

Documentación

Los tutoriales de Java™

Sendero: Aprender el lenguaje Java

Lección: Clases y objetos

Sección: Más sobre clases

Los tutoriales de Java se escribieron para JDK 8. Los ejemplos y prácticas descritos en esta página no aprovechan las mejoras introducidas en versiones posteriores y pueden utilizar tecnología que ya no está disponible.

Consulte [Cambios en el lenguaje Java](#) para obtener un resumen de las funciones de lenguaje actualizadas en Java SE 9 y versiones posteriores.

Consulte [las Notas de la versión de JDK](#) para obtener información sobre nuevas funciones, mejoras y opciones eliminadas o obsoletas para todas las versiones de JDK.

Usando esta palabra clave

Dentro de un método de instancia o un constructor, `this` hay una referencia al *objeto actual*: el objeto cuyo método o constructor se llama. Puede hacer referencia a cualquier miembro del objeto actual desde un método de instancia o un constructor utilizando `this`.

Usar `this` con un campo

La razón más común para usar la `this` palabra clave es que un campo está sombreado por un método o parámetro de constructor.

Por ejemplo, la `Point` clase fue escrita así.

```
Punto de clase pública {
    público int x = 0;
    público int y = 0;

    //constructor
    Punto público (int a, int b) {
        x = a;
        y = b;
    }
}
```

pero podría haberse escrito así:

```
Punto de clase pública {
    público int x = 0;
    público int y = 0;

    //constructor
    Punto público (int x, int y) {
        esto.x = x;
        esto.y = y;
    }
}
```

Cada argumento del constructor oculta uno de los campos del objeto; dentro del constructor `x` hay una copia local del primer argumento del constructor. Para hacer referencia al `Point` campo `x`, el constructor debe utilizar `this.x`.

Usando `this` con un constructor

Desde dentro de un constructor, también puedes usar la `this` palabra clave para llamar a otro constructor en la misma clase. Hacer esto se denomina *invocación explícita del constructor*. Aquí hay otra `Rectángulo` clase, con una implementación diferente a la de la sección [Objetos](#).

```
rectángulo de clase pública {
    privado int x, y;
    ancho, alto privado int;

    Rectángulo público() {
        esto(0, 0, 1, 1);
    }
    Rectángulo público (int ancho, int alto) {
        this(0, 0, ancho, alto);
    }
}
```

```
    }  
    Rectángulo público (int x, int y, int ancho, int alto) {  
        esto.x = x;  
        esto.y = y;  
        this.width = ancho;  
        this.height = altura;  
    }  
    ...  
}
```

Esta clase contiene un conjunto de constructores. Cada constructor inicializa algunas o todas las variables miembro del rectángulo. Los constructores proporcionan un valor predeterminado para cualquier variable miembro cuyo valor inicial no lo proporcione un argumento. Por ejemplo, el constructor sin argumentos crea un 1x1 `Rectangle` en las coordenadas 0,0. El constructor de dos argumentos llama al constructor de cuatro argumentos, pasando el ancho y el alto pero siempre usando las coordenadas 0,0. Como antes, el compilador determina a qué constructor llamar, según el número y el tipo de argumentos.

Si está presente, la invocación de otro constructor debe ser la primera línea del constructor.

[Acerca de Oracle](#) | [Contáctenos](#) | [Avisos Legales](#) | [Términos de uso](#) | [Sus derechos de privacidad](#)

Copyright © 1995, 2022 Oracle y/o sus filiales. Reservados todos los derechos.

Página anterior: Devolver un valor de un método

Página siguiente: Controlar el acceso a los miembros de una clase