

Programación Frontend y Backend

BLOQUE JAVA

Objetos I

01

Objetos



GOBIERNO
DE ESPAÑA

MINISTERIO
DE INDUSTRIA, COMERCIO
Y TURISMO



Escuela de
organización
industrial



Unión Europea
Fondo Social Europeo
Iniciativa de Empleo Juvenil
El FSE invierte en tu futuro



¿Qué es un Objeto?

Meter Libro / Sacar Libro



Poner / Quitar



1,0l



0,8l



0,5l



¿Qué es un Objeto?

- Mediante el uso de variables de tipo primitivo podemos representar el nombre de una persona, la edad de una animal, la matrícula de un vehículo, etc. Pero y si queremos representar a una Persona o un Animal o incluso un Vehículo.

¡VAMOS A CREAR UNA PERSONA!

¿Qué variables necesitamos para crear una persona?

¿Qué es un Objeto?

Tipos de Pelo

- Color
- Negro
- Rubio
- Colorin (rojo)
- Castaño
- Blanco

Longitud

- Largo
- Corto
- Calvo

Forma

- Ondulado (crespo)
- liso

Color de ojos

- Azul
- Verde
- Café
- Negro

¿...?

Abierta
Activa
Actual
Adaptable
Afable
Ágil de mente
Agresiva
Alerta
Amable
Ambiciosa
Amplia de mente
Analítica
Animosa
Aplomo
Asertiva
Atenta
Autentica
Capaz
Con carácter
Cauta
Clara

Cooperativa
Coordinadora
Cordial
Cortes
Creativa
Coherente
Con criterio
Crítica
Cuidadosa
Culta
Cumplidora
Decidida
Delegadora
Desenvuelta
Dialogante
Diplomática
Discreta
Duro
Económica
Ecuánime
Eficaz

Fiable
Fiel
Firme
Flemática
Flexible
Formal
Gerencial
Hábil
Honesta
Imaginativa
Independiente
Especializada
Justa
Laboriosa
Leal
Capacidad de líder
Lógica
Madura
Dotes de mando
Mañosa
Matemática

Persuasiva
Polivalente
Ponderada
Positiva
Practica
Precavida
Precisa
Productiva
Puntual
Rápida
Razonable
Recta
Con recursos
Reflexiva
Relacionada
Respetuosa
Responsable
Resolutiva
Con salud
Segura
Sensata



¿Qué es un Objeto?

- Empezamos a hacernos una idea de lo complicado y tedioso que se puede volver con los mecanismos que conocemos actualmente, ya que únicamente para guardar todas las características de la cara de una persona necesitaríamos unas 100 variables.
- Imaginad si tenemos que crear a 2 personas o 10 o 100 ... mantener ese código, hacer un cambio en alguna característica ...
- El dato primitivo empieza a quedarse un poco corto para todo lo que necesitamos hacer, necesitaríamos poder crear nosotros/as mismos/as un tipo de dato ... y si lo llamamos Objeto.

¿Qué es un Objeto?



Molde



Cupcakes

¿Qué es un Objeto?



Clase



Objetos

¿Qué es una Clase?

- Una clase es un modelo o prototipo definido por el usuario a partir del cual se crean los objetos. Representa el conjunto de propiedades o métodos que son comunes a todos los objetos de un tipo.

 Cupcake.java



```
public class Cupcake {  
      
}
```



GOBIERNO
DE ESPAÑA

MINISTERIO
DE INDUSTRIA, COMERCIO
Y TURISMO



Escuela de
organización
industrial



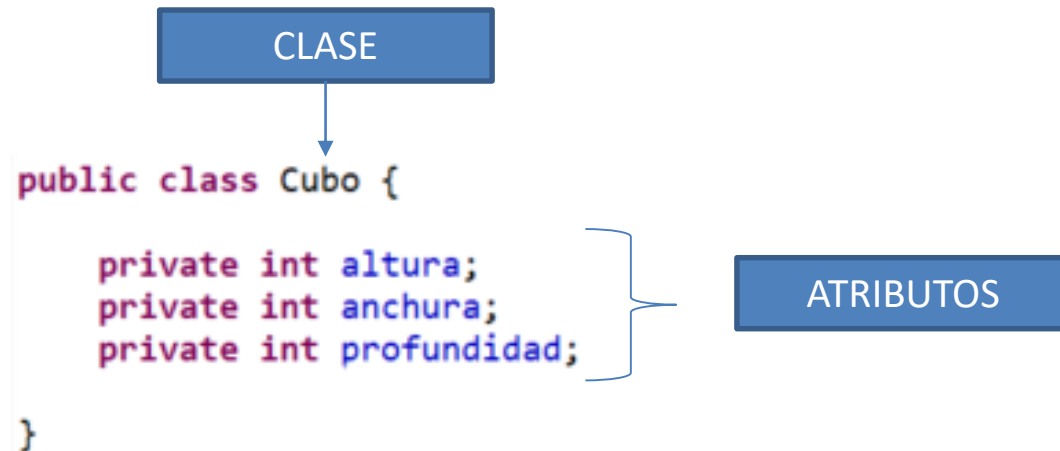
Unión Europea
Fondo Social Europeo
Iniciativa de Empleo Juvenil
El FSE invierte en tu futuro



Clase

- El archivo java tendrá el mismo nombre que el nombre de la clase.
- Dentro de una clase podemos definir:
 - Estados / Propiedades / Atributos > nombre, color, tamaño
 - Constructores
 - Comportamientos / Métodos > cantar, saltar, ladrar

Clase (Atributos)



Clase (Constructores)

CLASE

```
public class Cubo {  
  
    private int altura;  
    private int anchura;  
    private int profundidad;  
  
    public Cubo(int alt, int anch) {  
        this.altura = alt;  
        this.anchura = anch;  
        this.profundidad = 5;  
    }  
  
    public Cubo(int alt, int anch, int prof) {  
        this.altura = alt;  
        this.anchura = anch;  
        this.profundidad = prof;  
    }  
}
```

ATRIBUTOS

CONSTRUCTOR

GOBIERNO
DE ESPAÑAMINISTERIO
DE INDUSTRIA, COMERCIO
Y TURISMOEscuela de
organización
industrialUnión Europea
Fondo Social Europeo
Iniciativa de Empleo Juvenil
El FSE invierte en tu futuroSISTEMA NACIONAL DE
GARANTÍA
JUVENIL

Clase (Constructores)

- Toda clase tiene un constructor vacío por defecto.
- No hay límite de constructores siempre y cuando tengan una combinación de parámetros y tipo de parámetros distinta.
- Son muy útiles ya que se encarga de asignar valores a cada objeto durante el proceso de instanciación.

```
public class Cubo {  
  
    private int altura;  
    private int anchura;  
    private int profundidad;  
  
    public Cubo(int alt, int anch) {  
        this.altura = alt;  
        this.anchura = anch;  
        this.profundidad = 5;  
    }  
  
    public Cubo(int alt, int anch, int prof) {  
        this.altura = alt;  
        this.anchura = anch;  
        this.profundidad = prof;  
    }  
}
```



Clase (Constructores)

- Al crear un objeto estamos instanciando una clase, podríamos decir que estamos usando el molde (clase) para crear un objeto nuevo.

```
public static void main(String[] args) {  
  
    Cubo cuboS = new Cubo(2, 2, 2);  
    Cubo cuboM = new Cubo(3, 3, 3);  
    Cubo cuboL = new Cubo(4, 4, 4);  
  
}
```

```
public class Cubo {  
  
    private int altura;  
    private int anchura;  
    private int profundidad;  
  
    public Cubo(int alt, int anch) {  
        this.altura = alt;  
        this.anchura = anch;  
        this.profundidad = 5;  
    }  
  
    public Cubo(int alt, int anch, int prof) {  
        this.altura = alt;  
        this.anchura = anch;  
        this.profundidad = prof;  
    }  
  
}
```


Clase (Atributos)

- Todos los atributos de una clase suelen ser privados, esto quiere decir que únicamente la clase Cubo tiene acceso tanto para obtener el valor como para asignarle uno nuevo.

```
Cubo cuboS = new Cubo(2, 2, 2);  
Cubo cuboM = new Cubo(3, 3, 3);  
Cubo cuboL = new Cubo(4, 4, 4);
```

cuboS.

- equals(Object obj) : boolean - Object
- getClass() : Class<?> - Object
- hashCode() : int - Object
- notify() : void - Object
- notifyAll() : void - Object
- toString() : String - Object
- wait() : void - Object
- wait(long timeout) : void - Object
- wait(long timeout, int nanos) : void - Object

¿Altura?

¿Anchura?

¿Profundidad?

Clase (Atributos)

- Tenemos 2 opciones:
- Convertir los atributos en públicos.

```
public int altura;  
public int anchura;  
public int profundidad;
```

```
Cubo cuboS = new Cubo(2, 2, 2);  
Cubo cuboM = new Cubo(3, 3, 3);  
Cubo cuboL = new Cubo(4, 4, 4);
```

cuboS.

- altura : int - Cubo
- anchura : int - Cubo
- profundidad : int - Cubo

Clase (Atributos)

- Tenemos 2 opciones:
 - Crear **Getter & Setter** (Recomendado)

```
public int getAltura() {  
    return altura;  
}  
  
public void setAltura(int altura) {  
    this.altura = altura;  
}
```

```
public class Cubo {  
  
    private int altura;  
    private int anchura;  
    private int profundidad;  
  
    public Cubo(int alt, int anch) {  
        this.altura = alt;  
        this.anchura = anch;  
        this.profundidad = 5;  
    }  
  
    public Cubo(int alt, int anch, int prof) {  
        this.altura = alt;  
        this.anchura = anch;  
        this.profundidad = prof;  
    }  
}
```

get

- `getAltura()` : int - Getter for 'altura'
- `getAnchura()` : int - Getter for 'anchura'
- `getProfundidad()` : int - Getter for 'profundidad'

Clase (Atributos)

- Tenemos 2 opciones:
 - Crear **Getter & Setter** (Recomendado)

```
Cubo cuboS = new Cubo(2, 2, 2);  
Cubo cuboM = new Cubo(3, 3, 3);  
Cubo cuboL = new Cubo(4, 4, 4);
```

```
System.out.println("Cubo S: "+  
    cuboS.getAltura()+"x"+  
    cuboS.getAnchura()+"x"+  
    cuboS.getProfundidad());
```

} get

```
cuboS.setAltura(3);  
cuboS.setAnchura(3);  
cuboS.setProfundidad(3);
```

} set

```
System.out.println("Cubo S: "+  
    cuboS.getAltura()+"x"+  
    cuboS.getAnchura()+"x"+  
    cuboS.getProfundidad());
```

} get

```
public int getAltura() {  
    return altura;  
}  
  
public void setAltura(int altura) {  
    this.altura = altura;  
}  
  
public int getAnchura() {  
    return anchura;  
}  
  
public void setAnchura(int anchura) {  
    this.anchura = anchura;  
}  
  
public int getProfundidad() {  
    return profundidad;  
}  
  
public void setProfundidad(int profundidad) {  
    this.profundidad = profundidad;  
}
```

```
Markers Console  
<terminated> EjemploCubo [Java Application]  
Cubo S: 2x2x2  
Cubo S: 3x3x3
```

¿Podemos utilizar ya nuestros cubos?

Objetos (Declaración e Instanciación)

- Igual que declaramos variables de tipos de datos primitivos, esta es la manera de declarar variables de tipo Objeto.

```
Cubo cubo1;  
Cubo cubo2;
```

Declaración

```
Cubo cubo1 = new Cubo(1,1,1);  
Cubo cubo2 = new Cubo(1,1,1);
```

Declaración e Instanciación

¿Podemos utilizar ya nuestros cubos?

Objetos (Declaración e Instanciación)

- Al haber declarado varios constructores en la clase Cubo, la manera de instanciar nuestros cubos puede varias:

```
Cubo cubo1 = new Cubo(1,1,1);  
Cubo cubo2 = new Cubo(1,1,1);
```

Declaración e Instanciación

```
Cubo cubo3 = new Cubo(1,1);  
Cubo cubo4 = new Cubo(1,1);
```

Declaración e Instanciación

Clase (Métodos)

- Otra capacidad de la programación orientada a objetos, consiste en definir un comportamiento común que tendrán todos los objetos que sean instanciados de una clase.

```
public int calculaVolumen() {  
    return this.altura * this.anchura * this.profundidad;  
}
```

```
Cubo cuboS = new Cubo(2, 2, 2);  
Cubo cuboM = new Cubo(3, 3, 3);  
Cubo cuboL = new Cubo(4, 4, 4);
```

```
System.out.println("El volumen del cuboM es:" + cuboM.calculaVolumen());
```

02

Ejercicios



GOBIERNO
DE ESPAÑA

MINISTERIO
DE INDUSTRIA, COMERCIO
Y TURISMO



Escuela de
organización
industrial



Unión Europea
Fondo Social Europeo
Iniciativa de Empleo Juvenil
El FSE invierte en tu futuro



Ejercicio I

Vamos a crear una clase `Persona.java` esta clase tendrá todas las propiedades que declaramos en el ejercicio de los datos primitivos, debemos poder acceder y/o modificar a cada propiedad, tendrá un constructor con todos los parámetros.

A continuación desde una clase `main` llamada `Clase.java` almacenaremos 10 personas en un array, por último recorreremos el array e imprimiremos el siguiente mensaje por cada persona (El mensaje lo escribirá cada persona a través de un método en la clase `Persona` llamado `saludar()`):

Hola me llamo **{Nombre} {Apellidos}** y tengo **{Edad}** años.

Nací el **{fecha de nacimiento}** y vivo en **{Ciudad}**

Actualmente **estoy/no estoy trabajando y estoy/no estoy estudiando.**

Ejercicio II

Vamos a crear una clase Vehiculo.java, deberemos de poder almacenar la marca, el modelo, el color, si es gasolina, diésel, eléctrico o híbrido, plazas, año de matriculación y el precio por el que se vende, así como el nombre y el teléfono del vendedor. También crearemos una clase main llamada SegundaMano.java dónde listaremos todos los vehículos que queremos anunciar en nuestra aplicación de la siguiente manera:

VENTA DE COCHES

Coche 1 – OPEL ASTRA 2006 – 4.400 €

Coche 2 – TOYOTA AYGO 2018 – 8.900 €

Coche 3 – KIA STONIC 2017 – 12.500 €

Seleccione el coche del que desea conocer más detalles: 3

KIA STONIC 2017 – 12.500 € (GASOLINA) – 5 Plazas

(RESTO DE CAMPOS ...)

Vendedor: Pedro García – 665789965