



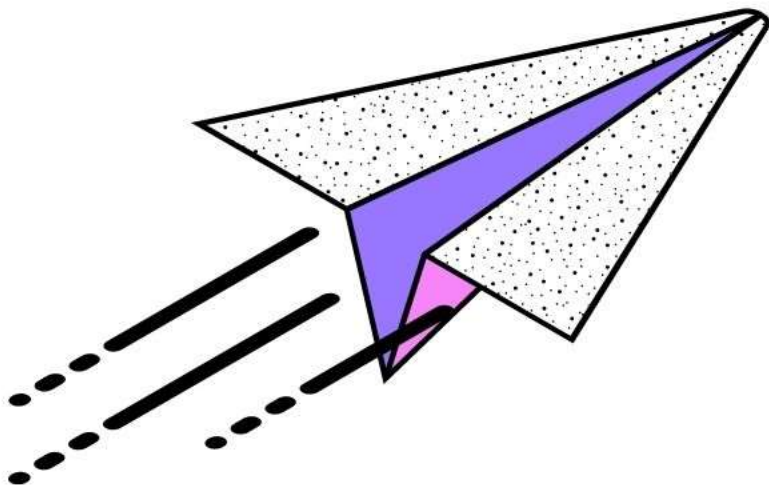
Java Standard Edition





Sección 2

Pensando en objetos



Identificar objetos y reconocer los criterios para definirlos.

Introducción al lenguaje Java.

Compilar y ejecutar un programa de prueba.

Identificar los componentes de una clase.

Definir clases.

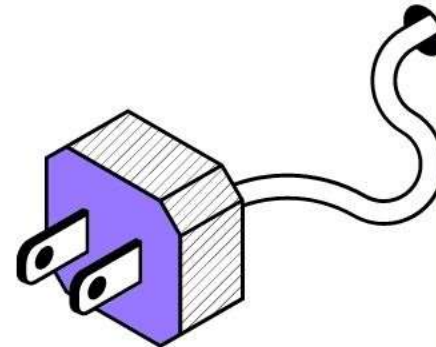
Crear y utilizar una clase de prueba.

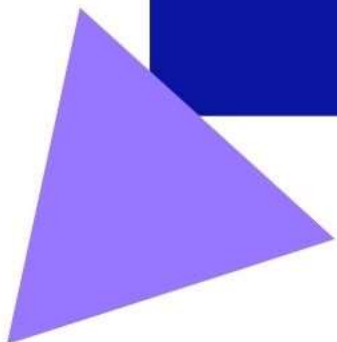
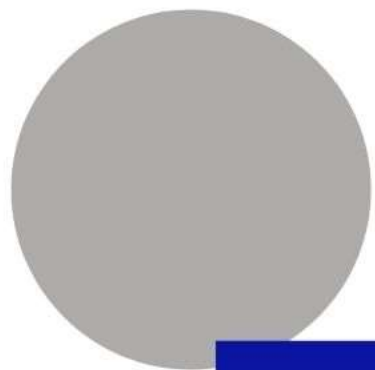


¿Qué es un Objeto?

Definición dentro de la POO

- Un objeto representa una **entidad** en el mundo real que se puede identificar claramente.
- Un objeto tiene una **identidad**, un **estado** y un **comportamiento** únicos.





Estado de un Objeto

El estado de un objeto (también conocido como sus propiedades o atributos) está representado por campos de datos con sus valores actuales.

Comportamiento de un Objeto

El comportamiento de un objeto (también conocido como sus acciones) se define mediante métodos.

De donde proviene un Objeto

Los objetos del mismo tipo se definen utilizando una clase común. Una clase es una plantilla o modelo o que define cuáles serán los campos de datos y métodos de un objeto.

Relación entre Clase y Objeto

Un objeto es la instancia de una clase. Puede crear muchas instancias de cierta clase. La creación de una instancia se conoce como instanciación.



Conceptos importantes

Las siguientes cincuenta palabras clave están reservadas para uso del lenguaje Java:

<code>abstract</code>	<code>double</code>	<code>int</code>	<code>super</code>
<code>assert</code>	<code>else</code>	<code>interface</code>	<code>switch</code>
<code>boolean</code>	<code>enum</code>	<code>long</code>	<code>synchronized</code>
<code>break</code>	<code>extends</code>	<code>native</code>	<code>this</code>
<code>byte</code>	<code>final</code>	<code>new</code>	<code>throw</code>
<code>case</code>	<code>finally</code>	<code>package</code>	<code>throws</code>
<code>catch</code>	<code>float</code>	<code>private</code>	<code>transient</code>
<code>char</code>	<code>for</code>	<code>protected</code>	<code>try</code>
<code>class</code>	<code>goto</code>	<code>public</code>	<code>void</code>
<code>const</code>	<code>if</code>	<code>return</code>	<code>volatile</code>
<code>continue</code>	<code>implements</code>	<code>short</code>	<code>while</code>
<code>default</code>	<code>import</code>	<code>static</code>	
<code>do</code>	<code>instanceof</code>	<code>strictfp*</code>	



Conceptos importantes

Las siguientes cincuenta palabras clave están reservadas para uso del lenguaje Java:

Caracter	Nombre	Descripción
{ }	Llaves de apertura y cierre	Denota un bloque para encerrar sentencias.
()	Paréntesis de apertura y cierre	Se utiliza con métodos.
[]	Corchetes de apertura y cierre	Denota una matriz.
//	Doble slash	Preceda una línea de comentario.
" "	Comillas de apertura y cierre	Encierre una cadena (es decir, una secuencia de caracteres).
;	Punto y coma	Marcar el final de una declaración.



Conceptos importantes

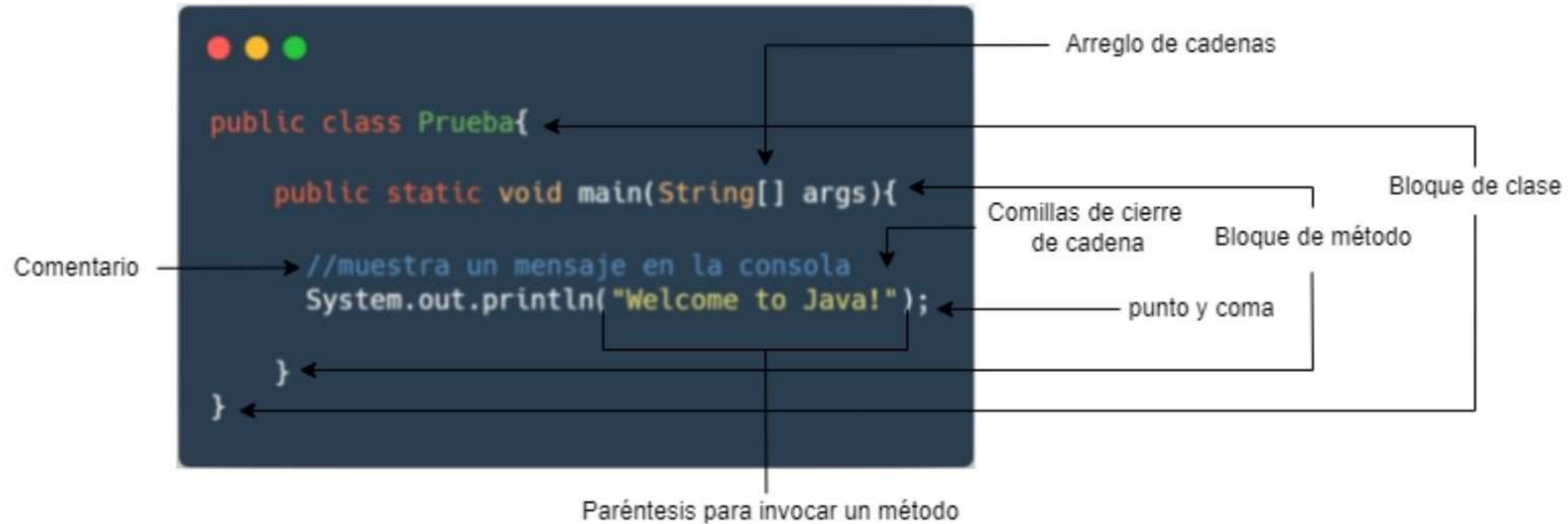
Las siguientes cincuenta palabras clave están reservadas para uso del lenguaje Java:

Caracter	Nombre	Descripción
{ }	Llaves de apertura y cierre	Denota un bloque para encerrar sentencias.
()	Paréntesis de apertura y cierre	Se utiliza con métodos.
[]	Corchetes de apertura y cierre	Denota una matriz.
//	Doble slash	Preceda una línea de comentario.
" "	Comillas de apertura y cierre	Encierre una cadena (es decir, una secuencia de caracteres).
;	Punto y coma	Marcar el final de una declaración.



Elementos de un programa

muestra un mensaje en consola





Componentes de una clase

todas las clases tienen esta estructura básica

```
class NombreClase{  
  
    int atributo1;  
    boolean atributo2;  
    String atributo3;  
  
    void metodo1(){  
        //alguna accion aqui  
    }  
  
    int metodo2(int param){  
        //alguna accion aqui  
    }  
  
    double metodo3(int param1, float param2){  
        //alguna accion aqui  
    }  
  
}
```

Atributos

Metodos



Componentes de una clase

todas las clases tienen esta estructura básica

```
class NombreClase{  
  
    int atributo1;  
    boolean atributo2;  
    String atributo3;  
  
    void metodo1(){  
        //alguna accion aqui  
    }  
  
    int metodo2(int param){  
        //alguna accion aqui  
    }  
  
    double metodo3(int param1, float param2){  
        //alguna accion aqui  
    }  
  
}
```

Atributos

Metodos



Componentes de una clase

todas las clases tienen esta estructura básica

```
class NombreClase{  
  
    int atributo1;  
    boolean atributo2;  
    String atributo3;  
  
    void metodo1(){  
        //alguna accion aqui  
    }  
  
    int metodo2(int param){  
        //alguna accion aqui  
    }  
  
    double metodo3(int param1, float param2){  
        //alguna accion aqui  
    }  
  
}
```

Atributos

Metodos

Ejercicios practicos



Diseñar una clase llamada Rectángulo y otra llamada Televisor.