CS203 Programming with Data Structures

Input and Output

Chengyu Sun
California State University, Los Angeles

Overview

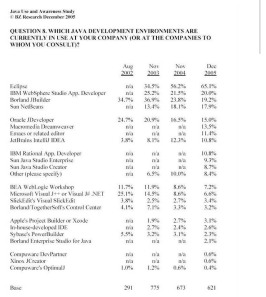
- Java IDEs
- Package
- Input and output
- Exception handling

Java IDEs

- JBuilder
  - Commercial with a scaled-down free version (`JBuilder Foundation`)
- Eclipse
  - http://www.eclipse.org
  - High quality and free
  - Many tools and plug-ins have to be installed separately
- Netbeans
  - http://www.netbeans.org
  - All-in-one package
  - Slow on older computers

IDE Usage Statistics

- Survey by *BZ Research* in December 2005
  - Eclipse: 65.1%
  - JBuilder: 19.2%
  - Netbeans: 17.9%



Java Use and Awareness Study
© BZ Research December 2005

QUESTION 8. WHICH JAVA DEVELOPMENT ENVIRONMENTS ARE CURRENTLY IN USE AT YOUR COMPANY (OR AT THE COMPANIES TO WHOM YOU CONSULT)?

	Aug 2002	Nov 2003	Nov 2004	Dec 2005
Eclipse	n/a	34.5%	56.2%	65.1%
IBM WebSphere Studio App. Developer	n/a	25.2%	21.5%	20.0%
Borland JBuilder	34.7%	30.9%	23.8%	19.2%
Sun NetBeans	n/a	13.4%	18.1%	17.9%
Oracle JDeveloper	24.7%	20.9%	16.5%	15.0%
Macromedia Dreamweaver	n/a	n/a	n/a	13.5%
Emacs or related editor	n/a	n/a	n/a	11.4%
JetBrains IntelliJ IDEA	3.0%	8.1%	12.3%	10.8%
IBM Rational App. Developer	n/a	n/a	n/a	10.8%
Sun Java Studio Enterprise	n/a	n/a	n/a	9.3%
Sun Java Studio Creator	n/a	n/a	n/a	8.7%
Other (please specify)	n/a	6.3%	10.0%	8.4%
BEA WebLogic Workshop	11.7%	11.9%	8.6%	7.2%
Microsoft's Visual Studio or Visual J# .NET	25.1%	14.9%	8.6%	6.4%
MetaMatrix Visual MetaEdit	3.4%	5.5%	3.7%	3.4%
Borland TogetherSoft's Control Center	4.1%	7.1%	3.3%	3.2%
Apple's Project Builder or Xcode	n/a	1.9%	2.7%	3.1%
In-house-developed IDE	n/a	2.7%	2.4%	2.6%
Sybase's PowerBuilder	5.9%	3.2%	3.1%	2.3%
Borland Enterprise Studio for Java	n/a	n/a	n/a	2.1%
Compuware DevPartner	n/a	n/a	n/a	0.8%
Xinox JCreator	n/a	n/a	n/a	0.6%
Compuware's OptimalJ	1.0%	1.2%	0.6%	0.4%
Base	291	775	673	621

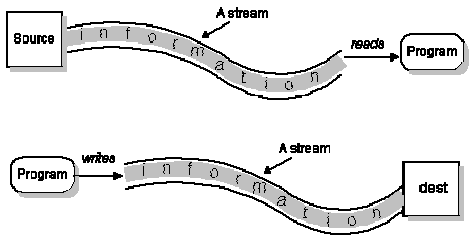
Package

- Why do we need it?
- What does it do?
- Package and directory

Input and Output

- Console
  - Scanner
  - System.out
- GUI
- File
  - Reader and Writer
  - InputStream and OutputStream
  - Scanner
- Command line parameters

Understand `java.io` Package

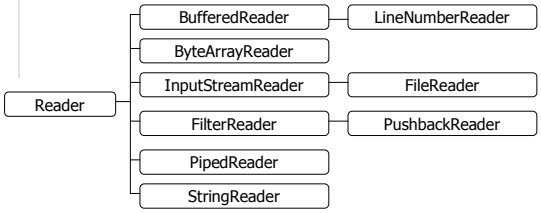


http://java.sun.com/j2se/1.5.0/docs/api/java/io/package-summary.html

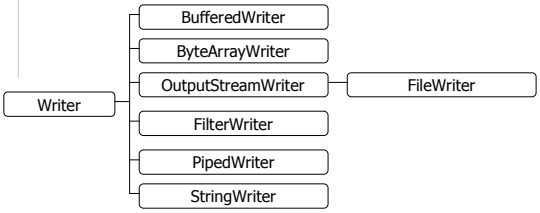
Stream Types

- Character streams
  - Textual information
  - Handled by *Reader* and *Writer* classes
- Byte streams
  - Binary information
  - Handled by *InputStream* and *OutputStream* classes

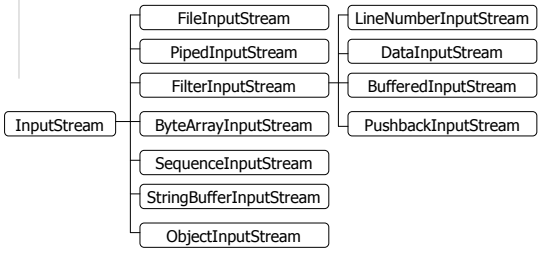
Reader Classes



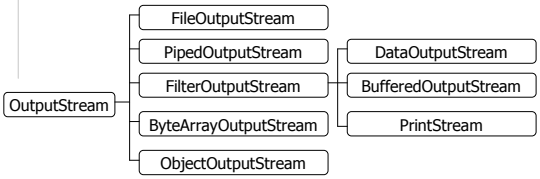
Writer Classes



InputStream Classes



OutputStream Classes



Basic Streams by Source/Destination

- Files
  - FileReader/Writer/InputStream/OutputStream
- Threads
  - PipedReader/Writer/InputStream/OutputStream
- Memory
  - ByteArrayReader/Writer/InputStream/OutputStream
  - StringReader/Writer
  - StringBufferInputStream
- General
  - InputStream, InputStreamReader
  - OutputStream, OutputStreamWriter

Basic Stream Operations

- Basic streams only recognize *bytes* or *characters*
- Operations
  - Read/write a single byte or character
  - Read/write an array of bytes or characters
- Inconvenient

Wrapper Streams by Function

- Data conversion
  - DataInputStream/OutputStream
- Printing
  - PrintStream
- Buffering
  - BufferedReader/Writer/InputStream/OutputStream
- Object serialization
  - ObjectInputStream/OutputStream
- Others

Important Wrapper Streams and Operations

- DataInputStream and DataOutputStream
  - Read and write primitive types
  - readInt(), readDouble(), …
  - writeInt(int i), writeDoube(double d), …
- BufferedReader
  - readLine()
- BufferedWriter
  - write(String s)

"Wrapping" Examples

```
// buffered text file read/write
BufferedReader br = new BufferedReader( new FileReader("file") );
BufferedWriter bw = new BufferedWriter( new FileWriter("file") );

// un-buffered binary file read/write
DataInputStream di = new DataInputStream( new FileInputStream("file") );
DataOutputStream do = new DataOutputStream( new FileOutputStream("file") );

// buffered binary file read/write
DataInputStream di2 = new DataInputStream( new BufferedInputStream(
                                    new FileInputStream() ) );
DataOutputStream do2 = new DataOutputStream( new BufferedOutputStream(
                                    new FileOutputStream() ) );
```

How to Choose from Stream Classes

- Step 1: Is the data in **binary** form or **textual** form?
  - Binary: Input/OutputStream
  - Textual: Reader/Writer
- Step 2: What's the data **source** or data **destination**?
  - Files, threads, memory, general
- Step 3: How to **process** the data?
  - Primitive data types, buffering, …

File Input Example

- Read from a file in the following format, and sum up all numbers

```
31    20    30
22    33    –1    43
23    33    44    45
79    1
-10
```

Paths

- Windows
  - Absolute path
    - c:\path\to\file
  - Relative path
    - path\to\file
- Unix
  - Absolute path
    - /path/to/file
  - Relative path
    - path/to/file
- File separators – "**/**", "****", **File.separator**

Exception Handling

- `throw` and `throws`
- `try`, `catch`, and `finally`
- `Exception` class
  - `getMessage()`
  - `printStackTrace()`

`Scanner` Class

- A Java 1.5 feature
- In `java.util` package

```
Scanner s = new Scanner(System.in);
String param = s.next();
int value = s.nextInt();
s.close();

int sum = 0;
Scanner s2 = new Scanner( new File("test") );
while( s2.hasNext() ) sum += s2.nextInt();
s2.close()
```

`File` Class

- Not directly related to I/O
- Check file status
  - Is a file or a directory
  - Exist, readable, writable
  - Name, path, parent
  - Length
- Perform common file/directory operations
  - Get parent, list children
  - Delete, rename, mkdir

Binary File vs. Text File

- If we can save data in either binary or text form, which one do we choose?
  - File size
  - Convenience
  - Speed
- Either way, always use buffering!

Random Access File

- The problem with the *stream* model
- Advantages of `RandomAccessFile`
  - Deal with both binary and text files
  - Provide both read and write methods
  - `seek(long pos)`
- … but we don't use it often. Why?