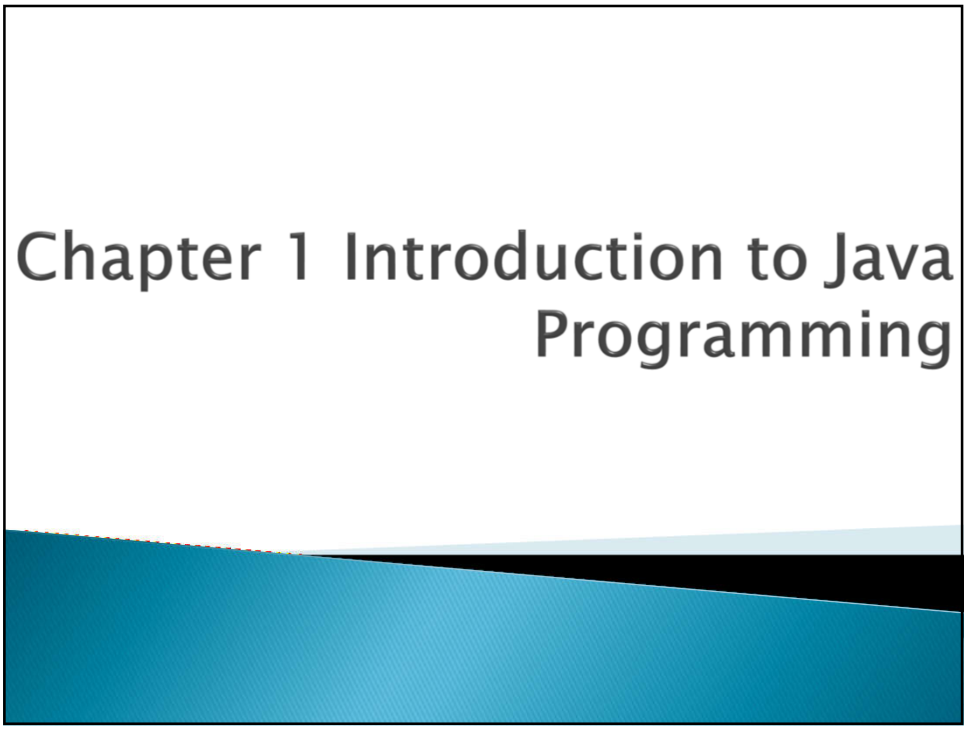
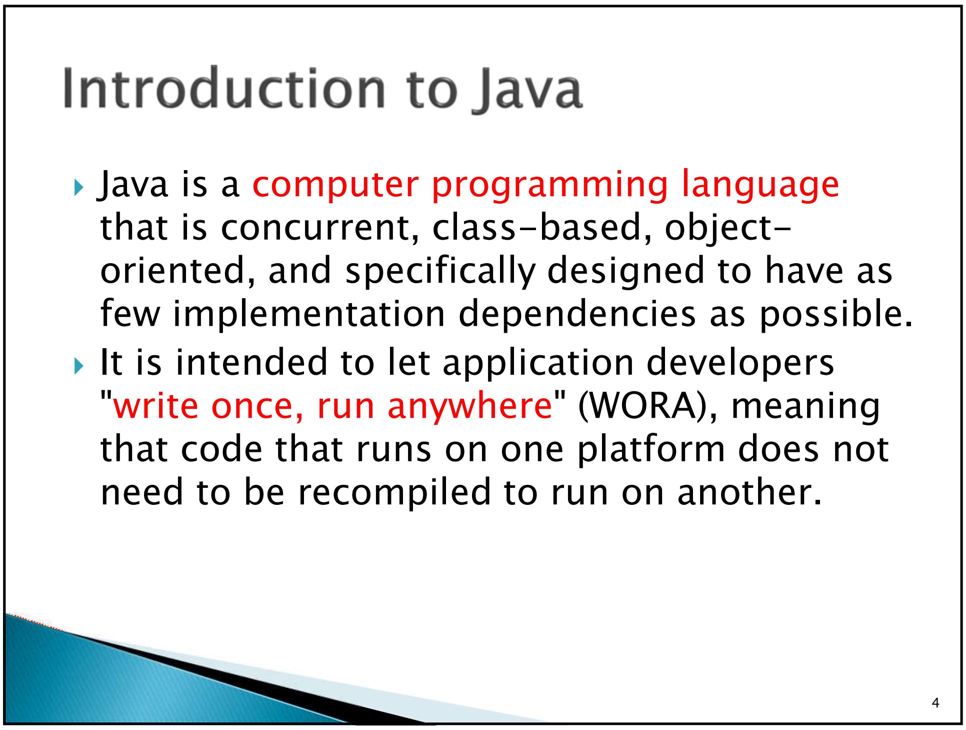
Java Programming Language

Textbook

全国优秀畅销书
Foundations of Programming in Java
清华大学出版社
Ja
入门
A Brain-Friendly Guide
Head First
Java
2nd Edition
Covers Java 5.0
Learn how threads can change your life
Make Java concepts stick to your brain
Fool around in the Java Library
Avoid embarassing OO mistakes
Bend your mind around 42 Java puzzles
Make attractive and useful GUIs
O'REILLY
Kathy Sierra & Bert Bates



Chapter 1 Introduction to Java Programming

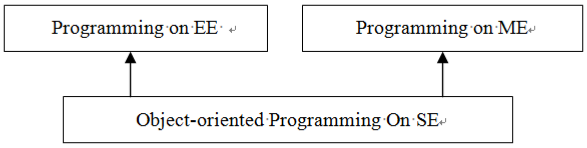


Introduction to Java

- Java is a computer programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible.
- It is intended to let application developers "write once, run anywhere" (WORA), meaning that code that runs on one platform does not need to be recompiled to run on another.

Java Editions

- Java Platform, Micro Edition (Java ME)
  - targeting environments with limited resources.
- Java Platform, Standard Edition (Java SE)
  - targeting workstation environments.
- Java Platform, Enterprise Edition (Java EE)
  - targeting large distributed enterprise or Internet environments.



History of Java

- James Gosling, Mike Sheridan, and Patrick Naughton initiated the Java language project in June 1991
- Java was originally designed for interactive television, but it was too advanced for the digital cable television industry at the time.
- The language was initially called Oak after an oak tree that stood outside Gosling's office; it went by the name Green later, and was later renamed Java, from Java coffee

Principles

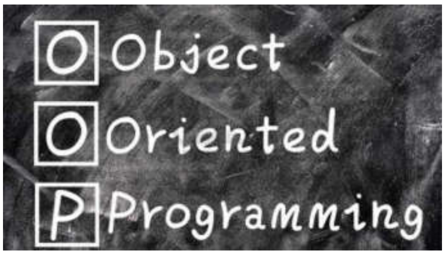
- There were five primary goals in the creation of the Java language:
  - It should be "simple, object-oriented and distributed"
  - It should be "robust and secure"
  - It should be "architecture-neutral and portable"
  - It should execute with "high performance"
  - It should be "interpreted, threaded, and dynamic"

Simple 简单

- The simplicity of Java means that a programmer could learn it quickly.
  - Java has a look and feel familiar to C, C++
  - Java does not support pointers
  - Memory is automatically allocated and deallocation is done by the garbage collector

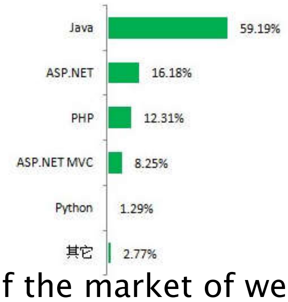
Object-oriented 面向对象

- Java is an Object-Oriented language from the ground up



Distributed 分布性

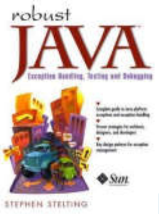
- Java has been designed to support applications on networks
  - It supports different levels of connectivity through classes in the java.net package
  - Java EE supports web applications and distributed systems



share of the market of web applications

Robust 健壮

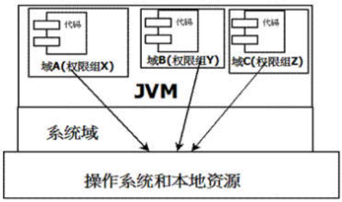
- The explicit use of pointers has been removed from Java
- Java is a strongly typed language and therefore extensive compile-time checking for potential type errors is done
- The exception handling model of Java allows to handle potential errors
- Advanced IDE supported



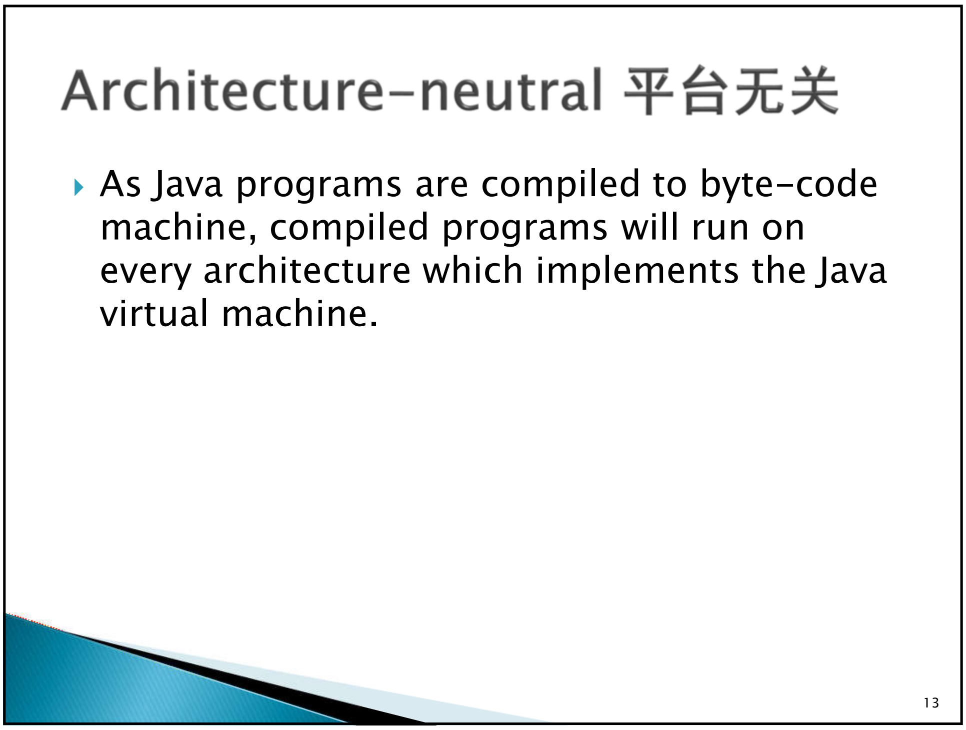
Robust JAVA
——JAVA异常处理、测试与调试

Secure 安全

- Java has no pointers and therefore a program cannot get out of its program segment
- Java compiler does not handle memory layout decisions and one cannot write dynamic code
- untrusted code is placed in a “sandbox” where it can play safely without doing any damage to the “real world” or full Java environment

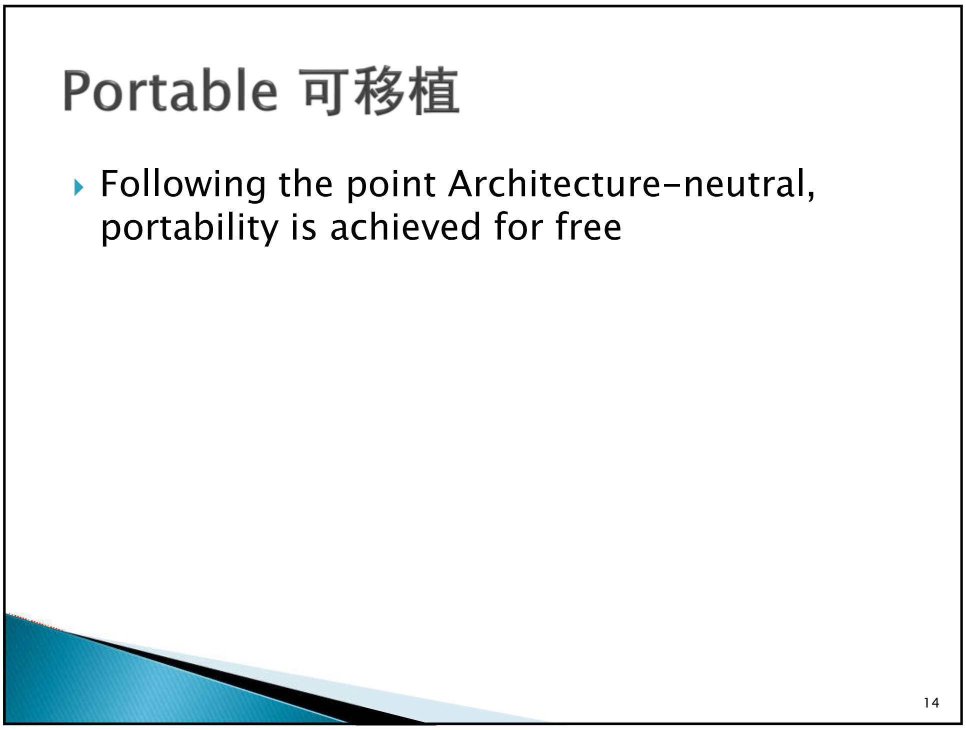


JAVA security model



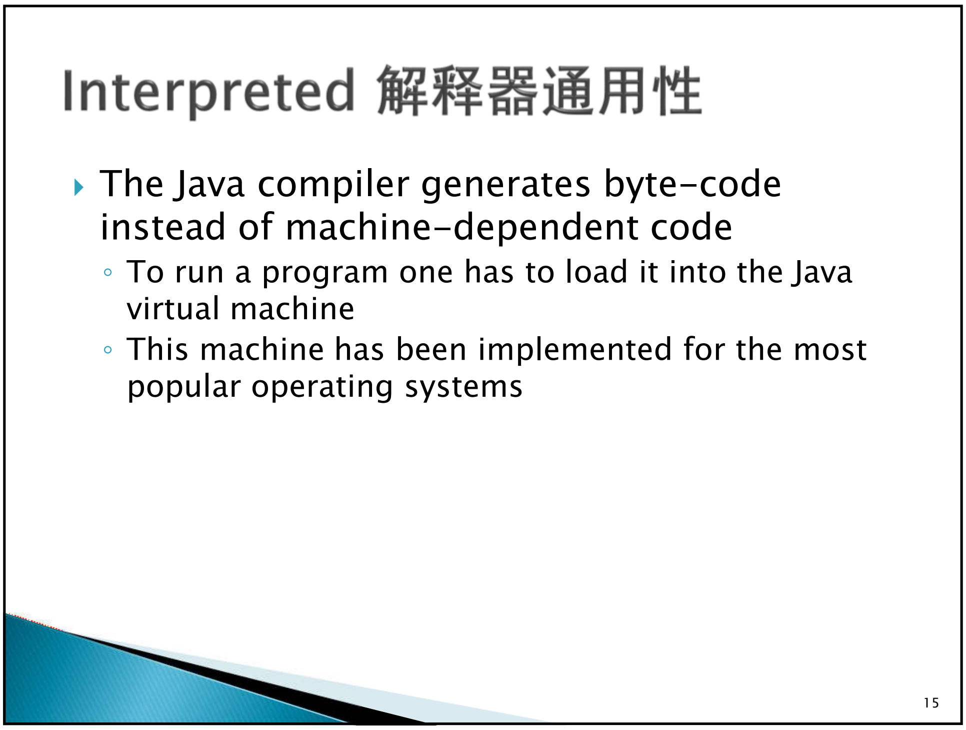
Architecture-neutral 平台无关

- As Java programs are compiled to byte-code machine, compiled programs will run on every architecture which implements the Java virtual machine.



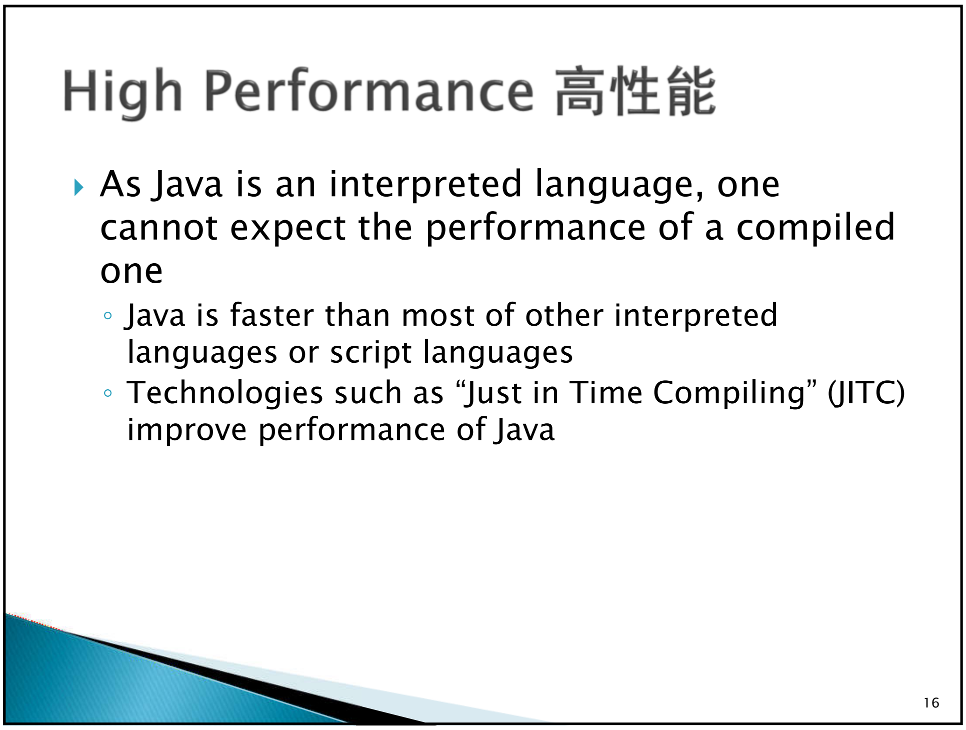
Portable 可移植

- Following the point Architecture-neutral, portability is achieved for free



Interpreted 解释器通用性

- The Java compiler generates byte-code instead of machine-dependent code
  - To run a program one has to load it into the Java virtual machine
  - This machine has been implemented for the most popular operating systems

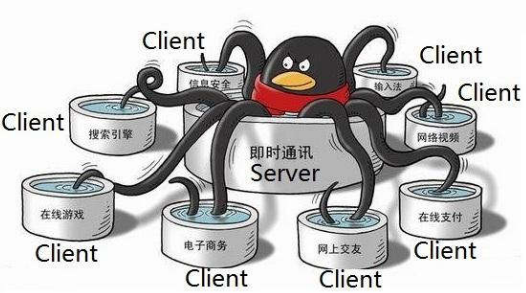


High Performance 高性能

- As Java is an interpreted language, one cannot expect the performance of a compiled one
  - Java is faster than most of other interpreted languages or script languages
  - Technologies such as “Just in Time Compiling” (JITC) improve performance of Java

Threaded 多线程

- Java has built-in constructs for multithreading
  - Java provides *Thread* class and *Runnable* interface to handle thread
  - Java uses *synchronized* keyword to handle concurrent problems



17

Dynamic 动态性

- Java manipulates memory in a dynamic way
  - Classes are loaded by demand even across a network.
  - It is possible for a Java interpreter to download and run code from across a network.

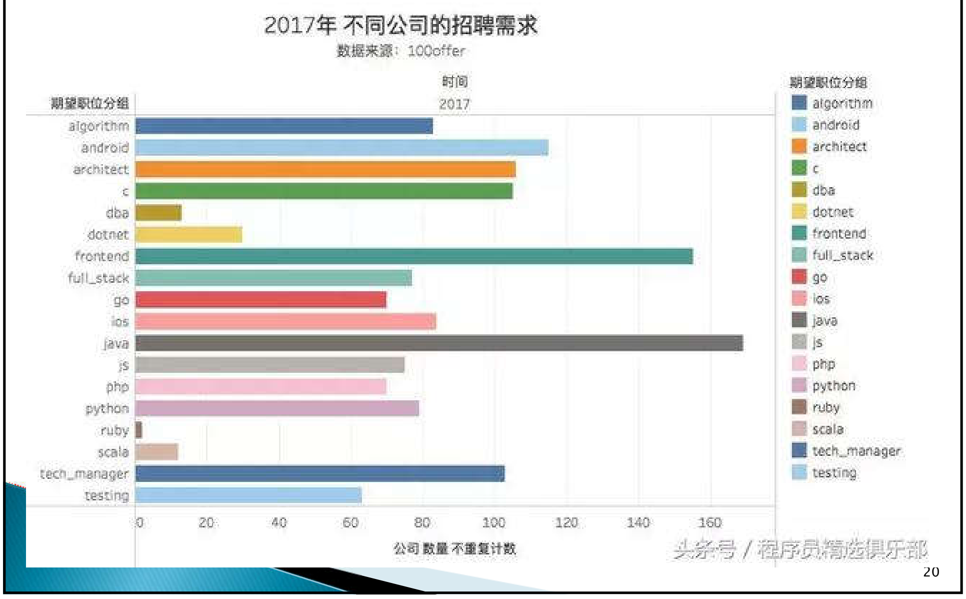
18

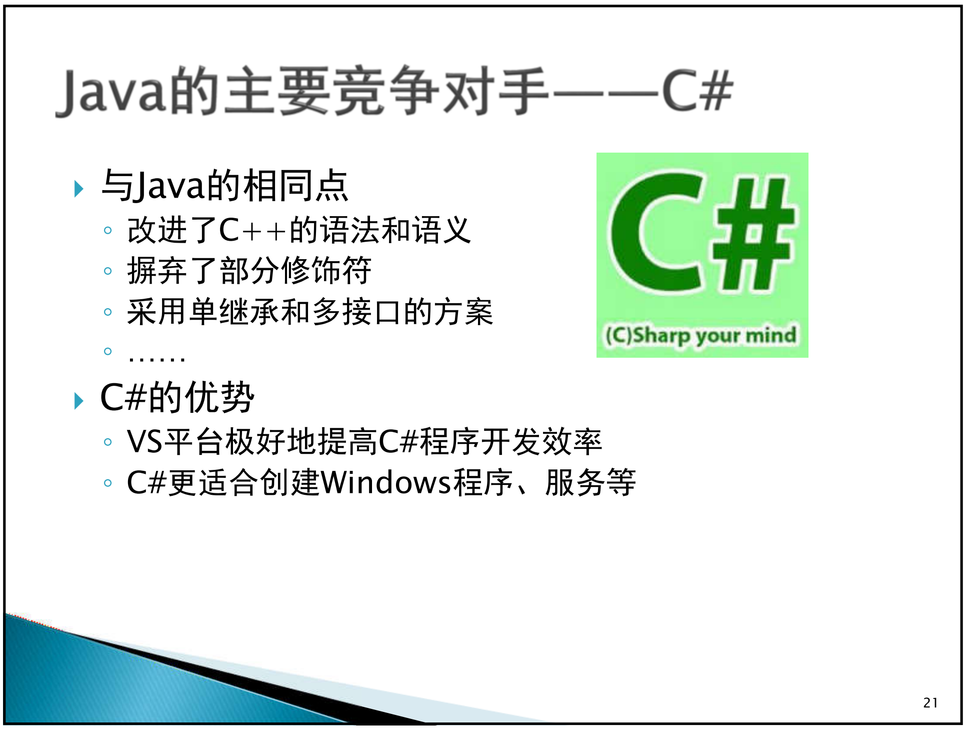
Advantages of Java

- Widely used
- Widely available
- Embraces full set of modern abstractions
- Variety of automatic checks for mistakes in programs
- Java Community Process
- Open code organizations supporting
  - Junit, Tomcat, Struts, Jboss, Eclipse, AJAX, Hibernate…

For Free

Java Economy



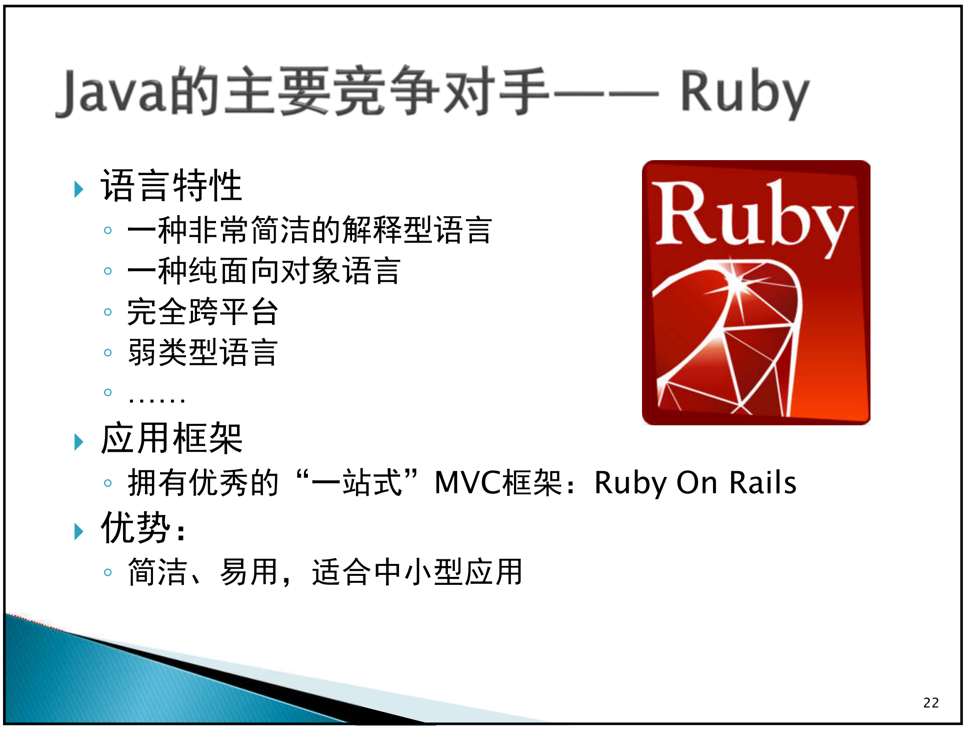


Java的主要竞争对手——C#

- 与Java的相同点
  - 改进了C++的语法和语义
  - 摒弃了部分修饰符
  - 采用单继承和多接口的方案
  - ……
- C#的优势
  - VS平台极好地提高C#程序开发效率
  - C#更适合创建Windows程序、服务等

C#
(C)Sharp your mind

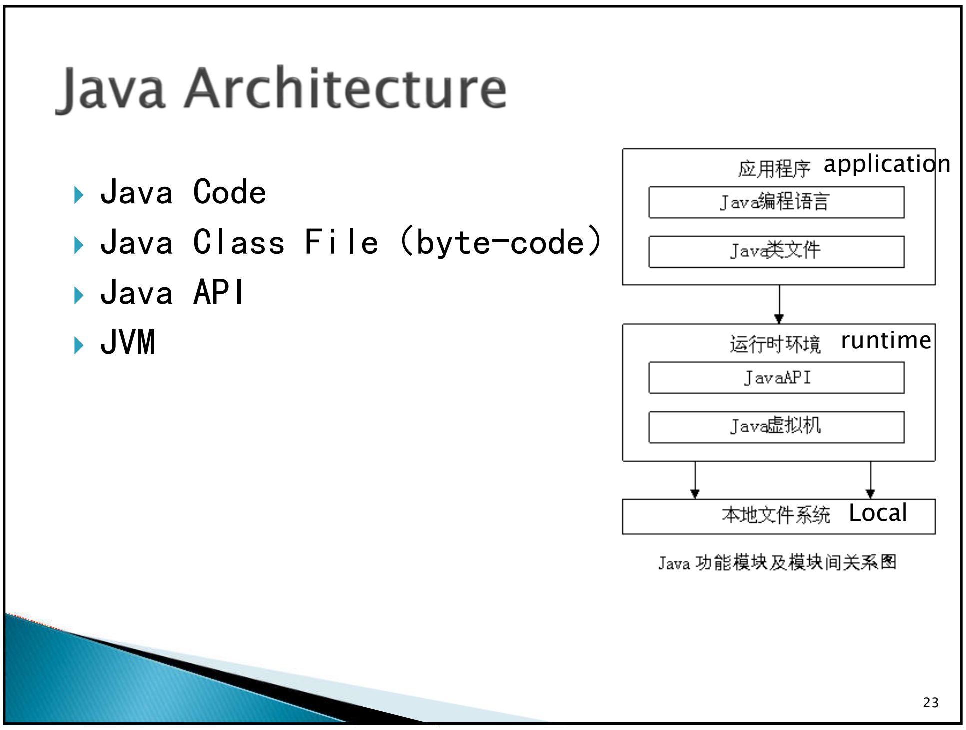
21



Java的主要竞争对手—— Ruby

- 语言特性
  - 一种非常简洁的解释型语言
  - 一种纯面向对象语言
  - 完全跨平台
  - 弱类型语言
  - ……
- 应用框架
  - 拥有优秀的“一站式”MVC框架：Ruby On Rails
- 优势：
  - 简洁、易用，适合中小型应用

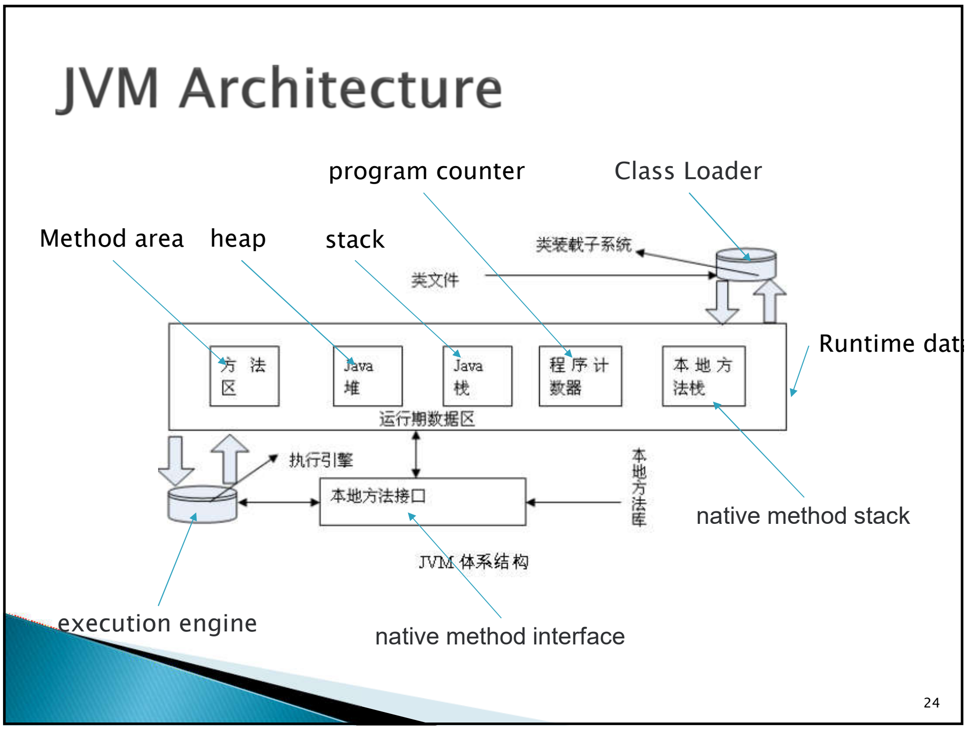
22



Java Architecture

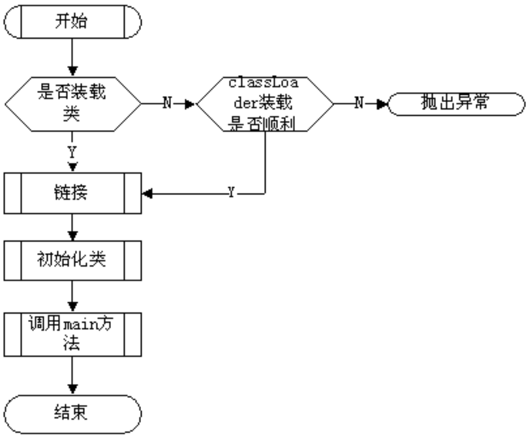
- Java Code
- Java Class File (byte-code)
- Java API
- JVM

Java 功能模块及模块间关系图



JVM Architecture

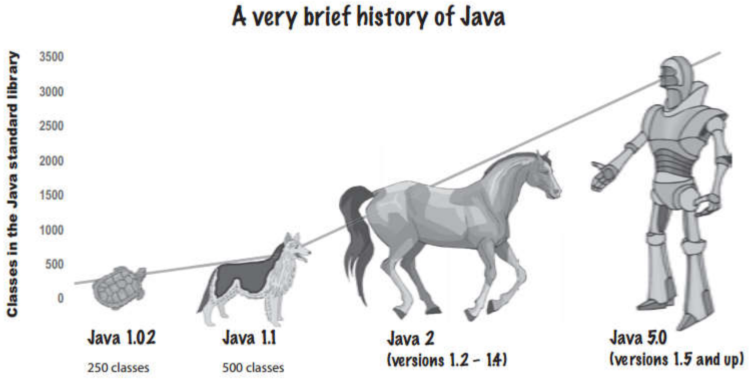
Operating mechanism of JVM



JVM 运行过程

Java API

- Java API also called Java Class Library (JCL) is a set of dynamically loadable libraries that Java applications can call at



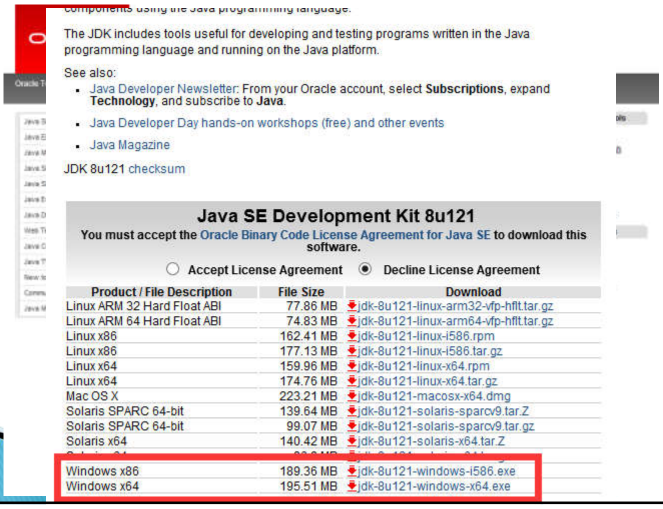
ily

Development Environment and Running Environment

- The Java Development Kit (JDK) consists of the primary programming tools such as a loader, compiler, interpreter and debugger.
- The Java Runtime Environment (JRE) provides the libraries, the Java Virtual Machine, and other components to run applets and applications written in the Java programming language.
- The Java Virtual Machine(JVM) is the code that ultimately runs Java programs by interpreting the intermediate byte-code format of the Java program.

Downloading JDK

http://www.oracle.com/technetwork/java/javase/downloads/index-jsp-138363.html



components using the Java programming language.

The JDK includes tools useful for developing and testing programs written in the Java programming language and running on the Java platform.

See also:

- Java Developer Newsletter: From your Oracle account, select **Subscriptions**, expand **Technology**, and subscribe to **Java**.
- Java Developer Day hands-on workshops (free) and other events
- Java Magazine

JDK 8u121 checksum

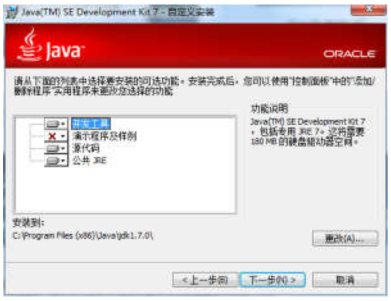
Java SE Development Kit 8u121

You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

○ Accept License Agreement ● Decline License Agreement

Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	77.86 MB	jdk-8u121-linux-arm32-vfp-hflt.tar.gz
Linux ARM 64 Hard Float ABI	74.83 MB	jdk-8u121-linux-arm64-vfp-hflt.tar.gz
Linux x86	162.41 MB	jdk-8u121-linux-i586.rpm
Linux x86	177.13 MB	jdk-8u121-linux-i586.tar.gz
Linux x64	159.96 MB	jdk-8u121-linux-x64.rpm
Linux x64	174.76 MB	jdk-8u121-linux-x64.tar.gz
Mac OS X	223.21 MB	jdk-8u121-macosx-x64.dmg
Solaris SPARC 64-bit	139.64 MB	jdk-8u121-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	99.07 MB	jdk-8u121-solaris-sparcv9.tar.gz
Solaris x64	140.42 MB	jdk-8u121-solaris-x64.tar.Z
Windows x86	189.36 MB	jdk-8u121-windows-i586.exe
Windows x64	195.51 MB	jdk-8u121-windows-x64.exe

Installing JDK and JRE



Java(TM) SE Development Kit 7 - 自定义安装

请从下面的列表中选择要安装的可选功能。安装完成后，您可以使用"控制面板"中的"添加/删除程序"实用程序来更改您选择的功能

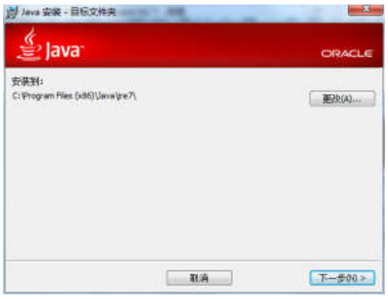
- 开发工具
- 演示程序及样例
- 源代码
- 公共 JRE

功能说明

Java(TM) SE Development Kit 7，包括专用 JRE 7。这将需要 180 MB 的硬盘驱动器空间。

安装到：

C:\Program Files (x86)\Java\jdk1.7.0\

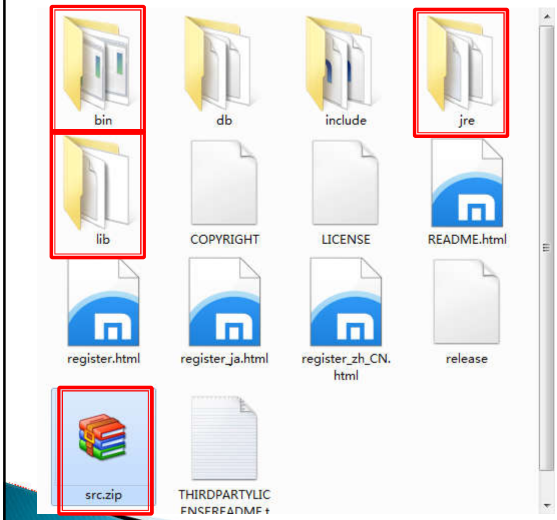


Java 安装 - 目标文件夹

安装到：

C:\Program Files (x86)\Java\jre7\

Contents of JDK

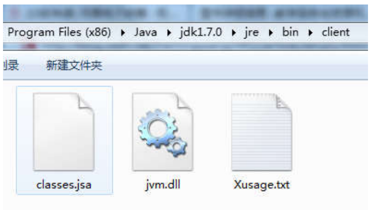


Applications

Java Runtime Enviroment

Class Libraries

Java source code



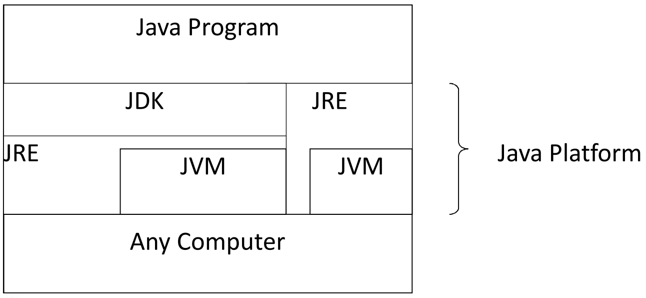
Program Files (x86) ▸ Java ▸ jdk1.7.0 ▸ jre ▸ bin ▸ client

JVM.dll file in JRE folder

JDK>JRE>JVM

Development Environment and Running Environment

- The Java platform consists of a Java virtual machine and all of the class libraries provided in the production environment.



Java Program

JDK

JRE

JRE

JVM

JVM

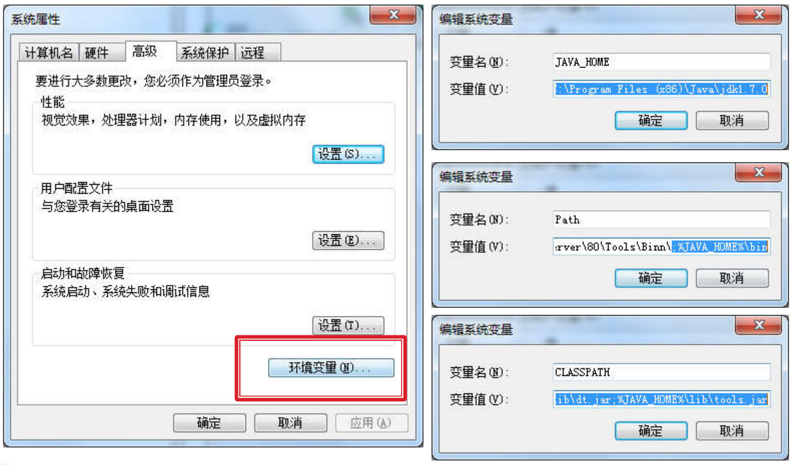
Java Platform

Any Computer

Important tools in bin folder

- javac：Java Compiler，compile java code to byte-code
- java：Java Interpreter，execute byte-code
- jdb：Java Debugger，set breakpoints，execute program step by step
- jar：a compression tool for packing java source to a jar or war file
- Javadoc

Set environment variable



要进行大多数更改，您必须作为管理员登录。

性能
视觉效果，处理器计划，内存使用，以及虚拟内存

用户配置文件
与您登录有关的桌面设置

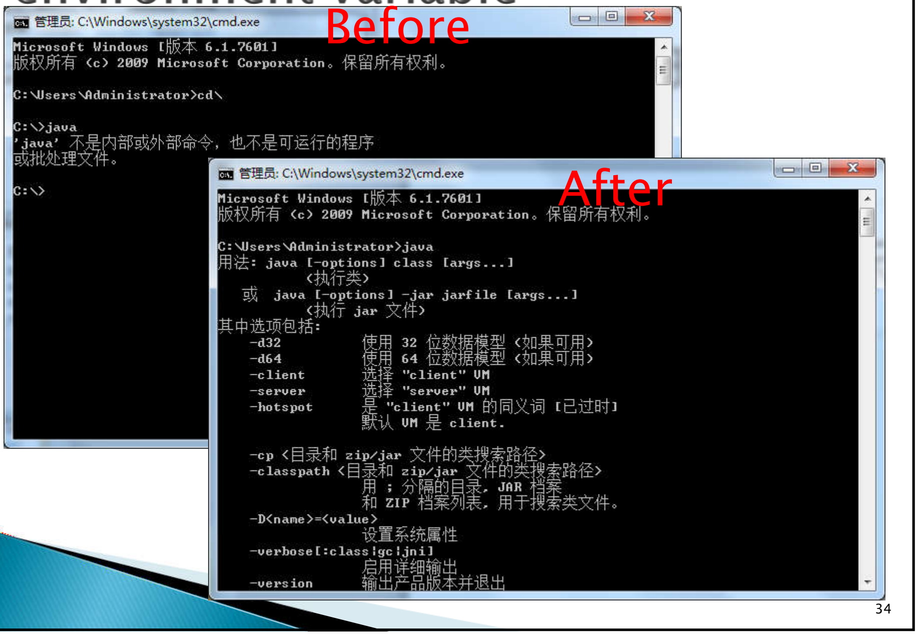
启动和故障恢复
系统启动、系统失败和调试信息

变量名 (N): JAVA_HOME
变量值 (V): :\Program Files (x86)\Java\jdk1.7.0

变量名 (N): Path
变量值 (V): rver\80\Tools\Binn\;%JAVA_HOME%\bin

变量名 (N): CLASSPATH
变量值 (V): ib\dt.jar;%JAVA_HOME%\lib\tools.jar

Why we need to configure environment variable


管理员: C:\Windows\system32\cmd.exe
管理员: C:\Windows\system32\cmd.exe

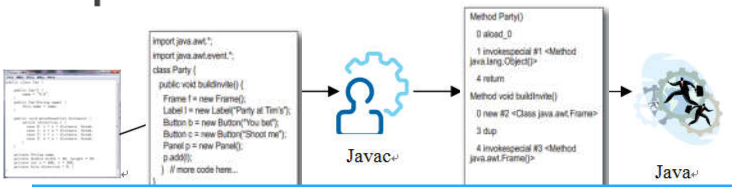
Before

```
Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

C:\Users\Administrator>cd\

C:\>java
'java' 不是内部或外部命令，也不是可运行的程序
或批处理文件。

C:\>
```

After

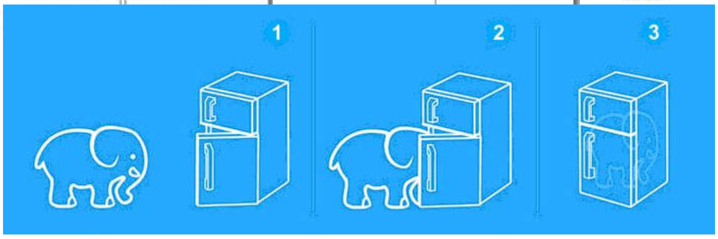
```
Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

C:\Users\Administrator>java
用法: java [-options] class [args...]
           (执行类)
   或  java [-options] -jar jarfile [args...]
           (执行 jar 文件)
其中选项包括:
    -d32          使用 32 位数据模型 (如果可用)
    -d64          使用 64 位数据模型 (如果可用)
    -client       选择 "client" VM
    -server       选择 "server" VM
    -hotspot      是 "client" VM 的同义词 [已过时]
                  默认 VM 是 client.

    -cp <目录和 zip/jar 文件的类搜索路径>
    -classpath <目录和 zip/jar 文件的类搜索路径>
                  用 ; 分隔的目录, JAR 档案
                  和 ZIP 档案列表, 用于搜索类文件。
    -D<name>=<value>
                  设置系统属性
    -verbose[:class|gc|jni]
                  启用详细输出
    -version      输出产品版本并退出
```

1.3 Programming in the Command Prompt Window



```
import java.awt.*;
import java.awt.event.*;
class Party {
  public void buildInvite() {
    Frame f = new Frame();
    Label l = new Label("Party at Tim's");
    Button b = new Button("You bet");
    Button c = new Button("Shoot me");
    Panel p = new Panel();
    p.add(l);
  } // more code here...
}
```

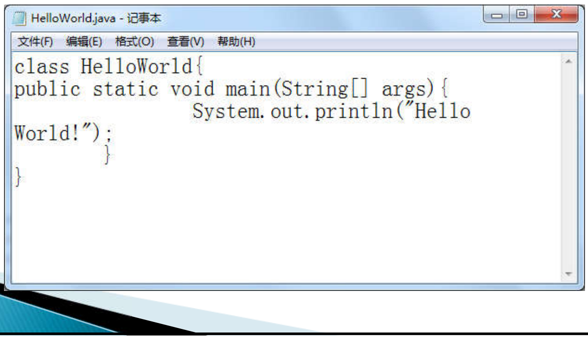
```
Method Party()
  0 aload_0
  1 invokespecial #1 <Method java.lang.Object()>
  4 return
Method void buildInvite()
  0 new #2 <Class java.awt.Frame>
  3 dup
  4 invokespecial #3 <Method java.awt.Frame()>
```



Edit
Compile
Run

Editing Java Code

- Create the program by typing it into a text editor, and save it as HelloWorld.java

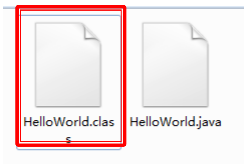


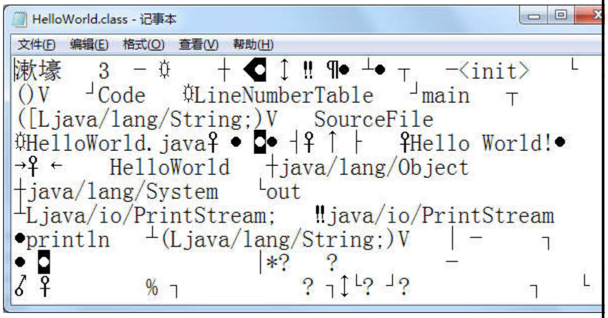
```
class HelloWorld{
public static void main(String[] args){
                  System.out.println("Hello
World!");
         }
}
```

Compiling Java Source

- Compile it by typing at the command-line:

```
javac HelloWorld.java
```



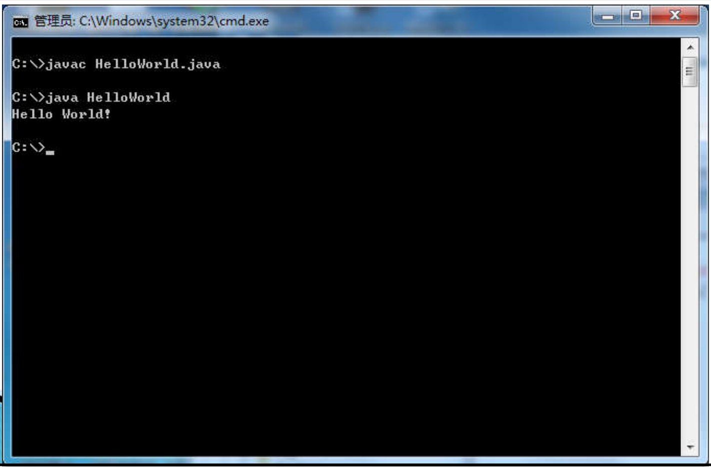

HelloWorld.class - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```
漱壕   3  ─ ☼    ┼ ◘ ↕ ‼ ¶• ┴• ┬   ─<init>     └
()V   ┘Code   ☼LineNumberTable   ┘main   ┬
([Ljava/lang/String;)V   SourceFile
☼HelloWorld.java♀ • ◘• ┤♀ ↑ ├   ♀Hello World!•
→♀ ←     HelloWorld   ┼java/lang/Object
┼java/lang/System    └out
┴Ljava/io/PrintStream;   ‼java/io/PrintStream
•println   ┴(Ljava/lang/String;)V   │ ─      ┐
• ◘                  │*?   ?          ─
♂ ♀         % ┐          ? ┐↕└? ┘?          ┐    └
```

running Java program

- Execute it by typing at the command-line:

```
java HelloWorld
```



```
C:\>javac HelloWorld.java

C:\>java HelloWorld
Hello World!

C:\>_
```

Using IDE (Integrated Development Environment)

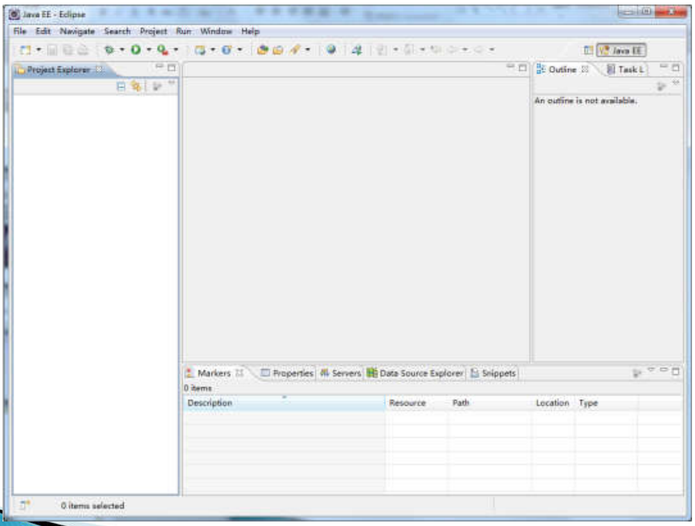
- Common Java IDEs
  - Netbeans
  - Intellij Idea
  - Eclipse

Eclipse

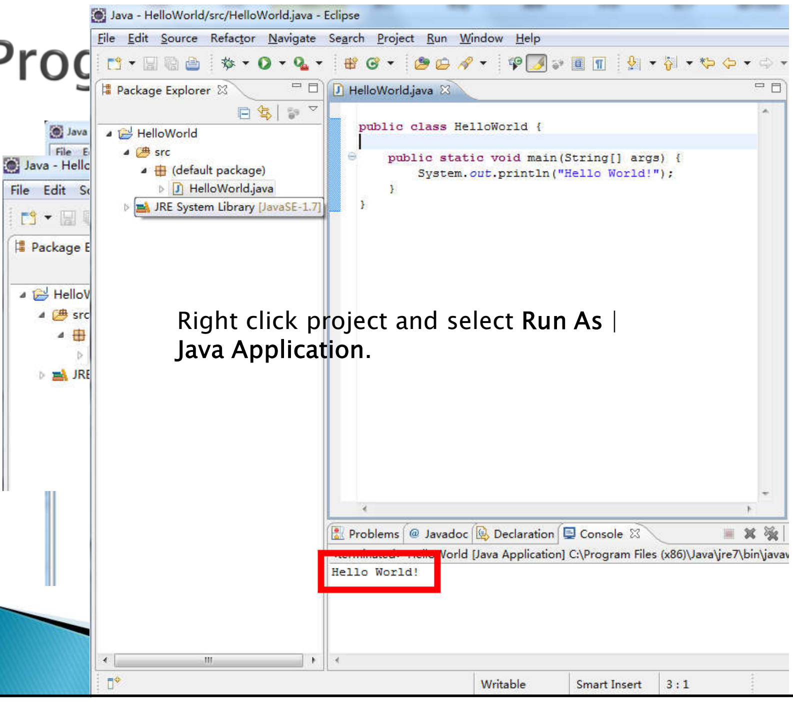
- The Eclipse IDE is an open-source integrated development environment. Eclipse enables developers to rapidly create web, enterprise, desktop, and mobile applications using the Java platform, as well as JavaFX, PHP, JavaScript and Ajax, Ruby and Ruby on Rails, Groovy and Grails, and C/C++.

eclipse
HELIOS
(c) Copyright Eclipse contributors and others, 2000, 2011. All rights reserved. Eclipse is a trademark of the Eclipse Foundation,Inc. Oracle and Java are registered trademarks of Oracle and/or itsaffiliates.Other names may be trademarks of their respective owners.

Programming in Eclipse



Prog

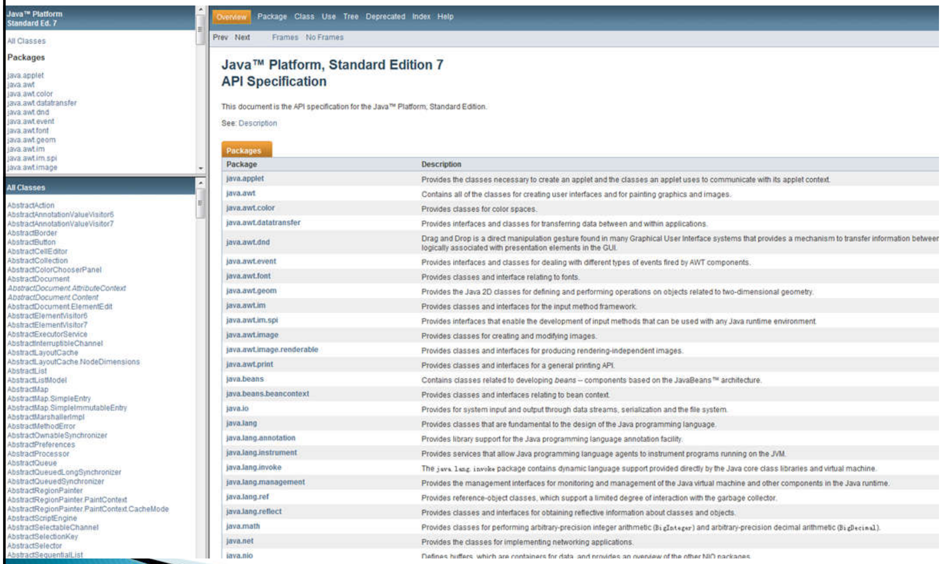

Java - HelloWorld/src/HelloWorld.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
HelloWorld
src
(default package)
HelloWorld.java
JRE System Library [JavaSE-1.7]
HelloWorld.java
Problems
Javadoc
Declaration
Console
Writable
Smart Insert
3 : 1

```
public class HelloWorld {

    public static void main(String[] args) {
        System.out.println("Hello World!");
    }
}
```

Right click project and select **Run As** | **Java Application.**

```
Vorld [Java Application] C:\Program Files (x86)\Java\jre7\bin\javav
Hello World!
```

Java API Documentation API文档



Java™ Platform
Standard Ed. 7

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

All Classes

AbstractAction
AbstractAnnotationValueVisitor6
AbstractAnnotationValueVisitor7
AbstractBorder
AbstractButton
AbstractCellEditor
AbstractCollection
AbstractColorChooserPanel
AbstractDocument
AbstractDocument.AttributeContext
AbstractDocument.Content
AbstractDocument.ElementEdit
AbstractElementVisitor6
AbstractElementVisitor7
AbstractExecutorService
AbstractInterruptibleChannel
AbstractLayoutCache
AbstractLayoutCache.NodeDimensions
AbstractList
AbstractListModel
AbstractMap
AbstractMap.SimpleEntry
AbstractMap.SimpleImmutableEntry
AbstractMarshallerImpl
AbstractMethodError
AbstractOwnableSynchronizer
AbstractPreferences
AbstractProcessor
AbstractQueue
AbstractQueuedLongSynchronizer
AbstractQueuedSynchronizer
AbstractRegionPainter
AbstractRegionPainter.PaintContext
AbstractRegionPainter.PaintContext.CacheMode
AbstractScriptEngine
AbstractSelectableChannel
AbstractSelectionKey
AbstractSelector
AbstractSequentialList

Java™ Platform, Standard Edition 7
API Specification

This document is the API specification for the Java™ Platform, Standard Edition.

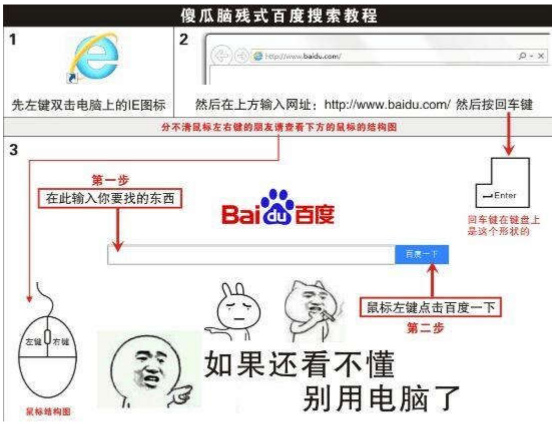
See: Description

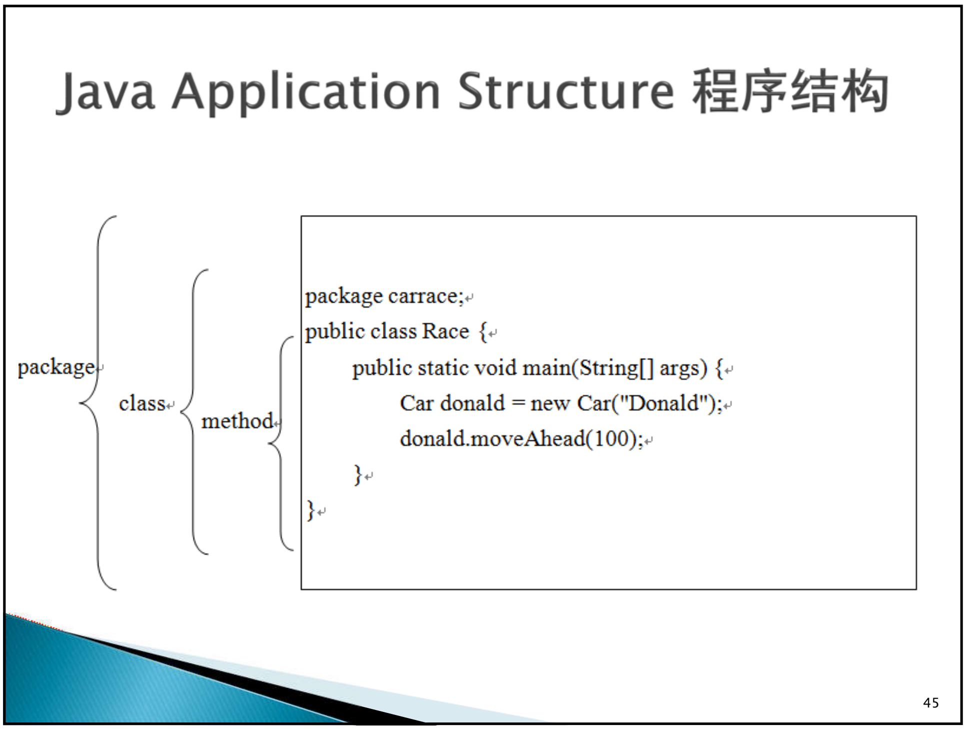
Packages

Package	Description
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 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 *beans* -- 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.annotation	Provides library support for the Java programming language annotation facility.
java.lang.instrument	Provides services that allow Java programming language agents to instrument programs running on the JVM.
java.lang.invoke	The `java.lang.invoke` package contains dynamic language support provided directly by the Java core class libraries and virtual machine.
java.lang.management	Provides the management interfaces for monitoring and management of the Java virtual machine and other components in the Java runtime.
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.

Related Websites 相关网站

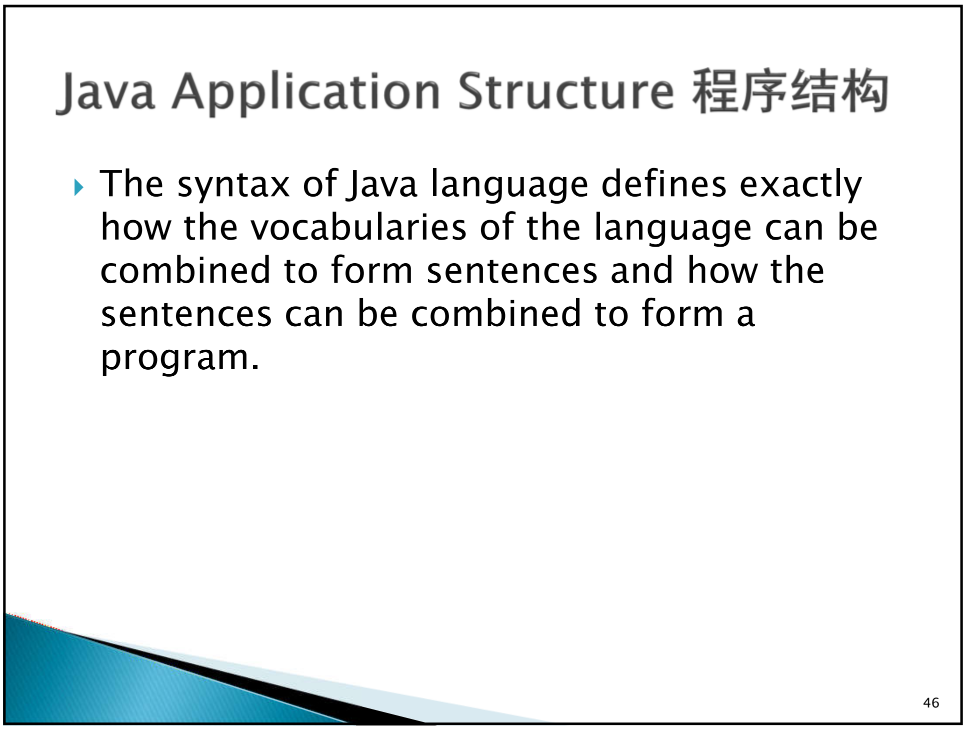
- http://www.oracle.com/technetwork/java/index.html
- http://jakarta.apache.org/





Java Application Structure 程序结构

```
package carrace;
public class Race  {
      public static void main(String[] args) {
            Car donald = new Car("Donald");
            donald.moveAhead(100);
      }
}
```



Java Application Structure 程序结构

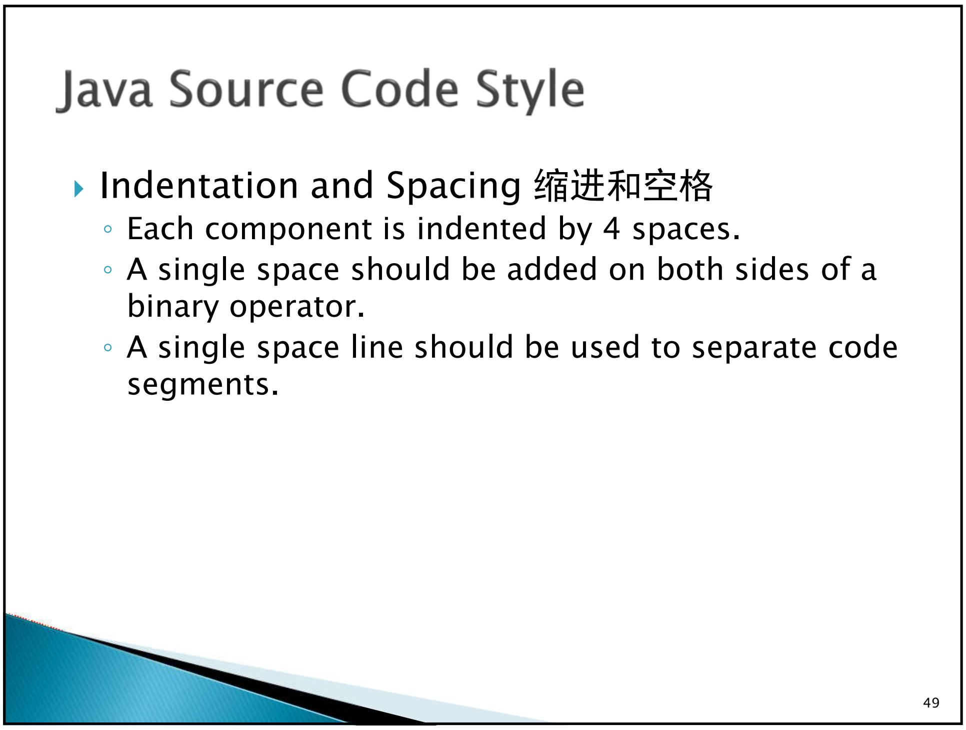
- The syntax of Java language defines exactly how the vocabularies of the language can be combined to form sentences and how the sentences can be combined to form a program.

Java Application Structure 程序结构

- compile-time error (编译时错误)
  - Any error identified by the compiler is referred to as a compile-time error
- run-time error (运行时错误)
  - The error produced by the interpreter, java, is called a run-time error.
- logical error (逻辑错误)
  - If your program compiles and executes without any complaint, but it produces wrong result, there must be some logical errors in your program.

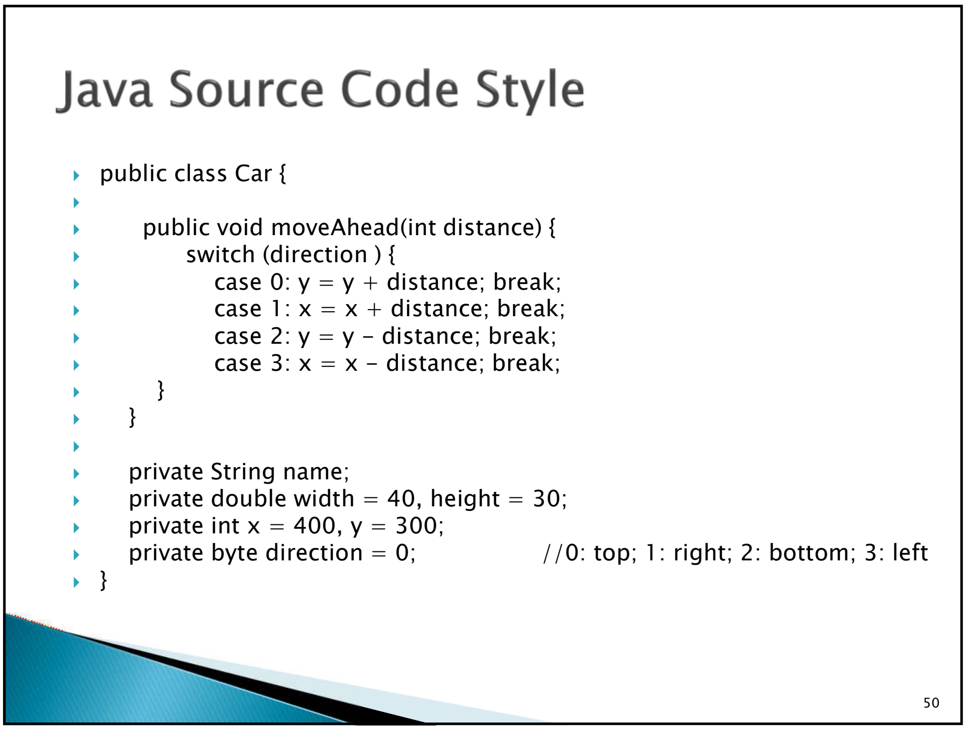
Java Source Code Style 代码风格

- Programming style and documentation are as important as coding
- Several guidelines for good style and documentation



Java Source Code Style

- Indentation and Spacing 缩进和空格
  - Each component is indented by 4 spaces.
  - A single space should be added on both sides of a binary operator.
  - A single space line should be used to separate code segments.



Java Source Code Style

```
public class Car {

    public void moveAhead(int distance) {
        switch (direction ) {
            case 0: y = y + distance; break;
            case 1: x = x + distance; break;
            case 2: y = y - distance; break;
            case 3: x = x - distance; break;
        }
    }

    private String name;
    private double width = 40, height = 30;
    private int x = 400, y = 300;
    private byte direction = 0;          //0: top; 1: right; 2: bottom; 3: left
}
```

Java Source Code Style

- **Block Styles**
  - next-line style

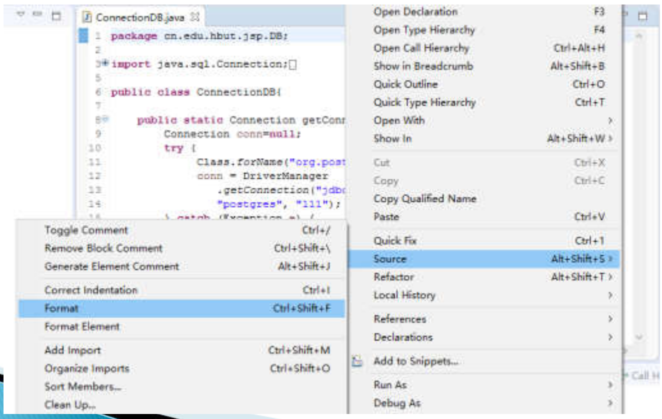
```
class A
{
    void aMethod()
    {
        //Do something
    }
}
```

  - end-of-line style

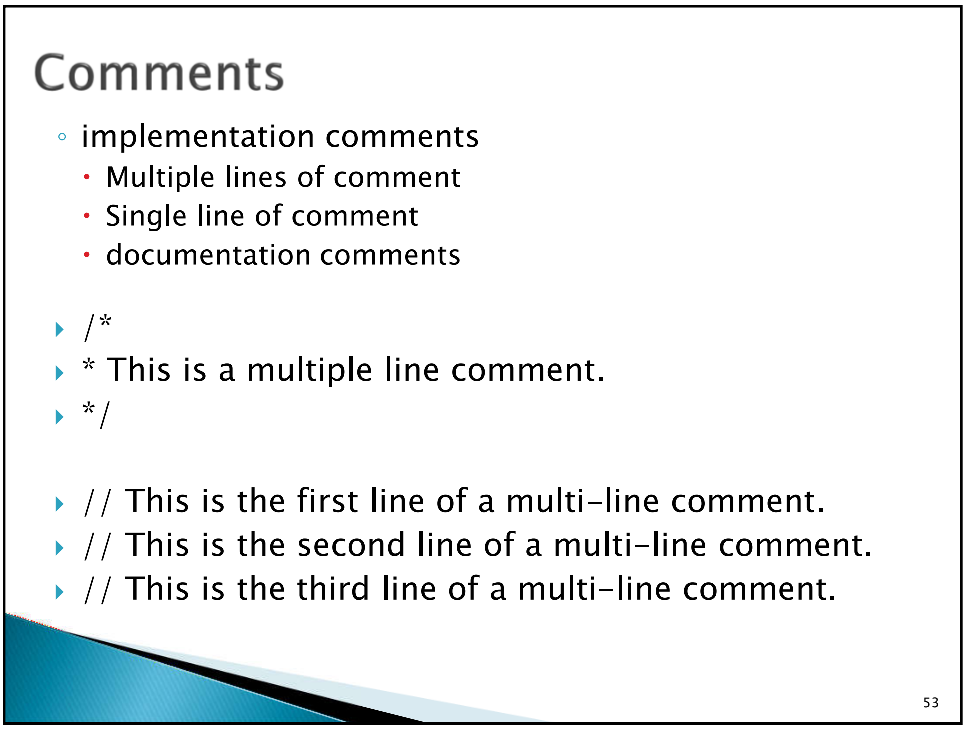
```
class A {
    void aMethod() {
        //Do something
    }
}
```

1.6 Java Source Code Style

- Choose **Source** | **Format**, or use Ctrl+Shift+F command, to format the entire file or a selection of code.



```
package cn.edu.hbut.jsp.DB;

import java.sql.Connection;

public class ConnectionDB{

    public static Connection getConn
        Connection conn=null;
        try {
            Class.forName("org.post
            conn = DriverManager
                .getConnection("jdbc
                "postgres", "111");
```



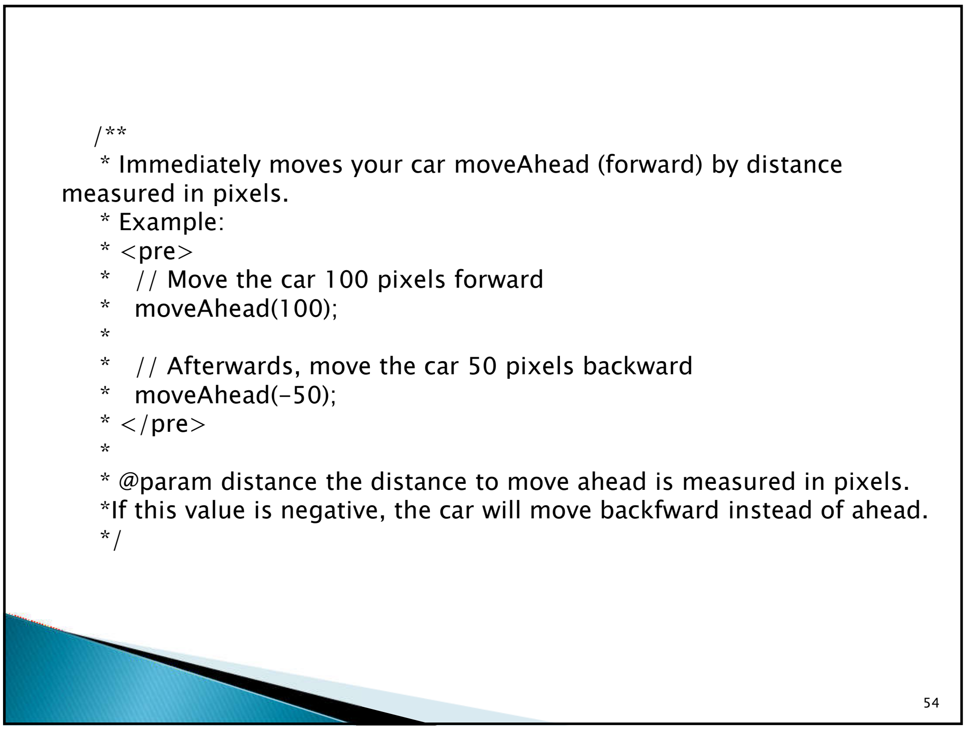
Comments

- implementation comments
  - Multiple lines of comment
  - Single line of comment
  - documentation comments

```
/*
* This is a multiple line comment.
*/
```

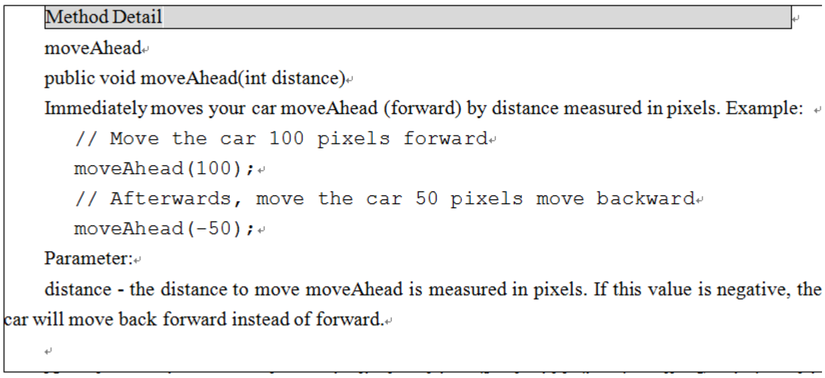
```
// This is the first line of a multi-line comment.
// This is the second line of a multi-line comment.
// This is the third line of a multi-line comment.
```

53



```
    /**
     * Immediately moves your car moveAhead (forward) by distance
measured in pixels.
     * Example:
     * <pre>
     *   // Move the car 100 pixels forward
     *   moveAhead(100);
     *
     *   // Afterwards, move the car 50 pixels backward
     *   moveAhead(-50);
     * </pre>
     *
     * @param distance the distance to move ahead is measured in pixels.
     *If this value is negative, the car will move backfward instead of ahead.
     */
```

54



Method Detail

moveAhead

public void moveAhead(int distance)

Immediately moves your car moveAhead (forward) by distance measured in pixels. Example:

```
// Move the car 100 pixels forward
moveAhead(100);
// Afterwards, move the car 50 pixels move backward
moveAhead(-50);
```

Parameter:

distance - the distance to move moveAhead is measured in pixels. If this value is negative, the car will move back forward instead of forward.

Java and Development Tools

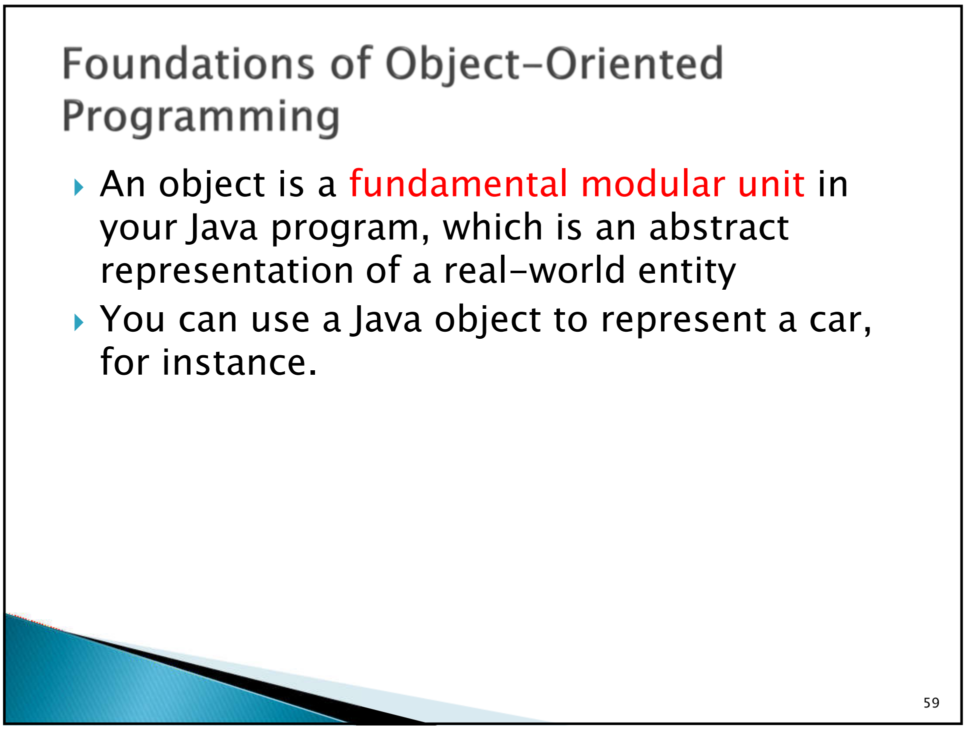
- Java generally refers to a combination of two things:
  - the object-oriented programming language;
  - the Java platform

Foundations of Object-Oriented Programming

- Java is an object-oriented language.
- An object in a computer models the real-world objects found in everyday life.

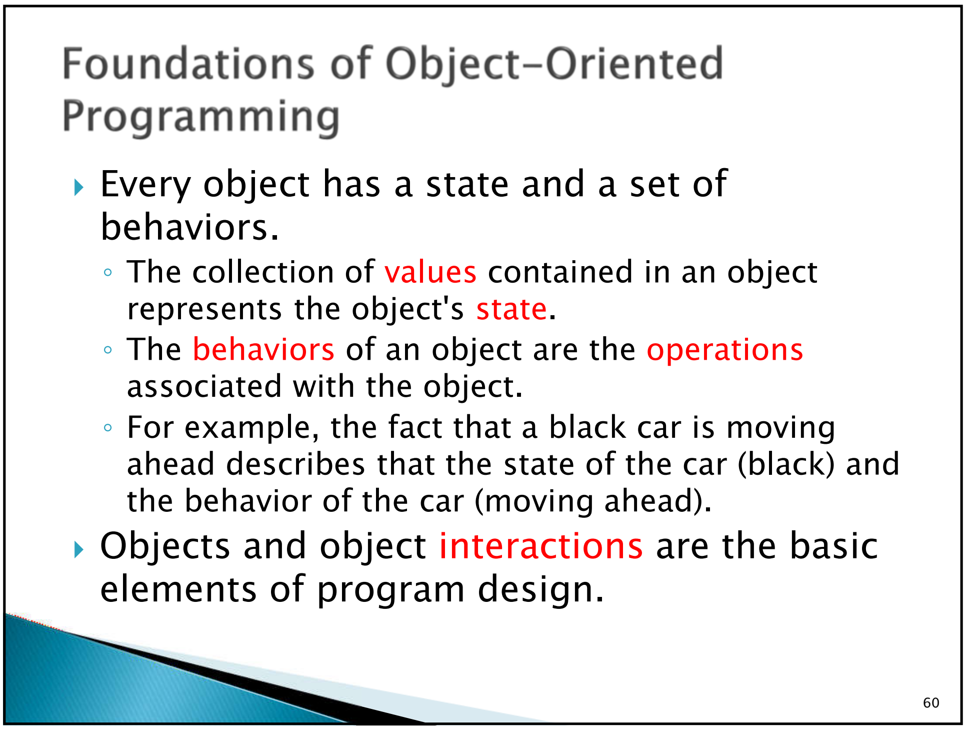
Foundations of Object-Oriented Programming

- Merriam Webster online dictionary, an object is:
  - some material that may be perceived by the senses
  - something that when viewed stirs a particular emotion (as pity)物体
  - something mental or physical toward which thought, feeling, or action is directed 情感
  - something physical that is perceived by an individual and becomes an agent for psychological identification规律
  - the goal or end of an effort or activity : purpose, objective目标
  - a cause for attention or concern 原因
  - a thing that forms an element of or constitutes the subject matter of an investigation or science原理要素



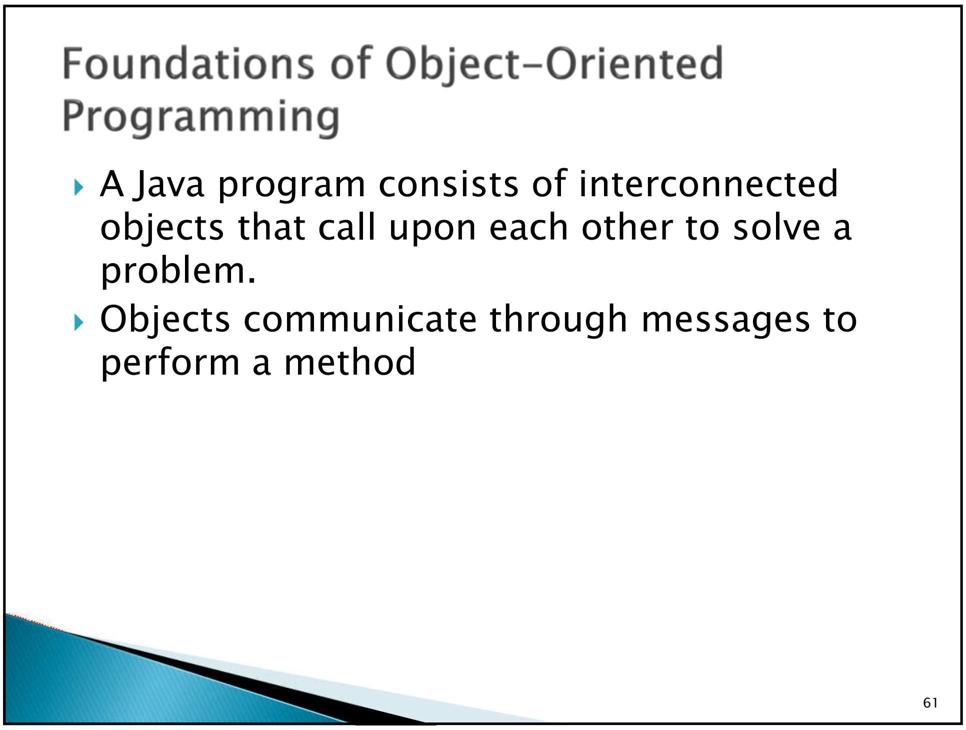
Foundations of Object-Oriented Programming

- An object is a fundamental modular unit in your Java program, which is an abstract representation of a real-world entity
- You can use a Java object to represent a car, for instance.



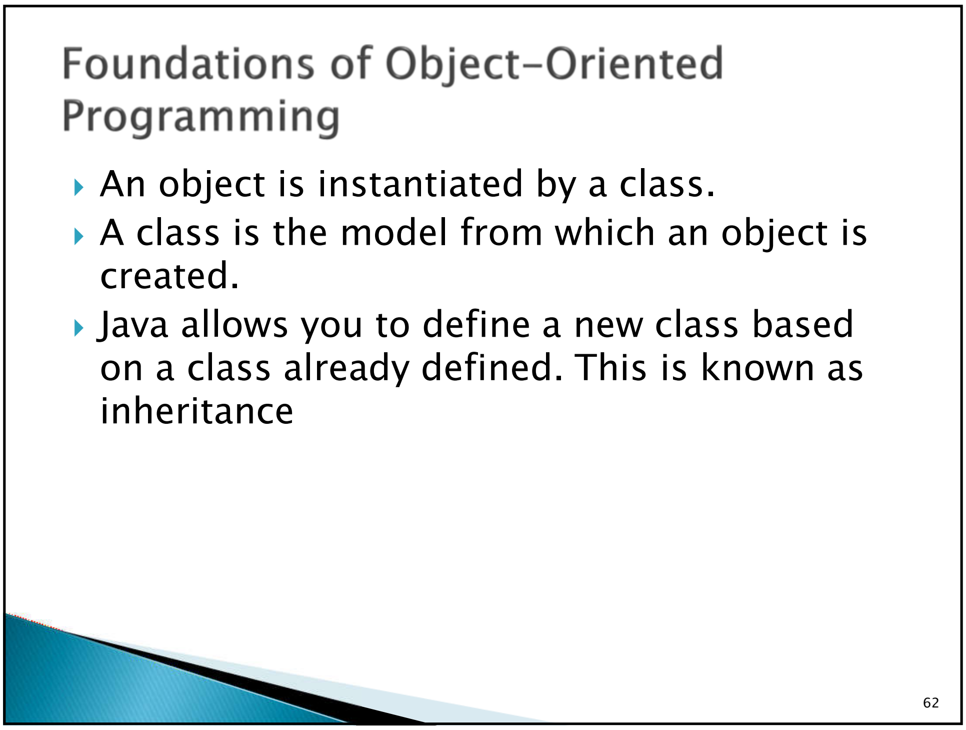
Foundations of Object-Oriented Programming

- Every object has a state and a set of behaviors.
  - The collection of values contained in an object represents the object's state.
  - The behaviors of an object are the operations associated with the object.
  - For example, the fact that a black car is moving ahead describes that the state of the car (black) and the behavior of the car (moving ahead).
- Objects and object interactions are the basic elements of program design.



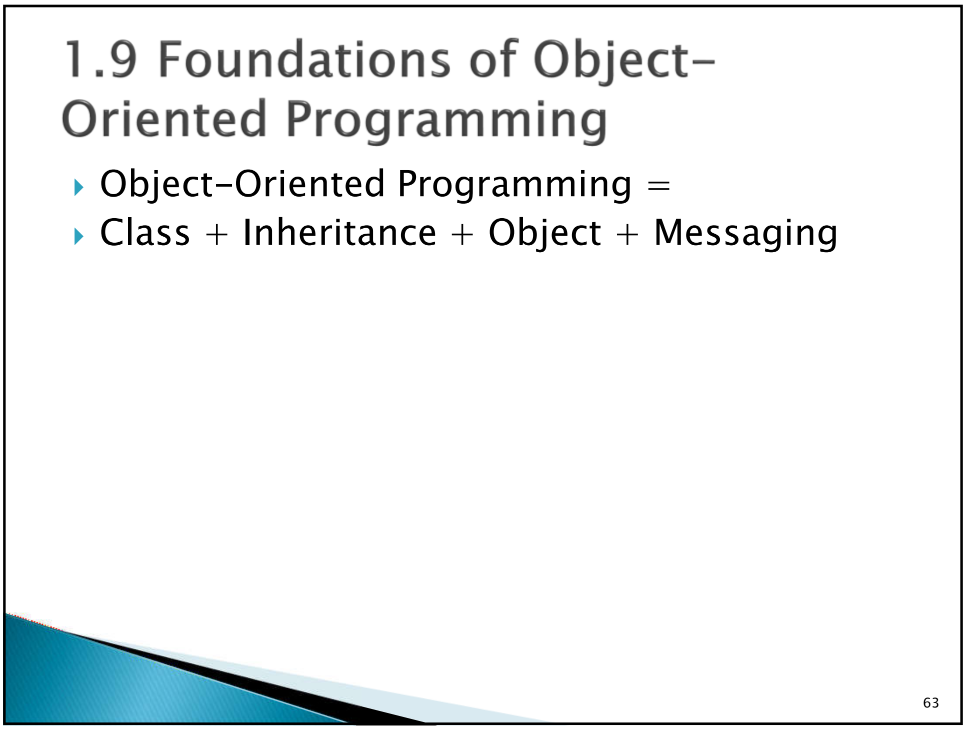
Foundations of Object-Oriented Programming

- A Java program consists of interconnected objects that call upon each other to solve a problem.
- Objects communicate through messages to perform a method



Foundations of Object-Oriented Programming

- An object is instantiated by a class.
- A class is the model from which an object is created.
- Java allows you to define a new class based on a class already defined. This is known as inheritance



1.9 Foundations of Object-Oriented Programming

- Object-Oriented Programming =
- Class + Inheritance + Object + Messaging