



CLASES Y OBJETOS

SEGUNDO CICLO
CARRERA DE SOFTWARE

CLASE

- En su forma más simple, una clase se define por la palabra reservada
- class seguida del nombre de la clase. El nombre de la clase debe empezar
- por mayúscula. Si el nombre es compuesto, entonces cada palabra debe
- empezar por mayúscula. Circulo, Rectangulo, Triangulo y
- FiguraGeometrica son nombres válidos de clases
- Por ejemplo, la clase Circulo se define con tres atributos: el radio y las
- coordenadas x, y que definen la posición del centro del círculo

```
/* Esta clase define los atributos de un círculo */  
  
public class Circulo {  
  
    int x;  
    int y;  
    int radio;  
  
}
```

OBJETOS

Una vez que se ha declarado una clase, se pueden crear objetos a partir de ella. A la creación de un objeto se le denomina instanciación. Es por esto que se dice que un objeto es una instancia de una clase y el término instancia y objeto se utilizan indistintamente.

Para crear objetos, basta con declarar una variable de alguno de los tipos de figuras geométricas:

```
Circulo circulo1;  
Circulo circulo2;
```

Para crear el objeto y asignar un espacio de memoria es necesario realizar la instanciación con el operador `new`.

```
circulo1 = new Circulo();  
circulo2 = new Circulo();
```

Después de crear los objetos, `circulo1` y `circulo2` almacenan los valores predeterminados de la clase `Circulo`. A partir de este momento los objetos ya pueden ser referenciados por su nombre. Los nombres `circulo1` y `circulo2` son las referencias válidas para utilizar ambos objetos.

ATRIBUTOS

- La información de un objeto se almacena en atributos. Los atributos pueden
- ser de tipos primitivos de Java o de tipo objeto. Por ejemplo, para el
- catálogo de vehículos de una empresa de alquiler, es necesario conocer la
- matrícula del coche, su marca, modelo, color, la tarifa del alquiler y su
- disponibilidad.

```
public class Vehiculo {  
  
    String matricula;  
    String marca;  
    String modelo;  
    String color;  
    double tarifa;  
    boolean disponible;  
  
}
```

En este ejemplo, los atributos `matricula`, `marca`, `modelo` y `color` son cadenas de caracteres, `tarifa` es un número real y `disponible` es un valor lógico.

Métodos y constructores

Es necesario definir los métodos que determinan su comportamiento. Toda clase debe definir un método especial denominado constructor para instanciar los objetos de la clase. Este método tiene el mismo nombre de la clase. Por ejemplo, para la clase Vehiculo, el identificador del método constructor es Vehiculo.

El método constructor se ejecuta cada vez que se instancia un objeto de la clase. Este método se utiliza para inicializar los atributos del objeto que se instancia.

Para diferenciar entre los atributos del objeto y los identificadores de los parámetros del método constructor, se utiliza la palabra `this`.

```
this.marca = marca;
```


EJEMPLO

```
public class Vehiculo {
    String matricula;
    String marca;
    String modelo;
    String color;
    double tarifa;
    boolean disponible;

    // el método constructor de la clase Vehiculo

    public Vehiculo(String matricula,
                    String marca,
                    String modelo,
                    String color,
                    double tarifa) {
        this.matricula = matricula;
        this.marca = marca;
        this.modelo = modelo;
        this.color = color;
        this.tarifa = tarifa;
        this.disponible = false;
    }
}
```

get, set

Para acceder a los atributos de los objetos de la clase Vehiculo se definen los métodos 'get' y 'set'. Los métodos 'get' se utilizan para consultar el estado de un objeto y los métodos 'set' para modificar su estado. En la clase Vehiculo es necesario definir un método 'get' para cada uno de sus atributos: getMatricula(), getMarca(), getModelo(), getColor(), getTarifa() y getDisponible(). Los métodos 'set' solo se definen para los atributos que pueden ser modificados después de que se ha creado el objeto.

En este caso es necesario definir setTarifa(double tarifa) y setDisponible(boolean disponible) para modificar la tarifa del alquiler del vehículo y su disponibilidad, respectivamente

Ejemplo

```
public class Vehiculo {
    String matricula;
    String marca;
    String modelo;
    String color;
    double tarifa;
    boolean disponible;

    // los métodos 'get' y 'set' de la clase Vehiculo

    public String getMatricula() {
        return this.matricula;
    }
    public String getMarca() {
        return this.marca;
    }
    public String getModelo() {
        return this.modelo;
    }
    public String getColor() {
        return this.color;
    }
    public double getTarifa() {
        return this.tarifa;
    }
    public boolean getDisponible() {
        return this.disponible;
    }
    public void setTarifa(double tarifa) {
        this.tarifa = tarifa;
    }
    public void setDisponible(boolean disponible) {
        this.disponible = disponible;
    }
}
```


Ejemplo: Modificación del valor de la tarifa a 90 euros

```
Vehiculo vehiculo1 = new Vehiculo("4050 ABJ",  
                                   "VW",  
                                   "GTI",  
                                   "Blanco",  
                                   100.0);  
  
System.out.println("Matricula      : " +  
                   vehiculo1.getMatricula());  
System.out.println("Tarifa         : " +  
                   vehiculo1.getTarifa());  
  
vehiculo1.setTarifa(90.0);  
  
System.out.println("Matricula      : " +  
                   vehiculo1.getMatricula());  
System.out.println("Tarifa         : " +  
                   vehiculo1.getTarifa());
```

Ciclo de vida de un producto

El ciclo de vida de un objeto empieza por su declaración, su instanciación, uso en un programa java y finalmente desaparece liberando la memoria asignada al objeto y la reutiliza

Atributos .- Tipos de atributos

- **Atributos de clase** .- Existen siempre , son independientes de que existan objetos instanciados, se declaran utilizando **static**
- **Atributos de Objeto**.- Existen durante el ciclo de vida de un objeto, se crea cuando se instancia el objeto y se puede utilizar mientras el objeto exista.
- **SINTAXIS**.- tipo de acceso tipo nombre [= valor inicial];

Atributos .- Tipos de accesos `private`, `protected` o `public`

- **SINTAXIS.**- tipo de acceso tipo nombre [= valor inicial];
- **`private`.**- Solo se puede acceder desde la propia clase que los define
- **`protected` .**- Se puede acceder desde la propia clase que los define y desde sus subclases
- **`public` .**- Se puede acceder libremente desde otras clases

Ejemplo de atributos private

```
public class Vehiculo {  
    private String matricula;  
    private String marca;  
    private String modelo;  
    private String color;  
    private double tarifa = 0.0;  
    private boolean disponible = false;  
}
```


Métodos

Los métodos son funciones que determinan el comportamiento de los objetos, un objeto se comporta de una u otra forma dependiendo de los métodos de la clase a la que pertenece.

- **Métodos de consulta.**- extraen información de los objetos
- **Métodos modificadores.**- sirven para modificar el valor de los atributos del objeto
- **Métodos de operación .**- definen el comportamiento de un objeto

Método consulta get.- M. modificador set

Los métodos 'get' se utilizan para extraer el valor de un atributo del objeto y los métodos 'set' para modificarlo. En la clase Vehiculo es necesario definir un método 'get' para cada uno de sus atributos: `getMatricula()`, `getMarca()`, `getModelo()`, `getColor()`, `getTarifa()` y `getDisponible()`. Los métodos 'set' solo se definen para los atributos que pueden ser modificados después de que se ha creado el objeto. En este caso es necesario definir los métodos `setTarifa(double tarifa)` y `setDisponible(boolean disponible)` para modificar la tarifa del alquiler del vehículo y su disponibilidad, respectivamente.

Un método 'get' se declara public

El valor de retorno es double, igual que el atributo tarifa

La lista de parámetros de un método 'get' queda vacía

```
public double getTarifa() {  
    return this.tarifa;  
}
```

Un método 'get' utiliza return para devolver el valor del atributo. En este caso el identificador del atributo es tarifa y se refiere a él como this.tarifa

Ejemplo .- Método get

El método `getAtributos()` es un caso particular de método 'get' que devuelve los valores concatenados de los atributos del objeto.

```
public class Vehiculo {  
    private String matricula;  
    private String marca;  
    private String modelo;  
    private String color;  
    private double tarifa = 0.0;  
    private boolean disponible = false;  
  
    public String getAtributos() {  
        return "Matrícula: " + this.matricula +  
            " Modelo: " + this.marca + " " + this.modelo +  
            " Color: " + this.color +  
            " Tarifa: " + this.tarifa +  
            " Disponible: " + this.disponible;  
    }  
}
```

Ejemplo .- Método set

Un método 'set' se declara `public` y devuelve `void`. La lista de parámetros de un método 'set' incluye el tipo y el valor a modificar. Es un método modificador. Por ejemplo, el método `setTarifa(double tarifa)` debe modificar el valor de la tarifa del alquiler almacenado en el objeto. El cuerpo de un método 'set' asigna al atributo del objeto el parámetro de la declaración.

Un método 'set' se declara `public`

El valor de retorno es `void`

La lista de parámetros de un método 'set' incluye el tipo y el nombre del parámetro

```
public void setTarifa(double tarifa) {  
    this.tarifa = tarifa;  
}
```

Un método 'set' modifica el valor de un atributo del objeto. En este caso el identificador del atributo es `tarifa` y se refiere a él como `this.tarifa` para asignarle el valor del parámetro

Ejemplo .- Método set

Un método 'set' se declara `public` y devuelve `void`. La lista de parámetros de un método 'set' incluye el tipo y el valor a modificar. Es un método modificador. Por ejemplo, el método `setTarifa(double tarifa)` debe modificar el valor de la tarifa del alquiler almacenado en el objeto. El cuerpo de un método 'set' asigna al atributo del objeto el parámetro de la declaración.

Un método 'set' se declara `public`

El valor de retorno es `void`

La lista de parámetros de un método 'set' incluye el tipo y el nombre del parámetro

```
public void setTarifa(double tarifa) {  
    this.tarifa = tarifa;  
}
```

Un método 'set' modifica el valor de un atributo del objeto. En este caso el identificador del atributo es `tarifa` y se refiere a él como `this.tarifa` para asignarle el valor del parámetro

Ejemplo .- Método operación

Un método de tipo operación es aquel que realiza un cálculo o modifica el estado de un objeto. Este tipo de métodos pueden incluir una lista de parámetros y puede devolver un valor o no. Si el método no devuelve un valor, se declara void.

```
public class Circulo {  
    public static final double PI = 3.1415926536;  
    private double radio;  
  
    public Circulo(double radio) {  
        this.radio = radio;  
    }  
  
    public double getRadio() {  
        return this.radio;  
    }  
  
    public double calcularPerimetro() {  
        return 2 * PI * this.radio;  
    }  
  
    public double calcularArea() {  
        return PI * this.radio * this.radio;  
    }  
}
```

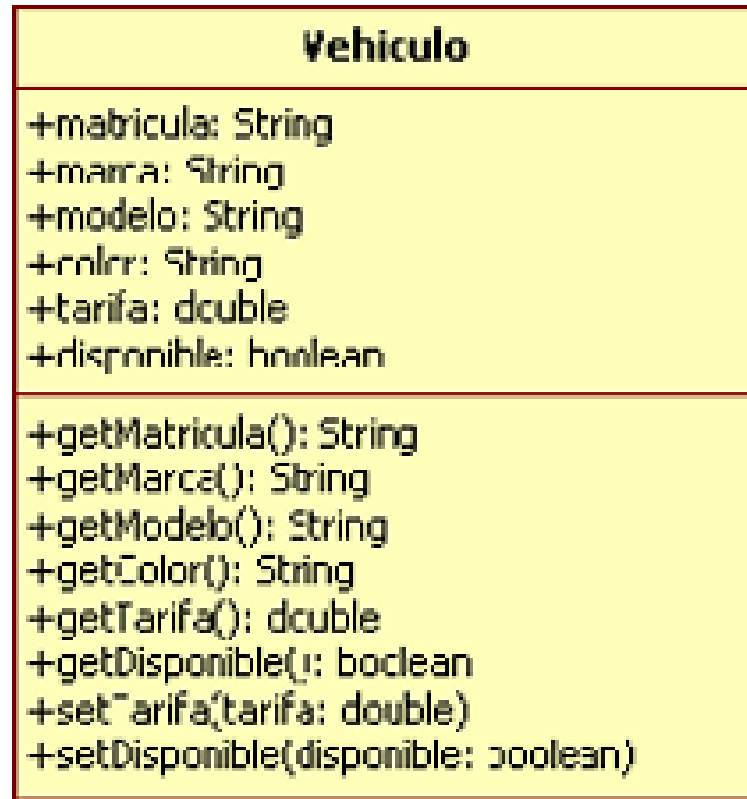
En este ejemplo, los métodos `calcularPerimetro()` y `calcularArea()` devuelven un valor `double` y ninguno de ellos recibe parámetros.

Representación de clases y objetos.- utilizando diagramas de clases

Nombre de la clase

Declaración de atributos

Declaración de métodos



vehiculo1 : Vehiculo

matricula = "4050 ABJ"
marca = "vw"
modelo = "GTI"
color = "Blanco"
tarifa = 100.0
disponible = true

vehiculo2 : Vehiculo

matricula = "2345 JVM"
marca = "SEAT"
modelo = "León"
color = "Negro"
tarifa = 80.0
disponible = false

Los objetos se representan como cajas , la clase a la que pertenecen y el estado del objeto