



CLASE

Una clase describe a un conjunto de objetos que comparten una estructura y un comportamiento común.

Una clase es un molde o plantilla que indica cómo será un objeto de dicha clase. En el área de la construcción, una clase podría ser el plano de una casa que indica la estructura que debe tener en cada una de las casas, y los objetos son la materialización de las casas construidas a partir de dicho plano. Es por ello que se define a un objeto como una instancia de una clase.

Para definir una clase en java se utiliza la palabra reservada 'class'. La sintaxis de dicha instrucción requiere además de la especificación de un nombre para la clase.

Opcionalmente, la palabra class puede estar precedida por modificador de acceso public; si la el modificador de la clase es de tipo public debe ser guardada en un archivo .java cuyo **nombre tiene que ser exactamente el mismo que el de la clase.**

ELEMENTOS DE LA CLASE

ATRIBUTO (ESTADO)

Un atributo es una propiedad que ayuda a describir un Objeto. Los Atributos pueden ser dos clases **Tipo Primitivo y Tipo de Dato de Referencia.**



Tipo de dato Primitivo: Corresponde a un tipo de dato predefinido por el Lenguaje. Cuando se define un atributo (o variable) de este tipo, hay que separar un espacio en memoria para guardar su valor.

Los tipos de datos Primitivos son:

- byte
- short
- int
- long
- float
- double
- boolean
- char.

Tipo de dato de Referencia: Corresponde a un objeto de una clase(no obligatoriamente distinta a la definida). En este punto es donde radica gran parte de la importancia, flexibilidad y reutilización del paradigma de orientación a objetos; porque cuando se definen atributos cuyo tipo sea una clase, se amplía el espectro de posibilidades. Cuando se define un atributo de este tipo no se separa espacio en memoria para un nuevo objeto, sino que se define una referencia que apuntara a un espacio de memoria con la estructura definida en la clase.

METODO (COMPORTAMIENTO)

Abstracción de un acción, servicio, comportamiento o tarea que puede ser realizado por un objeto. Generalmente, un método manipula la información registrada en los atributos a través de una o más instrucciones.



El tipo de dato que retorna como resultado el método. Este tipo de dato puede corresponder a un tipo de dato primitivo, a cualquier tipo de dato de referencia, o en caso de no retornar ningún valor debe especificar el valor void.

Representación de la Clase:**REPRESENTACIÓN UML**

| Nombre de la Clase (Sustantivo) |
|---|
| Atributos [Modificador] [Tipo de dato] [Identificador]; |
| Métodos [Modificador] [Tipo de dato retorna] [Identificador](argumentos) |

CODIFICACION EN JAVA

UNIDAD 2 : Clases, Objetos y métodos



```
[public] class [nombre de la clase]{
```

```
    [Modificador] [Tipo de dato ] [Identificador]=valor; // Atributo
```

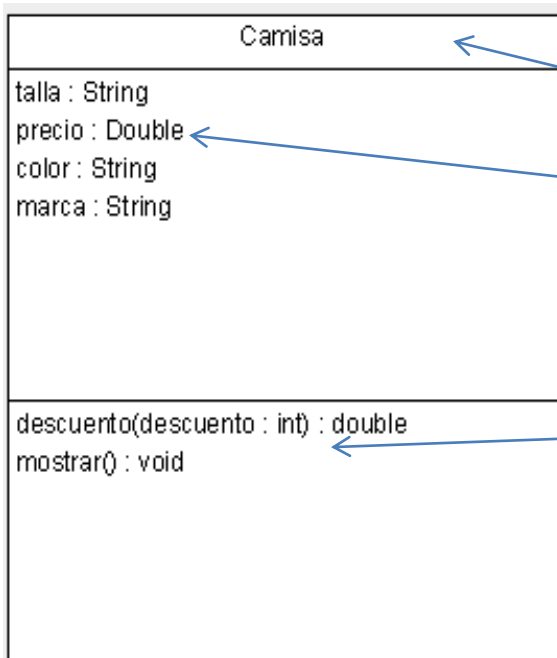
```
        [Modificador] [Tipo de dato retorna ] [Identificador](  
            argumentos) // metodo
```

```
{  
    // bloque del metodo  
}
```

El siguiente ejemplo la Clase Camisa se representa en Diagrama uml de la Clase y codificado en el lenguaje de programación java.

Representación Uml (Diagrama de Clase)

Código en Lenguaje Java



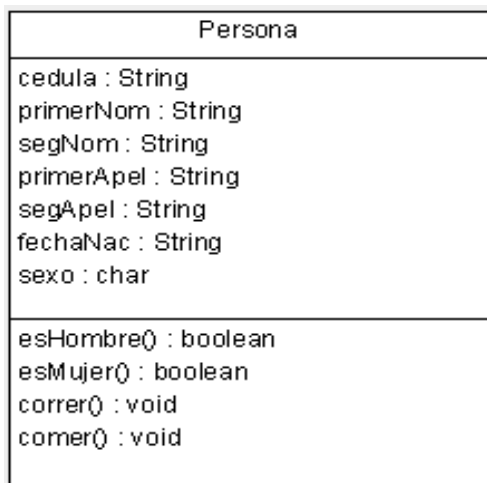
```
public class Camisa {  
    public String talla;  
    public double precio;  
    public String color;  
    public String marca;  
  
    public double descuento(int descuento){  
        precio=precio*(100-descuento)/100;  
        return precio;  
    }// Cierre llave del método  
  
    public void mostrar(){  
        System.out.println("talla "+talla);  
        System.out.println("precio "+precio);  
        System.out.println("color "+color);  
        System.out.println("marca "+marca);  
    } // cierre llave del metodo  
} // cierre llave de la clase
```

The Java code for the 'Camisa' class is shown. It starts with 'public class Camisa {'. It has four public attributes: 'String talla', 'double precio', 'String color', and 'String marca'. It has two public methods: 'descuento(int descuento)' which returns a 'double' and calculates a discount, and 'mostrar()' which prints the attributes. The code ends with '}'.



El siguiente ejemplo la Clase persona se representa en Diagrama uml de la Clase y codificado en el lenguaje de programación java.

Representación uml



Código Java

```
public class Persona {  
  
    public String cedula;  
    public String primerNom;  
    public String segNom;  
    public String primerApel;  
    public String segApel;  
    public String fechaNac;  
    public char sexo;  
  
    public boolean esHombre() {  
        boolean ret = false;  
        if (sexo == 'M' || sexo == 'm') {  
            ret = true;  
        }  
        return ret;  
    }  
  
    public boolean esMujer() {  
        boolean ret = false;  
        if (sexo == 'F' || sexo == 'f') {  
            ret = true;  
        }  
        return ret;  
    }  
  
    public void correr() {  
        System.out.println("Estoy corriendo");  
    }  
  
    public void comer() {  
        System.out.println("Estoy comiendo");  
    }  
}
```



OBJETO

Un objeto es un concepto, abstracción o cosa con límites bien definidos y con significado dentro del problema.

Como se puede apreciar la definición de objeto es realmente amplia pues en realidad cualquier cosa puede ser un objeto. De allí puedan existir objetos que representen cosas concretas (Automóviles, casa, libros, etc). Y objetos que representan cosas abstractas (como pensamientos, ideas etc).

Este es un ejemplo del objeto Tengo la clase Camisa y creo el objeto que se llama camisa1 que tiene atributos o estado el color es amarillo, la talla es m, precio es 200.000 mil pesos y es de marca Ralph Lauren, pueden tener un comportamiento que es mostrar los datos del objetos y si hay algún descuento.

| | |
|---|--|
|  | <p>Objeto Camisa</p> <p>Atributos (Estado)</p> <p>talla:m</p> <p>Precio: 200.000</p> <p>marca: Ralph Lauren</p> <p>color: Amarillo</p> <p>Métodos (Comportamiento)</p> <ul style="list-style-type: none">• Mostrar los datos en Pantalla Con el método Mostrar();• Realizar Descuento con el método con el argumento de tipo entero descuento retorna un tipo doble que el precio con descuento es: double descuento(int descuento) |
|---|--|



Imagen tomada de Buscador Google

INSTANCIA

En java, para crear un nuevo objeto es el momento cuando se instancia la clase, es necesario utilizar el comando new, seguido del nombre de la clase que se desea instanciar; entonces para crear un nuevo empleado se utiliza las siguientes instrucciones.

```
new Empleado();
```



Operador new Reserva Espacio en Memoria

Con esta instrucción java separa un espacio en memoria con la estructura definida para la clase, tomando como ejemplo la clase Empleado (debe haber espacio para almacenar cedula, primer Nombre, Segundo Nombre, etc.);

METODO CONSTRUCTOR

Método Especial que crea un objeto de la clase y que puede emplearse para especificar aquellas tareas que deben realizarse en el momento de la creación como, por ejemplo, la inicialización de los valores de los atributos.

El método constructor es un método especial, y su definición varia ligeramente del resto, las particularidades son:

UNIDAD 2 : Clases, Objetos y métodos



Un método constructor no puede tener definido ningún tipo de dato de retorno cuando ni siquiera el valor de void es valido.

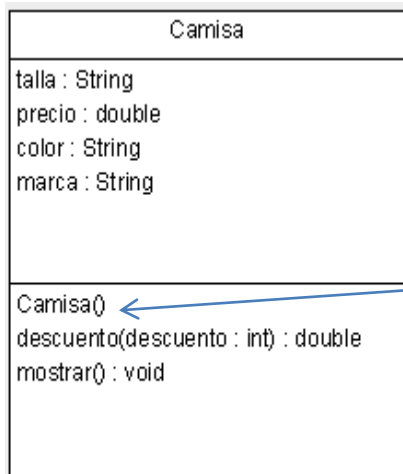
El nombre del método deber ser el mismo nombre de la clase (debe coincidir exactamente con el nombre cuidando hasta el uso de letras mayúscula y minúsculas).

```
new Empleado();
```



Constructor

El Lenguaje de programación Java sino encuentra un constructor, lo crea por defecto, el constructor puede tener argumentos, un ejemplo será la clase Camisa en el constructor inicializo los valores de los atributos de la clase .



Método
Constructor

```
public class Camisa {  
    public String talla;  
    public double precio;  
    public String color;  
    public String marca;
```

```
    public Camisa() {  
        talla="";  
        precio=0.0;  
        color="";  
        marca="";  
    }
```

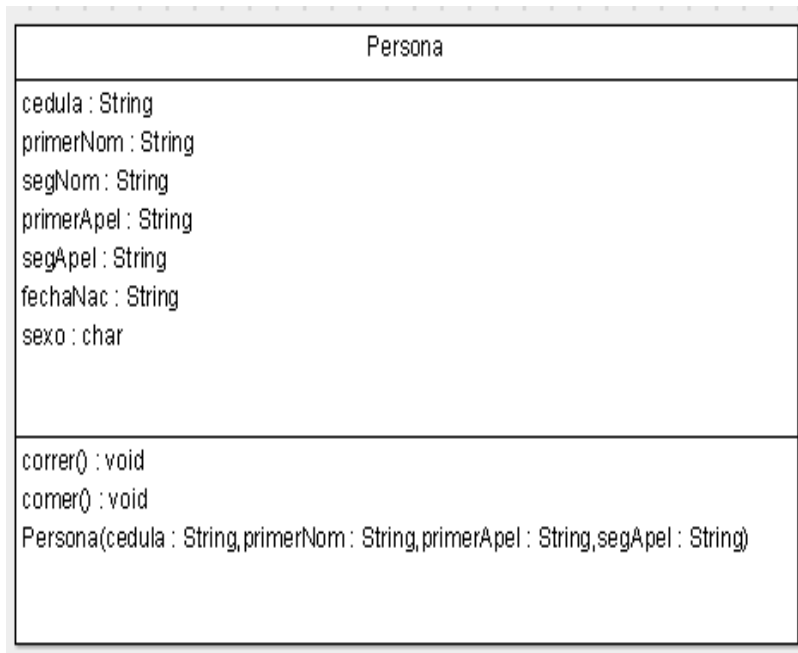
```
    public double descuento(int descuento){  
        precio=precio*(100-descuento)/100;  
        return precio;  
    }
```

```
    public void mostrar(){  
        System.out.println("talla "+talla);  
        System.out.println("precio "+precio);  
        System.out.println("color "+color);  
        System.out.println("marca "+marca);  
    }  
}
```

El constructor también puede contener argumentos o parámetros, a continuación en la clase Persona tiene los siguientes parámetros cedula, primer apellido, segundo apellido, primer nombre y segundo nombre en el constructor.

Representación Uml

UNIDAD 2 : Clases, Objetos y métodos



Método Constructor con
argumentos o
parámetros



Código Java

UNIDAD 2 : Clases, Objetos y métodos



```
public class Persona {  
  
    public String cedula;  
    public String primerNom;  
    public String segNom;  
    public String primerApel;  
    public String segApel;  
    public String fechaNac;  
    public char sexo;  
  
    public Persona(String cedulae, String primerNome, String segNome, String primerApele) {  
        cedula = cedulae;  
        primerNom = primerNome;  
        segNom = segNome;  
        primerApel = primerApele;  
    }  
  
    public void correr() {  
        System.out.println("Estoy corriendo");  
    }  
  
    public void comer() {  
        System.out.println("Estoy comiendo");  
    }  
}
```

Método
Constructor
con
argumentos o
parámetros

Bibliografía:

Programación Orientada a Objetos: Conceptos, Modelado, Diseño y Codificación en C++, Autor: Luis Joyanes Aguilar.

Programación Orientada a Objetos en Java, Autor: Luis Fernando García Lilinas