

Informática Avanzada

Ingeniería Electrónica
Ingeniería en Computación

Objetos, clases,
campos y métodos



Contenidos del Curso

- Introducción a la programación orientada a objetos...
- ...con un fuerte fundamento de ingeniería de software ...
- ...orientado a producir y mantener sistemas de software grandes, y de alta calidad.

Palabras Claves

Diseño dirigido por responsabilidad

herencia

encapsulamiento

iteradores

sobreescritura

acoplamiento

cohesion

javadoc

interface

clases collection

métodos mutador

llamado a métodos polimórficos

Metas

- Conocimiento consistente de los principios de programación
- Conocimiento consistente de la orientación a objetos
- Capacidad para evaluar críticamente la calidad de un sistema de software (pequeño)
- Capacidad para implementar un sistema de software pequeño en Java

Libro

David J. Barnes & Michael Kölling

Programación Orientada a Objetos con
Java. Una introducción práctica usando
BlueJ

Tercera edición,
Pearson Educación, 2007
ISBN 978-84-8322-350-5.

Visión general del curso (1)

- Objetos y clases
- Comprendiendo las definiciones de clase
- Interacción entre objetos
- Agrupando objetos
- Librerías con comportamiento más sofisticado
- Objetos con buen-comportamiento: testeo, mantenimiento, depuración
- Diseño de clases

Visión general del curso (2)

- Herencia
- Polimorfismo
- Estructuras de clase flexibles y extendibles
- Construcción de interfaces gráficas de usuario
- Manejo de errores
- Diseño de aplicaciones

Demostración - BlueJ

Iniciar el proyecto *formas*

Conceptos Fundamentales

- objeto
- clase
- método
- parámetro
- tipos de datos

Objetos y clases

- Objetos
 - representan ‘cosas’ del mundo real, o de algun dominio de problema (ejemplo: “el auto rojo allí en la playa de estacionamiento”)
 - *Crear una instancia de **Círculo** (**Circle**, click en boton derecho).*

Ej. 1.1

- Clases
 - representan todos los objetos de un tipo (ejemplo: “auto”)

Métodos y parámetros

- Los objetos tienen operaciones que pueden ser invocados (Java los llama *métodos*). *Hacer visible un círculo.*

Ej.: 1.2

- Los métodos pueden tener parámetros para pasar información adicional necesaria para la ejecución. *MoverHorizontal(distancia)*

Ej.: 1.3; 1.4; 1.5; 1.6; 1.7;

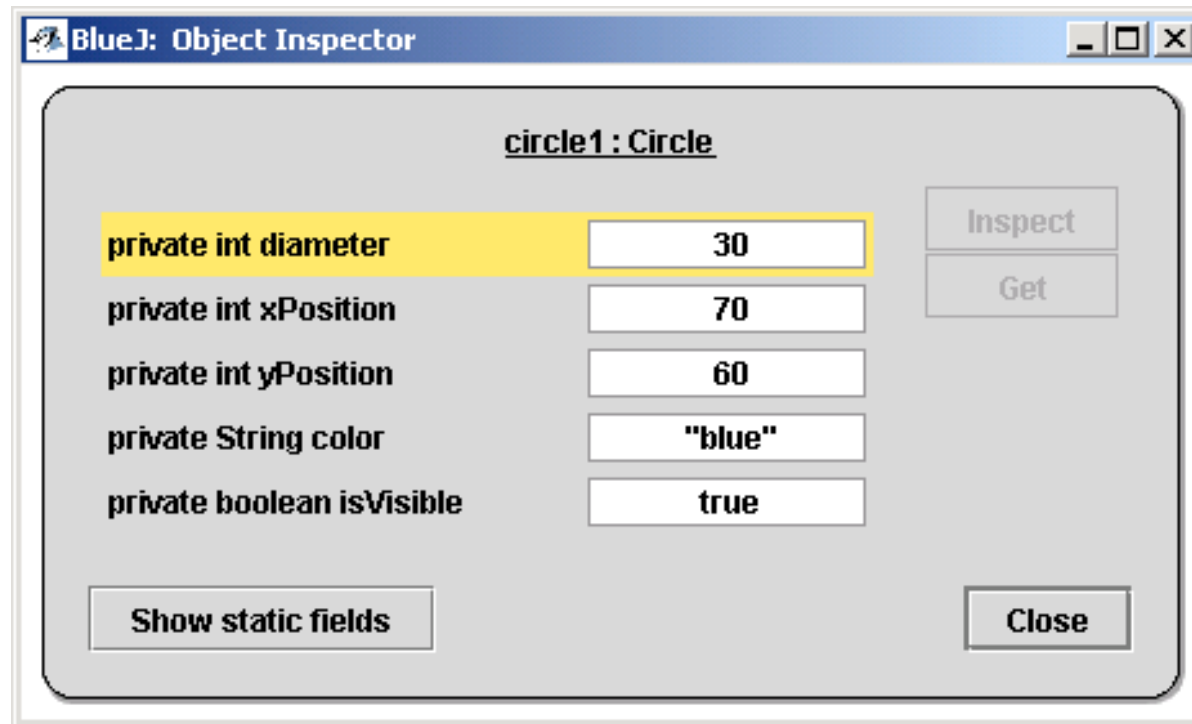
Otras observaciones

- A partir de una única clase se pueden crear muchas *instancias*.
- Un objeto tiene *atributos*: valores almacenados en *campos*. *Invocar el método inspect de un círculo.*

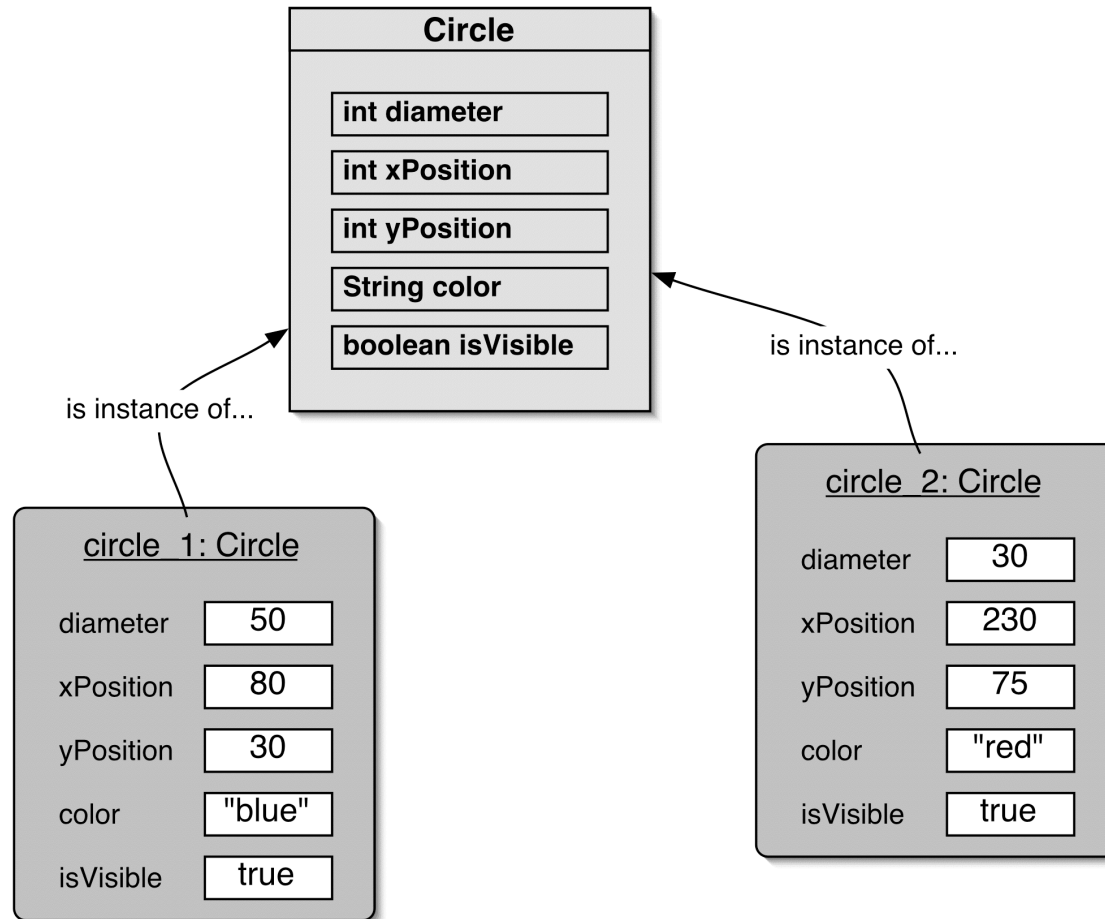
Ej. : 1.8

- La clase define que campos tiene un objeto, pero cada objeto almacena su propio conjunto de valores (el *estado* de un objeto).

Estado



Dos objetos círculo



Ej. : 1.9; 1.10; 1.11

Código fuente

- Cada clase tiene su propio código fuente (código Java) asociado con él y que define sus detalles (campos y métodos). *Abra el proyecto cuadro y seleccione Open Editor del menú contextual de una clase o haciendo doble click.*

Ej. : 1.12

- *Si el ícono de una clase aparece rayado entonces necesita ser compilada. (Botón Compile).*

Ej. : 1.13; 1.14; 1.15; 1.16

Código fuente

- *Cierre el proyecto cuadro (picture) y abra el proyecto curso-de-laboratorio (lab-classes).*

Ej. : 1.17

Valores retornados

- Los métodos pueden retornar un resultado vía un valor de *return*.

Ej. : 1.18

- La palabra *void* indica que un método no retorna ningún resultado.

Objetos como parámetros

- Los objetos pueden ser pasados como parámetros a los métodos de otros objetos.

Ej. : 1.19; 1.20; 1.21; 1.22; 1.23; 1.24; 1.25

Ejercicios de resumen

Términos empleados:

Objeto, clase, método, signature,
parámetro, tipo, estado, código fuente,
valor de retorno y compilador.

Tipos de datos en JAVA

Ej. : 1.26; 1.27; 1.28; 1.29; 1.30; 1.31; 1.32