

Writing Classes and instanciating Objects in Java with BlueJ



Chapter 2 – Section 1

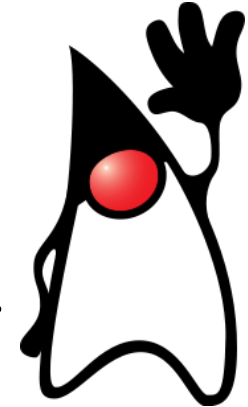


Table of contents

- ☉ The Java language
- ☉ Imperative vs. Object Oriented
- ☉ Java Classes and Objects
- ☉ Another example : the Ball class
- ☉ More about encapsulation
- ☉ A Java program



The Java language



Created by Sun, bought by Oracle

The target

- First: Embedded systems
- Now: Web and others



ORACLE®

Références

- <http://www.java.com/fr/>
- « A White Paper », 1998, Sun Microsystems



Java everywhere

- Web Browsers, Network computer
- WebPhones
- Cellphones
- Credit cards
- Rings





History

- 1991: "Oak", a language for "set-up boxes"

- 1993: Internet becomes more and more important



- 1995: Java 1.0

A Nokia Mediamaster 260 S digital satellite-television set-top box

- "Write Once, Run Anywhere"

- Applets, free



History

- ④ J2SE 1.2 (December 8, 1998)
 - ④ First industrial version, renamed Java 2
- ④ J2SE 1.3 (2000)
- ④ J2SE 1.4 (2002)
- ④ J2SE 5.0 (2004)
- ④ Java SE 6 (2006)
- ④ Java SE 7 (2011)
- ④ Java SE 8 (2014)

Maintained

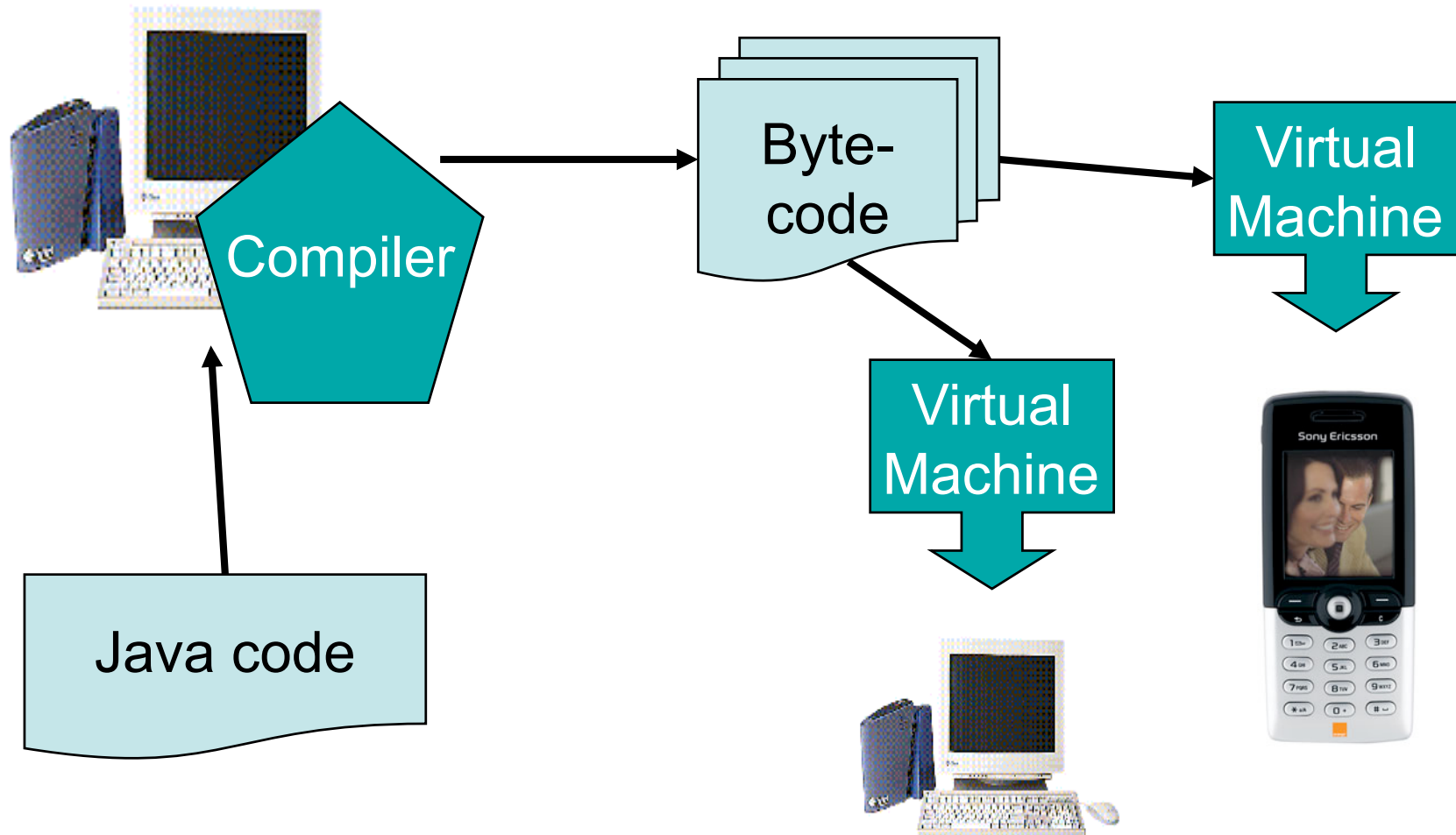


The primary goals of Java

- ① The language should be:
 - ① Simple, Object Oriented, and Familiar
 - ① Robust and Secure
 - ① Architecture neutral and Portable
 - ① Execute with High Performance
 - ① Interpreted, Threaded, and Dynamic



Writing a Java Program





Different platforms

☉ Java SE: Standard Edition (J2SE)

- ☉ For programming portable applications
- ☉ For writing Applets

☉ Java ME: Micro Edition (J2ME)

- ☉ For programming mobile devices and embedded systems

☉ Java EE: Enterprise Edition (J2EE)

- ☉ For server programming (application server, JSP, ...)



Different versions

☉ Software Development Toolkit (SDK)

- ☉ Set of development tools that allows a software engineer to create applications for a certain software package, software framework, hardware platform, computer system, video game console, operating system, or similar platform.
- ☉ Includes a compiler, a debugger, and other tools



Different versions

☉ Software Runtime Environment (JRE)

- ☉ A class loader,
- ☉ A bytecode interpreter,
- ☉ Client and Server virtual machines, optimised for their respective uses
- ☉ Several garbage collectors,
- ☉ A set of supporting runtime libraries.

☉ Named HotSpot

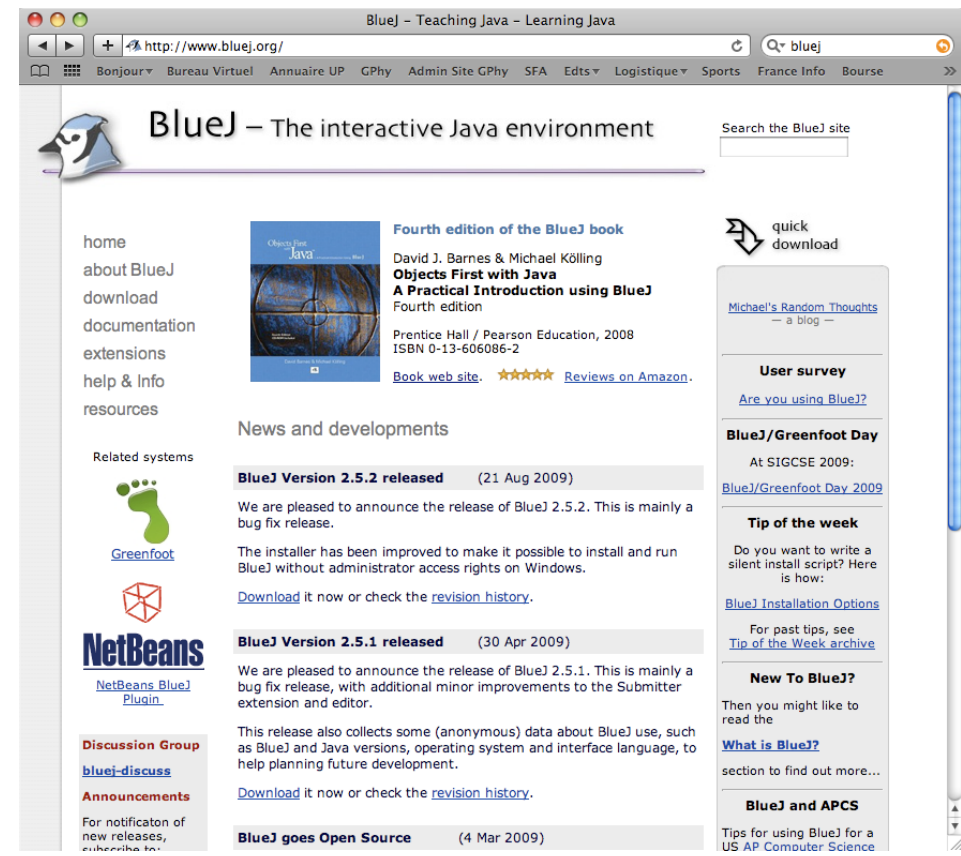
- ☉ Included in JDK



A tool: BlueJ

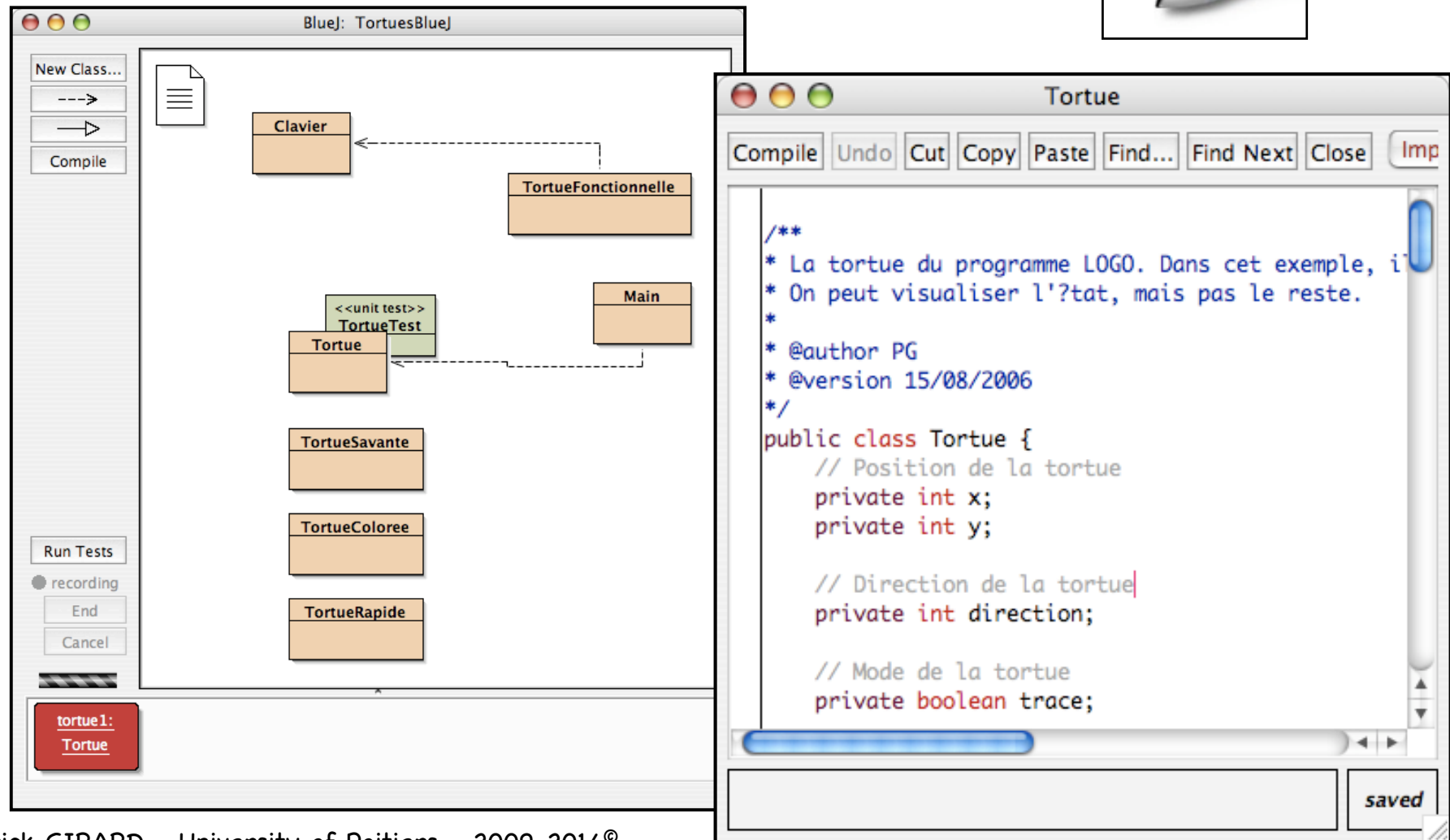


<http://www.bluej.org/>





A tool: BlueJ





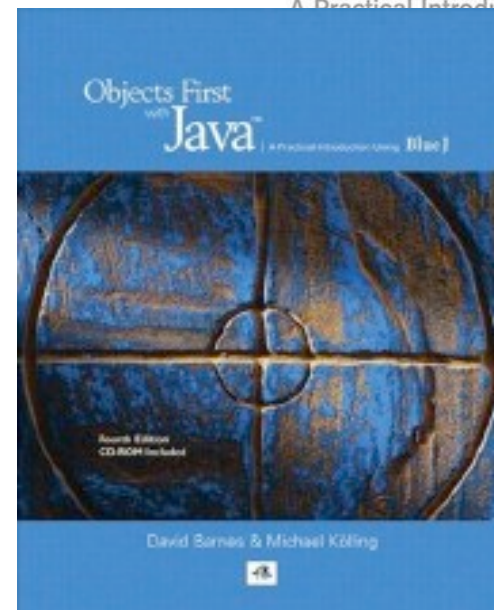
A strong suggestion

- ☉ Buy that book...
- ☉ and use it !
- ☉ Many examples
- ☉ in English



Objects First with Java

A Practical Introduction Using BlueJ



David J. Barnes
Michael Kölling



Other tools

☉ To better understand

- ☉ JELiot, JGrasp

☉ Integrated Development Environments (IDE)

- ☉ Eclipse

- ☉ NetBeans

- ☉ JBuilder...



References

🌐 Site of Java (Sun → Oracle)

- 🌐 <http://www.java.com/fr/>

🌐 Tutorials, examples, forums

- 🌐 <http://www.bluej.org>

- 🌐 <http://java.developpez.com>

- 🌐 fr.comp.lang.java



Java™ 2 Platform
Std. Ed. v1.4.2

[All Classes](#)

Packages
[java.applet](#)
[java.awt](#)
[java.awt.color](#)
[java.awt.datatransfer](#)
[java.awt.dnd](#)
[java.awt.event](#)
[java.awt.font](#)
[java.awt.geom](#)
[java.awt.im](#)
[java.awt.im.spi](#)
[java.awt.image](#)
[java.awt.image.renderable](#)

All Classes
[ARG_IN](#)
[ARG_INOUT](#)
[ARG_OUT](#)
[AWTError](#)
[AWTEvent](#)
[AWTEventListener](#)
[AWTEventListenerProxy](#)
[AWTEventMulticaster](#)
[AWTException](#)
[AWTKeyStroke](#)
[AWTPermission](#)
[AbstractAction](#)
[AbstractBorder](#)
[AbstractButton](#)
[AbstractCellEditor](#)
[AbstractCollection](#)
[AbstractColorChooserPanel](#)
[AbstractDocument](#)
[AbstractDocument.AttributeContext](#)
[AbstractDocument.Content](#)
[AbstractDocument.ElementEdit](#)
[AbstractInterruptibleChannel](#)
[AbstractLayoutCache](#)
[AbstractLayoutCache.NodeDimensions](#)
[AbstractList](#)
[AbstractListModel](#)
[AbstractMap](#)
[AbstractMethodError](#)
[AbstractPreferences](#)
[AbstractSelectableChannel](#)
[AbstractSelectionKey](#)
[AbstractSelector](#)
[AbstractSequentialList](#)
[AbstractSet](#)
[AbstractSpinnerModel](#)
[AbstractTableModel](#)
[AbstractUndoableEdit](#)
[AbstractWriter](#)
[AccessControlContext](#)
[AccessControlException](#)
[AccessController](#)
[AccessException](#)
[Accessible](#)
[AccessibleAction](#)
[AccessibleBundle](#)
[AccessibleComponent](#)

Overview Package Class Use Tree **Deprecated** Index Help
 PREV NEXT FRAMES NO FRAMES

Java™ 2 Platform, Standard Edition, v 1.4.2
API Specification

This document is the API specification for the Java 2 Platform, Standard Edition, version 1.4.2.

See:
[Description](#)

Java 2 Platform Packages

java.applet	Provides the classes necessary to create an applet and the classes an applet uses to communicate with its applet context.
java.awt	Contains all of the classes for creating user interfaces and for painting graphics and images.
java.awt.color	Provides classes for color spaces.
java.awt.datatransfer	Provides interfaces and classes for transferring data between and within applications.
java.awt.dnd	Drag and Drop is a direct manipulation gesture found in many Graphical User Interface systems that provides a mechanism to transfer information between two entities logically associated with presentation elements in the GUI.
java.awt.event	Provides interfaces and classes for dealing with different types of events fired by AWT components.
java.awt.font	Provides classes and interface relating to fonts.
java.awt.geom	Provides the Java 2D classes for defining and performing operations on objects related to two-dimensional geometry.
java.awt.im	Provides classes and interfaces for the input method framework.
java.awt.im.spi	Provides interfaces that enable the development of input methods that can be used with any Java runtime environment.
java.awt.image	Provides classes for creating and modifying images.
java.awt.image.renderable	Provides classes and interfaces for producing rendering-independent images.
java.awt.print	Provides classes and interfaces for a general printing API.
java.beans	Contains classes related to developing <i>beans</i> -- components based on the JavaBeans™ architecture.
java.beans.beancontext	Provides classes and interfaces relating to bean context.
java.io	Provides for system input and output through data streams, serialization and the file system.
java.lang	Provides classes that are fundamental to the design of the Java programming language.
java.lang.ref	Provides reference-object classes, which support a limited degree of interaction with the garbage collector.
java.lang.reflect	Provides classes and interfaces for obtaining reflective information about classes and objects.
java.math	Provides classes for performing arbitrary-precision integer arithmetic (BigInteger) and arbitrary-precision decimal arithmetic (BigDecimal).
java.net	Provides the classes for implementing networking applications.
java.nio	Defines buffers, which are containers for data, and provides an overview of the other NIO packages.
java.nio.channels	Defines channels, which represent connections to entities that are capable of performing I/O operations, such as files and sockets; defines selectors, for multiplexed, non-blocking I/O operations.
java.nio.channels.spi	Service-provider classes for the java.nio.channels package.
java.nio.charset	Defines charsets, decoders, and encoders, for translating between bytes and Unicode characters.
java.nio.charset.spi	Service-provider classes for the java.nio.charset package.
java.rmi	Provides the RMI package.
java.rmi.activation	Provides support for RMI Object Activation.
java.rmi.dgc	Provides classes and interface for RMI distributed garbage-collection (DGC).
java.rmi.registry	Provides a class and two interfaces for the RMI registry.
java.rmi.server	Provides classes and interfaces for supporting the server side of RMI.



Books

