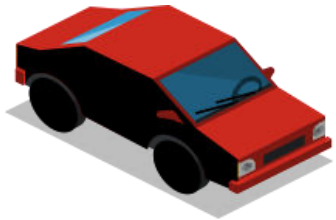
IMPLEMENTACIÓN DE LA ENCAPSULACIÓN

OBJETOS: ESTRUCTURA Y COMPORTAMIENTO

- En general, todos los objetos tienen una estructura (como están conformado) y un comportamiento (realizan una serie de operaciones).

ESTRUCTURA

- Plástico
- 4 ruedas
- 1 volante
- …

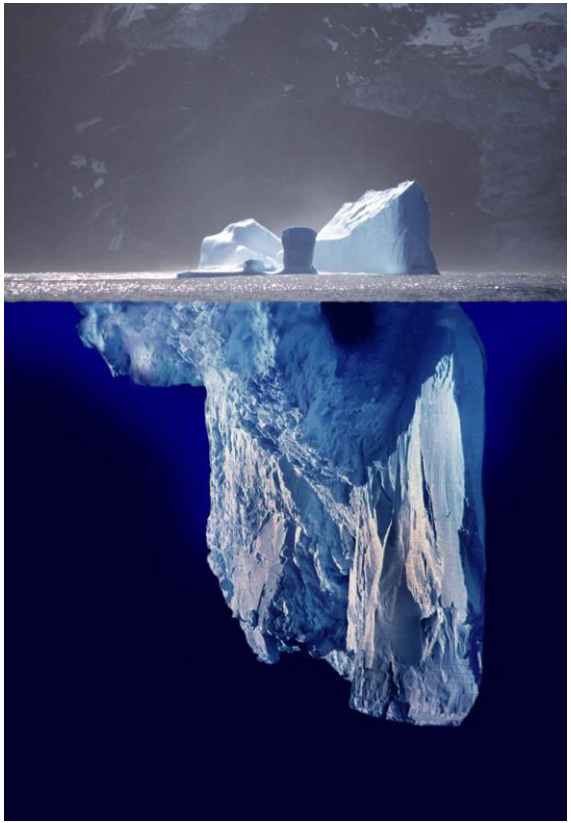


COMPORTAMIENTO

- Mover adelante
- Mover atrás
- …

ENCAPSULACIÓN

- Los objetos conocen solamente su estructura, no la de los demás.



ENCAPSULACIÓN

- El trato entre objetos se realiza a través de los métodos.
- Normalmente, los atributos de un objeto se deben consultar o editar a través de métodos.

DEFINICIÓN E IMPLEMENTACIÓN DE UNA CLASE

```
<modificador> class NombreDeLaClase {

        //propiedades
        int propiedad1;
        String propiedad2;
        float propiedad3;
        //…

        //metodos
        void metodo1() {
                //…
        }

        //…
}
```

MODIFICADORES DE ACCESO

- Nos permiten indicar quien puede hacer uso de una clase, o de sus atributos y métodos.
- *public*: cualquiera
- *private*: solo la propia clase
- *protected*: la propia clase y sus derivados
- Por defecto: las clases cercanas (que estén en el mismo paquete).

BEST PRACTICES

- La mayoría de las clases que se crean son públicas.
- Cada fichero .java tendrá solamente una clase pública, con el mismo nombre del fichero.

```
public class MiClase {

        //propiedades
        //…

        //metodos
        //…
}
```

MiClase.java

BEST PRACTICES (II)

- La mayoría de los atributos de una clase serán privados.
- Solamente algunas constantes, o casos muy particulares, tendrán otra modificador de acceso.

```
public class MiClase {

        //propiedades
        private int numero;
        private String nombre;

        //metodos
        //…
}
```

MiClase.java

BEST PRACTICES (III)

- Si una clase tiene atributos, seguramente tenga métodos públicos.
- Los métodos privados son interesantes para cálculos auxiliares o parciales (solo se pueden invocar desde la propia clase).

```
public class MiClase {

        //propiedades
        private int numero;
        private String nombre;

        //metodos
        public int getNumero() { … }
}
```

MiClase.java

TIPOS DE CLASES

- Java solo tiene “una forma” de crear clases, a través de **`class`**.

Podemos diferenciar las clases según su cometido:

- Modelo
- Servicios
- Auxiliares
- *Main*
- Test
- …

TIPOS DE CLASES

- **Modelo**: representan objetos o hechos de la naturaleza: un coche, un asiento contable, los datos meteorológicos de un día. Suelen tener atributos, *getters y setters*, *equals*, *hashCode*, *toString*, …
- **Servicios**: implementan la lógica de negocio. Suelen tener algunos atributos, pero sobre todo métodos públicos y privados.

TIPOS DE CLASES

- **Auxiliares**: sirven para realizar operaciones auxiliares de cálculo o transformación de datos. Mayoritariamente, sus métodos son estáticos.
- ***Main***: son el punto de entrada de la aplicación. La mayoría de las ocasiones, solo tienen este método, y si tienen más, suelen ser estáticos.

TIPOS DE CLASES

- **Test**: clases orientadas a realizar pruebas de nuestra aplicación. En Java, suelen ser test unitarios con JUnit.