
Java Language Introduction

Yang Wu

2004-07



Content

⦿ **Hello, World!**



⦿ **Environment**



⦿ **Language Syntax**



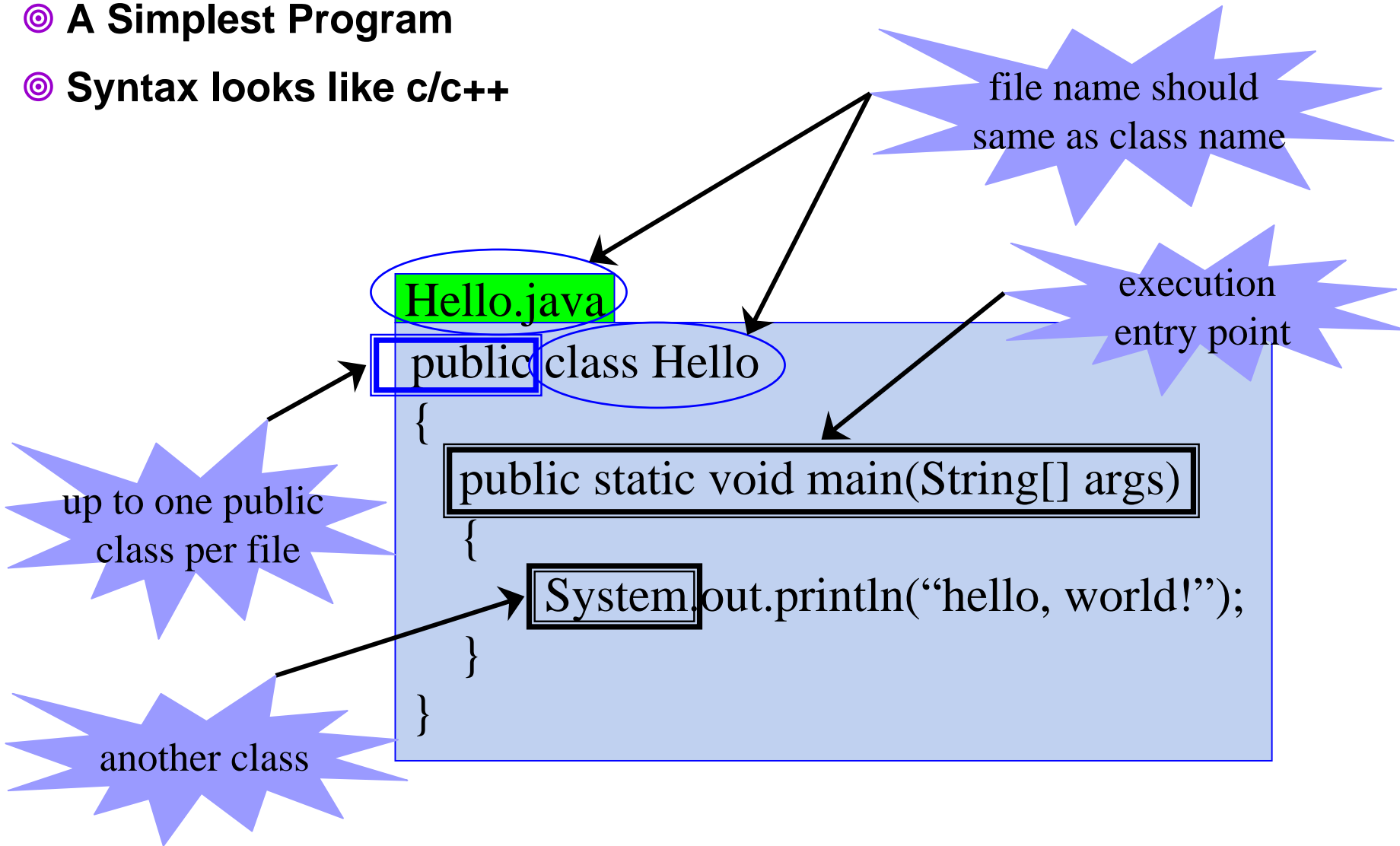
⦿ **Common Task**



Hello, World!

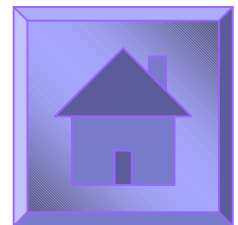
◎ A Simplest Program

◎ Syntax looks like c/c++



Module Q&A

- ⦿ <http://java.sun.com>
- ⦿ [http://java.sun.com/j2se/doc/
⦿ {version}/api/index.html](http://java.sun.com/j2se/doc/{version}/api/index.html)



Environment

- ◎ Programming cycle
- ◎ J2SE SDK Files
- ◎ JRE_HOME
- ◎ JAVA_HOME
- ◎ jvm
- ◎ jar
- ◎ Practice



Programming Cycle

◎ Coding

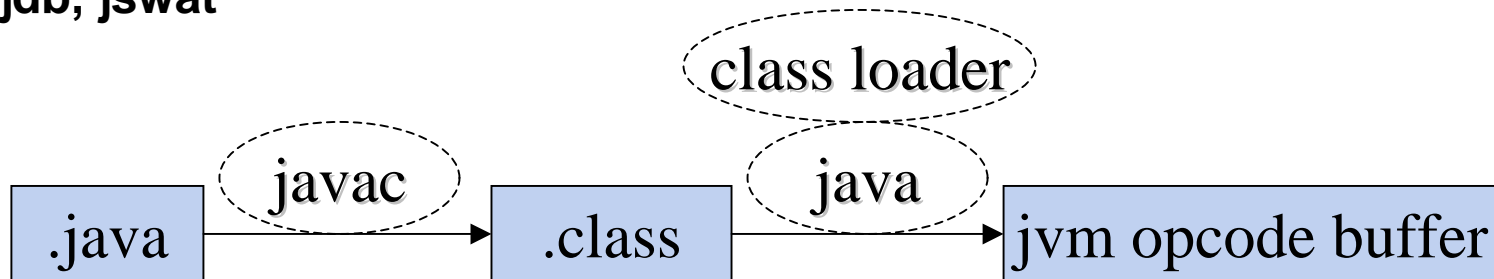
- ◎ Editor
- ◎ IDE: NetBeans, Eclipse, JBuilder ...

◎ Compling

- ◎ JDK: Sun JDK, IBM JDK
 - `javac -g *.java`
 - `-classpath -sourcepath -d`
 - `-encoding -source -target -deprecation`
- ◎ Jive, Gcj

◎ Running/Debug

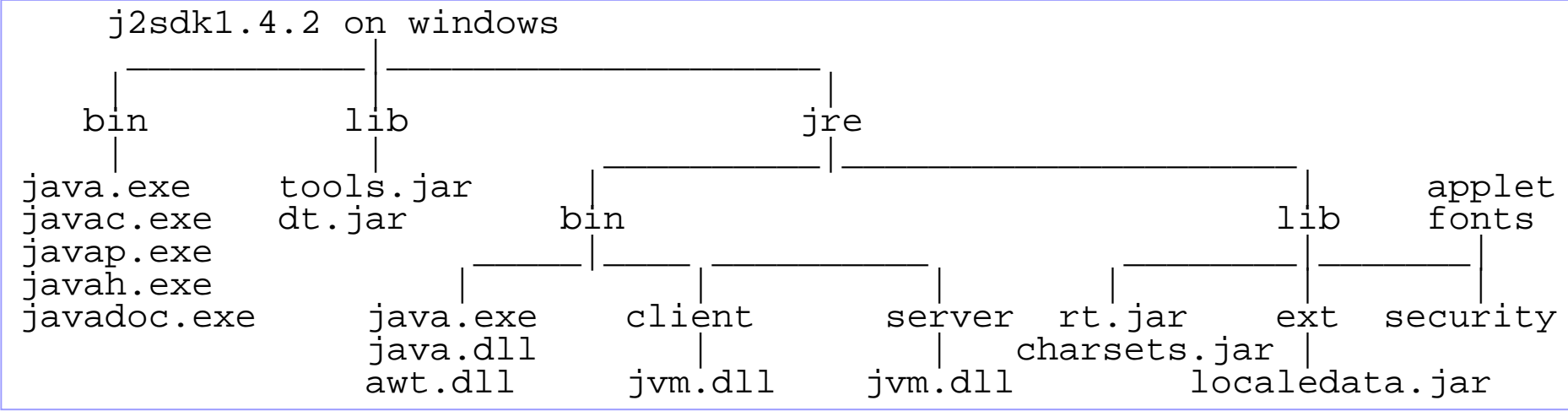
- ◎ JRE, JVM
 - `java ClassName`
- ◎ jdb, jswat



J2SE SDK Files

© jdk file overview

java
javac
javap
javah
javadoc



JRE_HOME

- ⊙ **JRE_HOME is the directory where jre installed.**
 - ⊙ i.e. c:\j2sdk141\jre
- ⊙ **jvm will guess the JRE_HOME as**
 - ⊙ parent of path of java.exe
 - ⊙ “Software\\JavaSoft\\Java Runtime Environment” registry
 - \\CurrentVersion i.e. 1.4
 - \\version\\JavaHome
 - **this step is windows specific feature*
- ⊙ **jvm use this path to locate system classes**



JAVA_HOME

- ⦿ **JAVA_HOME** is the directory where jdk installed.
- ⦿ **Many 3rd-part application need this path to locate tools**
 - ⦿ ant
 - ⦿ tomcat
- ⦿ **set as environment variable**



JVM on Windows

⊙ Where is the file?

- ⊙ ~\jre\bin\java.exe: Console application
- ⊙ ~\jre\bin\javaw.exe: Win32 application
- ⊙ ~\jre\bin\java.dll: real vm
 - ~\include\jni.h, JNI_CreateJavaVM()

⊙ Where is the class?

- ⊙ java.exe Hello → wants .\Hello.class
 - “.” is used to search running class (Hello) only
- ⊙ bootstrap classes:
 - ~\jre\lib\rt.jar
 - ~\jre\lib*.jar
- ⊙ extension classes: ~\jre\lib\ext*.jar
- ⊙ user classes
 - CLASSPATH environment variable
 - java.exe -bootclasspath -extdirs -classpath

⊙ java -verbose:class



⊙ JAR is java archive file

⊙ jdk\bin\jar.exe

- jar cf my.jar *.class

- jar xf my.jar

⊙ may use zip format compress

⊙ jvm understand .jar file

- java -classpath ../util.jar -jar my.jar

⊙ jvm understand non-compressed .zip file

⊙ manifest

⊙ META-INF/MANIFEST.MF

- *syntax is "Name: Value", section separate by empty line (CR)*

- Manifest-Version : 1.0

- Create-By: author

- Main-Class: qualified-ClassName

- Class-Path:

- Name:path/to/Class.class

⊙ META-INF/x.SF



Practice

- ⦿ download and install jdk (1.4 up)
- ⦿ what is your jre home and java home?
- ⦿ code, compile, run the hello world

Hello.java

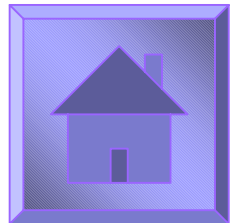
```
javac Hello.java  
java Hello
```

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



Module Q&A

1. sun jdk: <http://java.sun.com/j2se/>
2. ibm jdk: <http://www.ibm.com/java/jdk/>
<http://www-106.ibm.com/developerworks/java/jdk/>
3. jswat: <http://jswat.sourceforge.net/>
4. netbeans: <http://www.netbeans.org/>
5. eclipse: <http://www.eclipse.org/>
6. jbuilder: <http://www.borland.com/jbuilder/>



Language Syntax

- ⊙ **class**
- ⊙ **object**
- ⊙ **interface**
- ⊙ **package**
- ⊙ **comments**
- ⊙ **data type**
- ⊙ **declaration**
- ⊙ **expression & flow control**
- ⊙ **Exception**

James Gosling • Bill Joy • Guy Steele • Gilad Bracha

The Java™ Language Specification Second Edition

The Java™ Series



from the Source™



Ken Arnold • James Gosling • David Holmes

The Java™ Programming Language Third Edition

The Java™ Series



from the Source™



Joshua Bloch

Effective Java™ Programming Language Guide

Foreword by Guy Steele

The Java™ Series



from the Source™



Class - Basic

```
public class MyClass extends Object
                    implements IntfA, IntfB
{
    static {} //static initializer, run once on class load
    private int id = 0; //member variable

    MyClass() {
        super(); //call constructor of super class
    } //constructor, run very instance

    public int getId() { return id; } //member function
    synchronize public void setId(int id)
        throws Exception
    { this.id = id; }
}
```

JAVA is full OO language



Class – Member Access

⊙ Visibility Modifiers

⊙ **public > protected > *package > private**

Accessible to:	public	protected	package	private
Same class	yes	yes	yes	yes
Class in same package	yes	yes	yes	no
Subclass in different package	yes	yes	no	no
Non-subclass, different package	yes	no	no	no



Class - Advanced

- ⊙ *abstract* **Class**
- ⊙ *final* **Class**
- ⊙ **embed class**
 - ⊙ **Inner Class**
 - ⊙ **Locale**
 - ⊙ **Anonymous Class**
 - *new* AnonCls (...) { ... }
- ⊙ **member variable modifier**
 - ⊙ *static, final, transient, volatile*
- ⊙ **member function modifier**
 - ⊙ *static, final, abstract, native, synchronized*



Object

⊙ Declare

- ⊙ `ClassName objName;`

⊙ Create

- ⊙ `new ClassName(...)`

⊙ Destroy → Garbage Collection

- ⊙ no destructor!
- ⊙ `finalize()`
- ⊙ `System.gc()`

⊙ Memory Leek Issue

- ⊙ `System.out.println("now is "+(new Date())) : safe`
- ⊙ `vector.add(new Date()) : carefully`



Interface

```
public interface MyInterface extends IntfA, IntfB
{
    public final static MAX_ID = 100;
    public int id; //member variable

    public int getId(); //member function
    public abstract void setId(int id)
        throws Exception;
}
```

likes class with constrains:

1. public member only
2. no implementation of functions



Package – Directory Mapping

- © `package asb.tutorial.java;`
- © a implementation of name space

Hello.java

```
package asb.tutorial.java;  
public class Hello { ... }
```

```
java Hello
```

```
.\asb\tutorial\java\Hello.class
```

```
for each path in ($CLASSPATH)  
  $path\asb\tutorial\java\Hello.class
```



```
javac .\Hello.java
```

```
.\Hello.class
```

pacakge a\u00f8;
path → .\a@f8



Package – Class Visit

◎ Same package

- class `asb.tutorial.java.Hello` and `asb.tutorial.java.World`
- `Hello() { this.world = new World(); }`

◎ Other package

◎ qualified name

- `java.util.Vector v = new java.util.Vector();`

◎ import java.util.Vector;

- `Vector v = new Vector();`
- `v.add(new java.util.Date());`

◎ import java.util.*;

- `Vector v = new Vector();`
- `v.add(new Date());`
- `//you may not direct use classes under java.util.regex yet!`



Comments

- ◎ `/* ...lines... */`
- ◎ `//` comment to end of this line
- ◎ `/** ... */`
 - ◎ java doc comments
 - ◎ `javadoc.exe -d ../docs Hello.java`
 - ◎ <http://java.sun.com/j2se/1.4.2/docs/api/>
 - ◎ special tag in java doc comments
 - `@author author-name`
 - `@exception name description`
 - `@param name description, @return description`
 - `@see classname, @see classname#method-name`
 - `@version text`
 - `@deprecated`



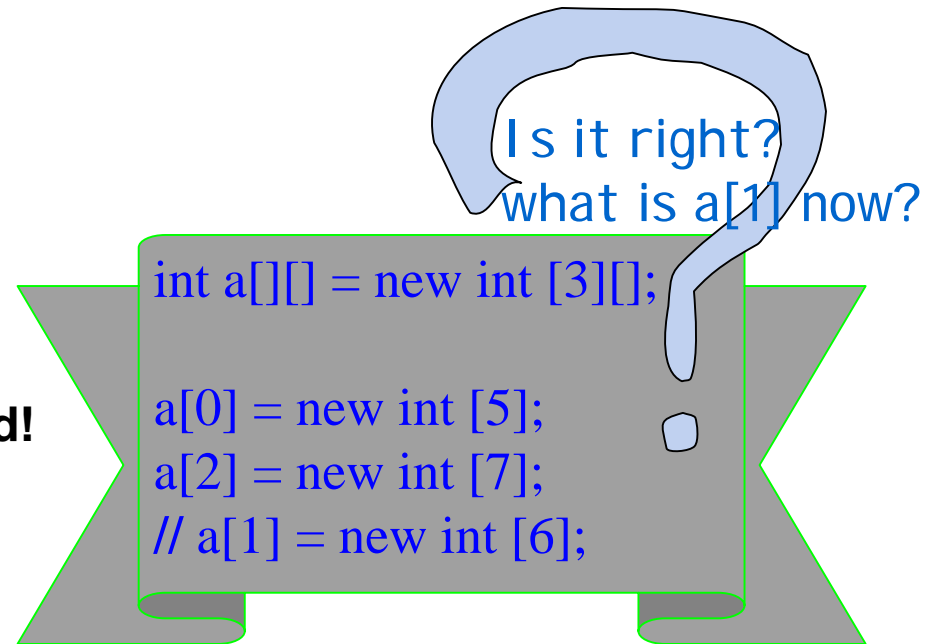
Data Type – Types

⊙ Primary Type

- ⊙ byte, short, int, long, char (16-bits unicode)
- ⊙ float, double
 - IEEE 754
 - POSITIVE_INFINITY, NEGATIVE_INFINITY, NaN
- ⊙ operator
 - same as c/c++
 - cannot override

⊙ Reference Type

- ⊙ class, interface
- ⊙ array
 - Type[] array → = new Type[n];
 - int array[] = {1, 2, 3, 4};
- ⊙ **java.lang.* : automatically imported!**
 - ⊙ Byte, Short, Integer, Long
 - ⊙ Char, String
 - ⊙ Float, Double



Data Type - Assignment

◎ Assignment Compatibility Between Integer Types

To/From	byte	char	short	int	long
byte	Assignable	Cast needed	Cast needed	Cast needed	Cast needed
char	Cast needed	Assignable	Cast needed	Cast needed	Cast needed
short	Assignable	Cast needed	Assignable	Cast needed	Cast needed
int	Assignable	Assignable	Assignable	Assignable	Cast needed
long	Assignable	Assignable	Assignable	Assignable	Assignable

◎ Class Type Assignment

- ◎ `java.lang.Class`, `java.lang.Object`
- ◎ `instanceof` operator
- ◎ `class.isInstanceOf(object)`
- ◎ `class.isAssignableFrom(class)`



Data Type – Useful Types

- ⊙ **java.lang.String, java.lang.StringBuffer**
 - ⊙ **stringA + stringB**
 - ⊙ **stringA + objectC → stringA + objectC.toString()**
 - ⊙ **stringA + intD → stringA + (new Integer(intD)).toString()**
 - ⊙ **new String(str.getBytes(“8859_1”), “gb2312”);**
- ⊙ **java.util.Date, java.util.Calendar**
- ⊙ **java.util.Vector, java.util.HashMap**

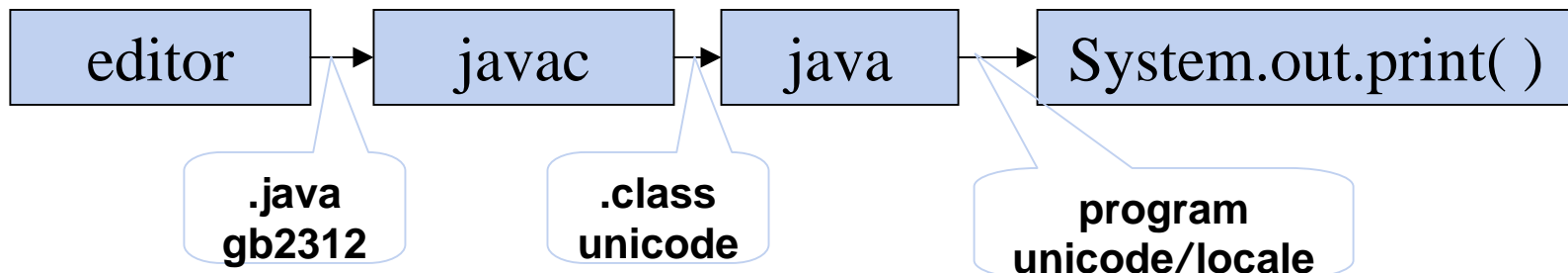


Declaration

◎ Name

- ◎ unicode name is acceptable
- ◎ \u4f60\u597d == 你好

```
public class Hello {  
    public static void 显示(String 参数)  
    {  
        System.out.println(参数);  
    }  
    public static void main(String[] args)  
    {  
        显示("你好，世界！");  
    }  
}
```



Expression & Flow Control

⊙ likes c/c++

- ⊙ if-then-else, switch-case-default
- ⊙ while, do-while, for
- ⊙ label :, break [label], continue [label]
- ⊙ return
- ⊙ throw new Exception("msg")

⊙ but not same

- ⊙ no goto
- ⊙ synchronized (expr) { ... }



Exception

- ◎ **java.lang.Throwable**
 - ◎ **java.lang.Exception**
 - ◎ **java.lang.Error**

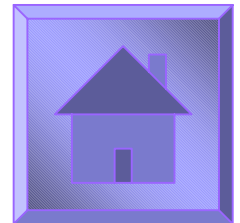
```
public class Hello
{
    static class BadWordE extends Exception
    {
        BadWordE(String word){super(word);}
    }
    public static void say(String words) throws BadWordE, Exception
    {
        if(words == null) throw new Exception("say what?!");
        if(words.indexOf("shit")>=0) throw new BadWordE("shit");
        System.out.println(words);
    }
    public static void main(String[] args){
        try                { say(args[0]);                }
        catch(BadWordE me){ System.out.println(me.getMessage()); }
        catch(Exception e){ e.printStackTrace();          }
        finally            { System.out.println("main() over."); }
    }
}
```



Bill Joy



Jams Gosling



Common Task

- ◎ Std In/Out
- ◎ String Manuplate
- ◎ Date and Time
- ◎ File I/O
- ◎ GUI
- ◎ Applet
- ◎ Network Communication



Std In/Out

⊙ class java.lang.System

⊙ public static InputStream in:

- std in
- int in.available()
- int in.read(byte[] buffer) /

⊙ public static PrintStream out:

- std out
- out.print(String msg)
- out.println()
- out.println(String msg)

```
import java.io.*;
public class JStdio{
    public static void main(String[] args)
    {
        byte[] name = new byte[8];
        System.out.print("please enter your name(<=8byte:");
        try{
            System.in.read(name);
            System.out.println();
            System.out.println("welcome, "+(new String(name)));
        }catch(IOException e){
            e.printStackTrace();
        }
    }
}
```

```
C:\>java JStdio
please enter your
name(<=8byte):a123456789

welcome, a1234567
```



String Manuplate

◎ java.lang.String

◎ Create

- String name = "abc";
- new String(byte[]) //take bytes as default charset encoded
- new String(byte[], String charsetName)

◎ Match

- name == "abc" ? → **No!**
- boolean equals(String)
- startsWith/endsWith/indexOf/lastIndexOf(str)
- matches(regexpr)

◎ Info

- length()
- getBytes(String charset)

◎ Change

- trim/replace/split/substring/concat

◎ java.lang.StringBuffer

- ◎ "a" + 4 + "c" → new StringBuffer().append("a").append(4).append("c").toString()
- ◎ insert(offset, param)
- ◎ append(parm) //param may be String, primary types or Object, char[]
- ◎ replace(start, end, str)



Date and Time

⊙ java.util.Date

- ⊙ before/after/equals

⊙ java.util.Calendar

- ⊙ singleton: getInstance()
- ⊙ get/set/add/roll/()
- ⊙ getMaximum/Minimum

⊙ java.sql. TimeStamp

⊙ java.text.DateFormat

- ⊙ format()
- ⊙ parse()

⊙ java.text.SimpleDateFormat

- ⊙ new SimpleDateFormat("y")
- ⊙ applyPattern()
- ⊙ format() / parse()

```
import java.util.*; import java.text.*;
public class Doy{
    public static void main(String[] args) {
        Calendar now = Calendar.getInstance();
        SimpleDateFormat df = new SimpleDateFormat();
        try{
            df.applyPattern("yyyy-MM-dd");
            now.setTime( df.parse(args[0]) );
            System.out.print(" Day (" +df.format(now.getTime())+" ) is ");
            System.out.print(now.get(Calendar.WEEK_OF_YEAR));
            System.out.print(" week of this year.");
        }catch(Exception e){
            e.printStackTrace();
        }
    }
}
```

```
C:\>java Doy 2004-7-31
```

```
Day (2004-07-31) is 31 week of this year.
```



⊙ java.io.File

- ⊙ `getPath() / isAbsolute() / getCanonicalPath`
- ⊙ `list() / listFiles()`
- ⊙ `isDirectory() / isFile() / exists() / length()`
- ⊙ `createNewFile() / createTempFile()`
- ⊙ `delete() / mkdir() / mkdirs() / renameTo()`

⊙ java.io.RandomAccessFile

⊙ basic stream

- ⊙ `BufferedReader → Reader → InputStream`
- ⊙ `BufferedWriter → Writer → OutputStream`
- ⊙ `PrintWriter`



GUI

◎ awt : Abstract Window Toolkit

- ◎ java.awt.Frame

- ◎ java.awt.Event

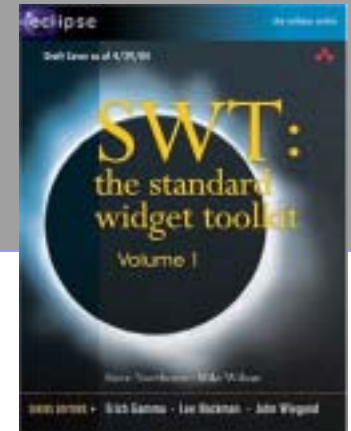
 - Listener, Adaptor

◎ swing

◎ swt



```
import java.awt.*; import java.awt.event.*;
public class MyAwt{
    public static void main(String[] args) {
        Frame win = new Frame("hello, awt gui!");
        win.setSize(200, 60);
        win.addWindowListener(
            new WindowAdapter(){
                public void windowClosing(WindowEvent e)
                { System.exit(0); }
            });
        win.show();
    }
}
```



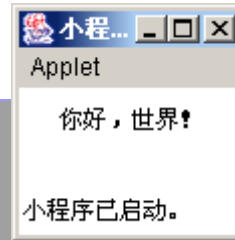
Applet

⊙ GUI embedded in Browser

- ⊙ IE need install java plugin to run applet written in swing
- ⊙ Open java console for debug

⊙ appletviewer.exe htmlfile

```
import java.awt.*; import java.applet.*;
public class MyApplet extends Applet{
    private String words = "hello!";    private int x=10, y=10;
    public void init() {
        try{
            words = getParameter("sayWhat");
            x = Integer.parseInt( getParameter("x") );
            y = Integer.parseInt( getParameter("y") );
        } catch( Exception e ) { e.printStackTrace(); }
    }
    public void paint(Graphics g){
        g.drawString(words, x, y);
    }
}
```



```
<html><body>
<applet code="MyApplet.class"
        width=100 height=50 >
    <param name="sayWhat"
            value="你好，世界！">
    <param name="x" value="20">
    <param name="y" value="20">
</applet>
</body>
</html>
```



Network Communication

◎ java.net.URI

- ◎ URI = { URL / URN } , absolute or relative
 - RFC 2396: *Uniform Resource Identifiers (URI): Generic Syntax*
- ◎ [scheme:]scheme-specific-part[#fragment]
 - mailto:java-net@java.sun.com
- ◎ [scheme:]//[authority][path][?query][#fragment]
 - http://java.sun.com/j2se/1.3/docs/guide/collections/designfaq.html#28
 - authority: [user-info@]host[:port]
 - ftp://user:passwd@ftp.some.com/pub/java_intro.ppt
 - jar:http://www.foo.com/bar/baz.jar!/COM/foo/Quux.class



◎ java.net.URL: absolute

- dependent URLStreamHandler
- URLConnection.openConnection()
- InputStream.openStream()

◎ java.net.URLConnection

- ◎ setup parameters → connect() → remote obj available (header/content)
- ◎ setup parameters
 - setAllowUserInteraction / setDoInput / setDoOutput / setIfModifiedSince / setUseCaches
 - setRequestProperty / setDefaultAllowUserInteraction / setDefaultUseCaches
- ◎ access remote object
 - getContent / getHeaderField / getInputStream / getOutputStream
 - freq. used header fields: getContentEncoding / getContentLength / getContentType ...

◎ Socket : java.net.Socket / java.net.ServerSocket



Module Q&A

Java™ 2 Platform, Standard Edition v 1.4

