre objeto y clase
- ***Crear*** un objeto sencillo y ***modelar***
  - sus atributos (por medio de variables)
  - su comportamiento (por medio de métodos)

¿Qué es un objeto?



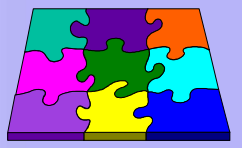
- Los **objetos** son representaciones (simples/complejas) (reales/imaginarias) de cosas: reloj, avión, empleado, etc.
- No todo puede ser considerado como un objeto, algunas cosas son simplemente características o **atributos** de los objetos: color, velocidad, etc.

¿Qué es un objeto?



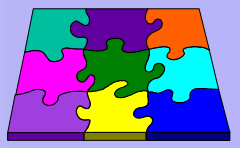
- ***Abstracción funcional***
  - Hay cosas que sabemos que los coches hacen pero no cómo lo hacen:
    - avanzar
    - parar
    - girar a la dcha
    - girar a la izda
- ***Abstracción de datos***
  - Un coche tiene además ciertos atributos:
    - color
    - velocidad
    - tamaño
    - etc..
  - La forma en que se definen los atributos no tiene importancia para el diseño

¿Qué es un objeto?



- Es una forma de agrupar un conjunto de datos (**estado**) y de funcionalidad (**comportamiento**) en un mismo bloque de código que luego puede ser referenciado desde otras partes de un programa
- La **clase** a la que pertenece el objeto puede considerarse como un nuevo **tipo de datos**.

Encapsulación de objetos



- ***Encapsulación:*** describe la vinculación de un **comportamiento** y un **estado** a un objeto en particular.
- ***Ocultación de información***: Permite definir qué partes del objeto son visibles (el interfaz público) que partes son ocultas (privadas)

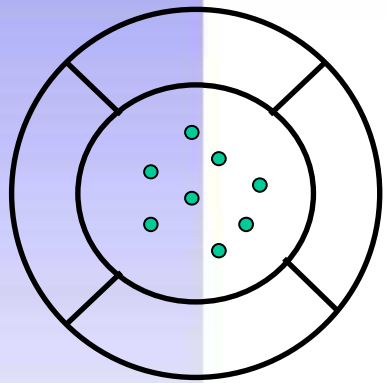
*• La llave de contacto es un **interfaz público** del mecanimo de arranque de un coche*

*• La implementación de cómo arranca realmente es **privada** y sobre ella sólo puede actuar la llave de contacto*

ventajas

El objeto puede cambiar y su interfaz público ser compatible con el original esto facilita reutilización de código

Encapsulación de objetos



MIEMBROS DE UNA CLASE

Los objetos encapsulan atributos permitiendo acceso a ellas únicamente a través de los métodos

- **Atributos (Variables)**: Contenedores de valores
- **Métodos**: Contenedores de funciones

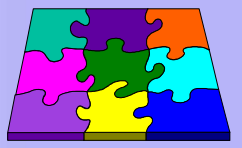
Un objeto tiene

- **Estado**: representado por el contenido de sus atributos
- **Comportamiento**: definido por sus métodos

Normalmente:

- Los métodos son públicos
- Los atributos son privados
- Puede haber métodos privados
- Es peligroso tener atributos públicos

Definición de objetos



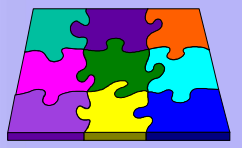
Miembros públicos

- los miembros públicos (describen **qué** pueden hacer los objetos de esa clase)
  - Qué puede hacer el objeto (métodos)
  - Qué es el objeto (su abstracción)

Miembros privados

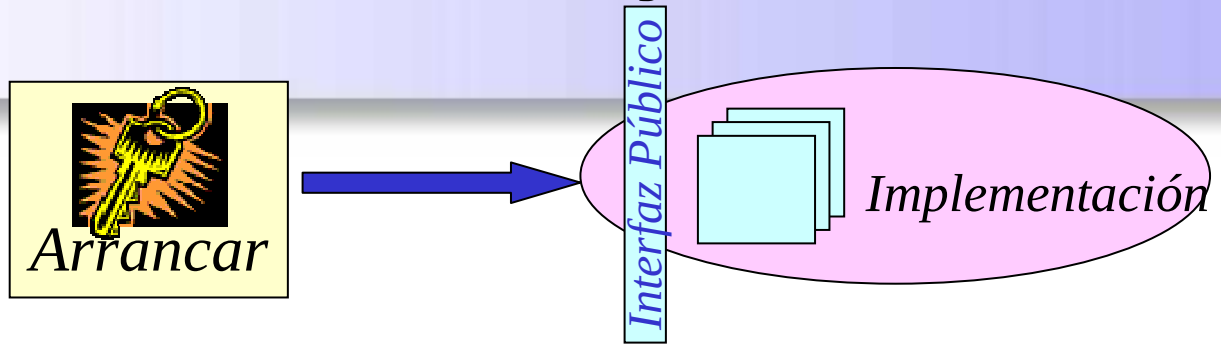
- describen la implementación de **cómo** lo hace.
  - Ejemplo el objeto contacto interacciona con el circuito electrico del vehículo, este con el motor, etc.
  - ***En sistemas orientados a objetos puros todo el estado es privado y sólo se puede cambiar a través del interfaz público***.
  - Ej: el método público frenar puede cambiar el valor del atributo privado velocidad.

Interacciones entre objetos



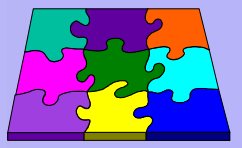
- El ***modelado de objetos*** modela:
  - los objetos y
  - sus interrelaciones
- Para realizar su tarea el objeto puede ***delegar*** trabajos en otro que puede ser parte de él mismo o de cualquier otro objeto del sistema.
- Los objetos interaccionan entre sí enviándose ***mensajes***

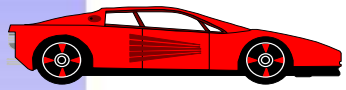
Paso de Mensajes



- Un objeto envía un ***mensaje*** a otro
  - Esto lo hace mediante una **llamada** a sus atributos o métodos
- Los *mensajes* son tratados por la **interfaz pública** del objeto que los recibe
  - Eso quiere decir que sólo podemos hacer llamadas a aquellos atributos o métodos de otro objeto que sean **públicos o accesibles** desde el objeto que hace la llamada
- El objeto receptor reaccionará
  - **Cambiando su estado** (es decir modificando sus atributos)
  - **Enviando otros *mensajes*** (es decir llamando a otros atributos o métodos del mismo objeto (públicos o privados) o de otros objetos (públicos o accesibles desde ese objeto))

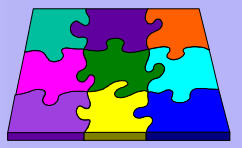
Clasificación de objetos






- **Clase**: Conjunto de objetos con estados y comportamientos similares
  - Podemos referirnos a la clase “Coche” (cualquier instancia de la clasificación coche)
- “Mi coche” es un **objeto**, es decir una **instancia** particular de la clase coche
- La clasificación depende del problema a resolver

Objetos vs. Clases



Una **clase** es una entidad abstracta

- Es un tipo de clasificación de datos
- Define el comportamiento y atributos de un grupo de estructura y comportamiento similar

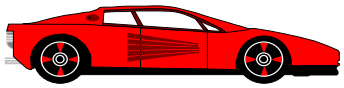
Clase coche
Métodos: arrancar, avanzar, parar, ...
Atributos: color , velocidad, etc.

→ Nombre de la clase
→ Métodos (funciones)
→ Atributos (datos)

Un **objeto** es una instancia de una clase

- Un objeto se distingue de otros miembros de la clase por sus atributos

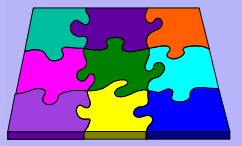
Objeto Ferrari
Perteneciente a la clase coche



Nombre: Ferrari
Métodos: arrancar, avanzar, parar, ...
Atributos: color = "rojo";
velocidad 300Km/h

- Una **clase** se declara, un **objeto** además se crea

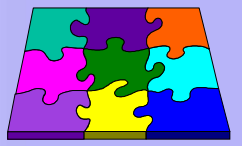
Constructores
Notas a recordar



- Cuando se crea un objeto sus miembros se ***inicializan*** con un método constructor
- Los constructores:
  - llevan el ***mismo nombre*** que la clase
  - ***No*** tienen ***tipo*** de resultado (ni siquiera void)
- Conviene que haya al menos 1 constructor
- Pueden existir varios que se distinguirán por los parámetros que aceptan (***sobrecarga***)
- Si no existen se crea un ***constructor por defecto*** sin parámetros que inicializa las variables a su valor por defecto.
- Si la clase tiene algún constructor, el constructor por defecto deja de existir. En ese caso, si queremos que haya un constructor sin parámetros tendremos que declararlo explícitamente.

Sobrecarga (Overloading)

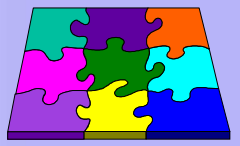
¿Qué es?



- Podemos definir una clase con dos métodos con el ***mismo nombre*** si los ***argumentos son distintos***.
- Se utiliza mucho para los constructores.
- Sabemos cual de los dos métodos tenemos que ejecutar por los parámetros que le pasamos cuando le llamamos.
- En este caso ***no hay ocultación*** de la información, se puede acceder a los dos métodos.

Sobrecarga (Overloading)

¿Para qué sirve?



Aula
- nombre
- descripcion
- localizacion
- decirNombre()
- decirDescripcion()
- decirDescripcion(String mobiliario)
- decirLocalizacion()

*Son dos métodos distintos porque aunque tengan el **mismo nombre** tienen **distintos argumentos***

*Tienen **distinta funcionalidad** como ocurre en el caso que se muestra en el ejemplo*

describe el aula en general

describe el mueble que se encuentra en el aula y que le pasamos como parámetro