



Conception objet en Java avec BlueJ une approche interactive

1. Introduction

David J. Barnes, Michael Kölling
version française: Patrice Moreaux



Contenu du cours

- Introduction à la programmation orientée objet ...
- ... fondée sur une solide approche génie logiciel ...
- ... dans le but de produire et maintenir de grands systèmes logiciels de haute qualité.



Mots à la mode...

conception dirigée par les responsabilités

encapsulation

héritage

itérateurs

surcharge

couplage

cohésion

interface

javadoc

modificateur (méthode)

classes collections

appels de méthodes polymorphes



Objectifs

- Acquérir une bonne connaissance des principes de programmation
- Acquérir de solides connaissances en orientation objet
- Être capable d'évaluer de manière critique la qualité d'un système logiciel (modeste)
- Savoir implanter un «petit» système logiciel en Java



Livre

David J. Barnes & Michael Kölling

Conception objet en Java avec BlueJ une approche interactive

Pearson Education France, 2003

ISBN 2-7440-7017-3





Pages Web

- La page de ce cours est à

www . - - - - -

Veillez la consulter régulièrement.

- Elle sera utilisée pour les annonces et la distribution des compléments.
- Voir aussi **www . bluej . org**



Plan du cours (1)

- Objets et classes
- Comprendre les définitions des classes
- Interactions entre objets
- Groupement d'objets
- Comportements plus complexes - bibliothèques
- Objets bien conçus - test, maintenance, mise au point
- Conception des classes



Plan du cours (2)

- Héritage
- Polymorphisme
- Structure de classes extensible et flexible
- Gestion des erreurs
- Conception des applications



Démonstration ...



Concepts fondamentaux

- Objet
- Classe
- Méthode
- Paramètre
- Type de données



Objets et classes

- objet
 - représente **un** «élément» du monde réel, ou d'un domaine de problème donné (exemple: «la voiture rouge, là, sur le parking»)
- classe
 - représente **tous** les objets d'une même «catégorie» (exemple: «voiture»)



Méthodes et paramètres

- les objets sont capables de réaliser des actions (appelées *méthodes* en Java) dont on peut demander l'exécution
- une méthode peut avoir des paramètres pour fournir des informations utiles à l'exécution

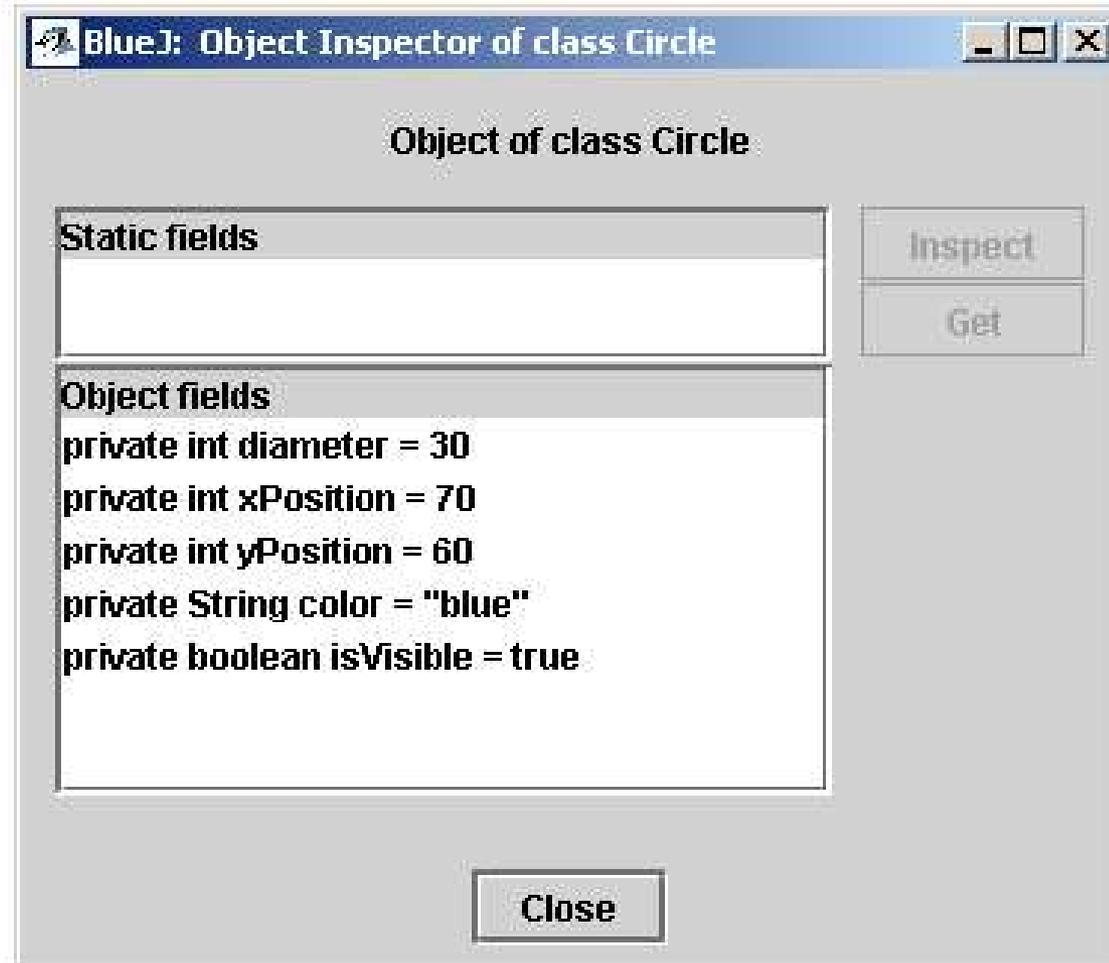


Autres points à noter

- on peut créer plusieurs *instances* d'une seule et même classe
- un objet a des *attributs*: valeurs stockées dans des *champs*
- la classe définit les champs de ses objets, mais chaque objet stocke son propre ensemble de valeurs (l'*état* de l'objet)

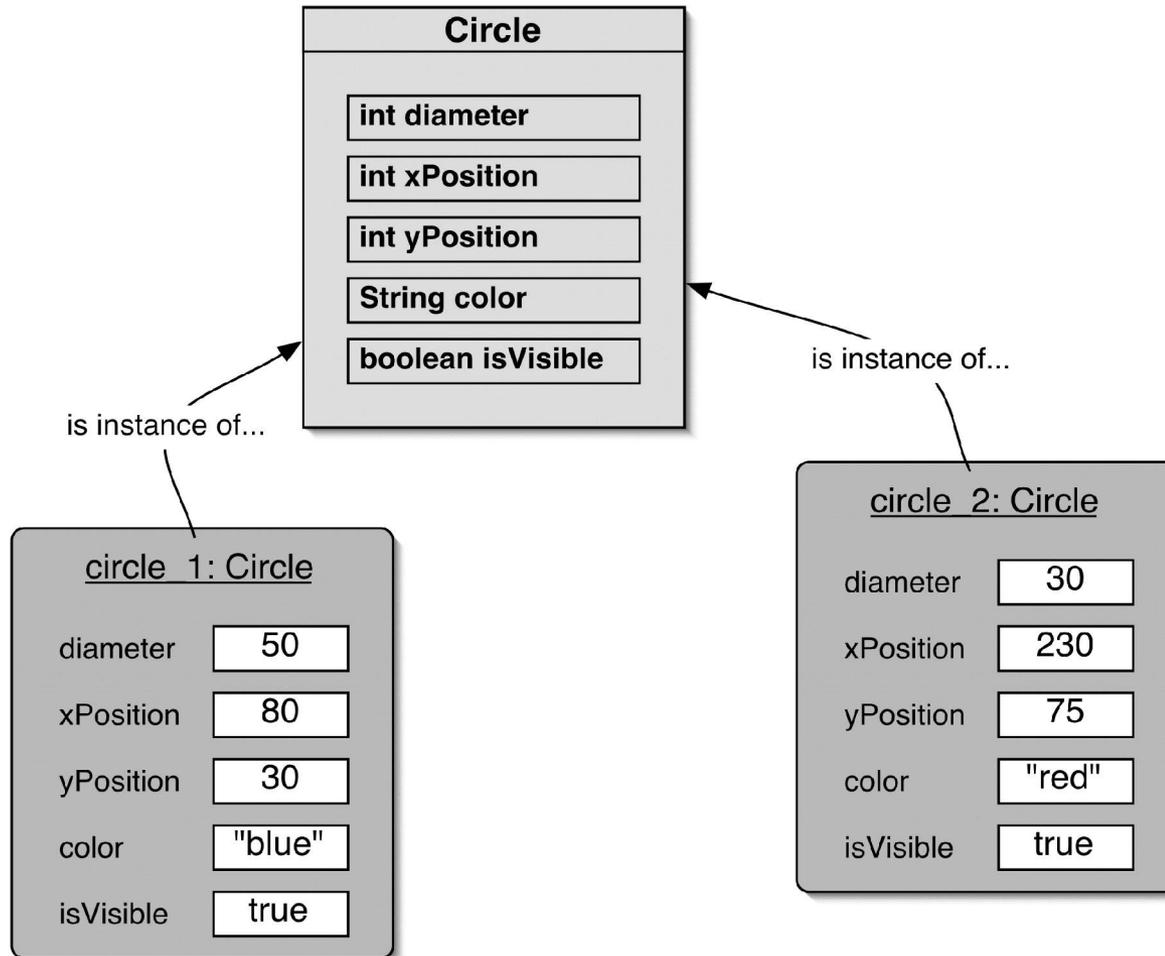


État





Deux objets Circle





Code source

- Chaque classe possède un code source (code Java) qui la définit en détails (champs et méthodes).



Valeurs de retour

- Les méthodes peuvent retourner un résultat via une valeur de retour.



Sommaire général

- 1. Introduction
- 2. Classes
- 3. Interactions d'objets
- 4. Collections et itérateurs
- 5. Bibliothèques de classes
- 6. Tests mise au point
- 7. Conception des classes
- 8. Héritage -1
- 9. Héritage -2
- 10. Classes abstraites et interfaces
- 11. Gestion des erreurs
- 12. Conception des applications