Input / Output (IO)

© 1992-2007 Pearson Education, Inc. All rights reserved.

Last time

- error handling and exceptions
- `try`, `catch`, `finally`, `throws`, and `throw` keywords
- Java exception hierarchy
- stack trace

© 1992-2007 Pearson Education, Inc. All rights reserved.

Objectives

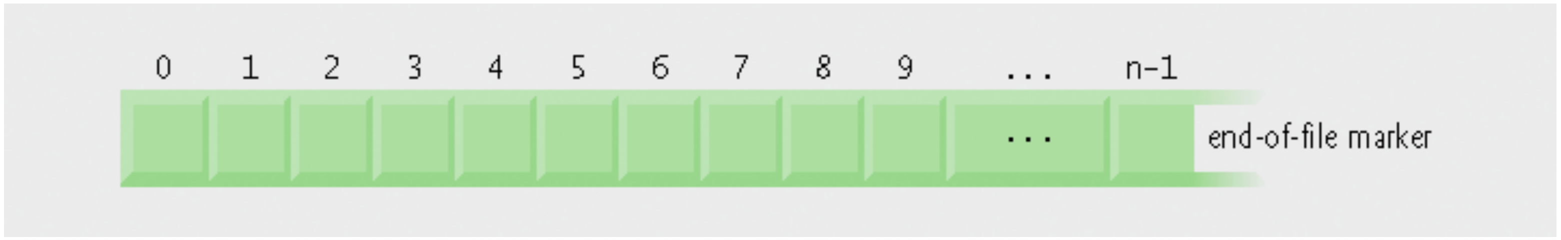
- data streams
- files and URLs
- various I/O examples
- example GUI programs

© 1992-2007 Pearson Education, Inc. All rights reserved.

Data Streams

- Java views data as a sequential stream of bytes
- For example, we open:
  - file stream
  - audio stream
  - network stream, . . .
- Operating system provides mechanism to determine end of file (*e.g.*, end-of-file marker)

© 1992-2007 Pearson Education, Inc. All rights reserved.



Java’s view of a file of *n* bytes.

© 1992-2007 Pearson Education, Inc. All rights reserved.

Standard streams

- `System.in` – standard input stream object
- `System.out` – standard output stream object
- `System.err` – standard error stream object

© 1992-2007 Pearson Education, Inc. All rights reserved.

Stream types

- Byte-based streams – stores data in binary format
  - Binary files – created from byte-based streams, read by a program that converts data to human-readable format

- Character-based streams – stores data as a sequence of characters
  - Text files – created from character-based streams, read by text editors

© 1992-2007 Pearson Education, Inc. All rights reserved.

Byte-Based Input and Output

- `InputStream` / `OutputStream` **abstract classes**

- `BufferedInputStream` / `BufferedOutputStream` **classes**
- `FileInputStream` / `FileOutputStream` **classes**
- `PrintStream` **output class**
- `PipedInputStream` / `PipedOutputStream` **classes**
- `FilterInputStream` / `FilterOutputStream` **classes**
- `SequenceInputStream` **class**
- `DataInput` / `DataOutput` **interfaces**

© 1992-2007 Pearson Education, Inc. All rights reserved.

Character-Based Input and Output

- `Reader` / `Writer` **abstract classes**

- `BufferedReader` / `BufferedWriter` **classes**
- `FileReader` / `FileWriter` **classes**
- `PrintWriter` **output class**
- `CharArrayReader` / `CharArrayWriter` **classes**
- `PipedReader` / `PipedWriter` **classes**
- `StringReader` / `StringWriter` **classes**

© 1992-2007 Pearson Education, Inc. All rights reserved.

Class `java.io.File`

- Class `File` useful for retrieving information about files and directories from disk
- Objects of class `File` do not open files or provide any file-processing capabilities

© 1992-2007 Pearson Education, Inc. All rights reserved.

Method	Description
`boolean canRead()`	Returns `true` if a file is readable by the current application; `false` otherwise.
`boolean canWrite()`	Returns `true` if a file is writable by the current application; `false` otherwise.
`boolean exists()`	Returns `true` if the name specified as the argument to the `File` constructor is a file or directory in the specified path; `false` otherwise.
`boolean isFile()`	Returns `true` if the name specified as the argument to the `File` constructor is a file; `false` otherwise.
`boolean isDirectory()`	Returns `true` if the name specified as the argument to the `File` constructor is a directory; `false` otherwise.
`boolean isAbsolute()`	Returns `true` if the arguments specified to the `File` constructor indicate an absolute path to a file or directory; `false` otherwise.

File methods.
(Part 1 of 2)

© 1992-2007 Pearson Education, Inc. All rights reserved.

Method	Description
`String getAbsolutePath()`	Returns a string with the absolute path of the file or directory.
`String getName()`	Returns a string with the name of the file or directory.
`String getPath()`	Returns a string with the path of the file or directory.
`String getParent()`	Returns a string with the parent directory of the file or directory (i.e., the directory in which the file or directory can be found).
`long length()`	Returns the length of the file, in bytes. If the `File` object represents a directory, `0` is returned.
`long lastModified()`	Returns a platform-dependent representation of the time at which the file or directory was last modified. The value returned is useful only for comparison with other values returned by this method.
`String[] list()`	Returns an array of strings representing the contents of a directory. Returns `null` if the `File` object does not represent a directory.

File methods.
(Part 2 of 2)

© 1992-2007 Pearson Education, Inc. All rights reserved.

Demo: Simple File I/O

- Write some text (strings) to a file
- Then read the text from that file

- Possible exceptions
  - `FileNotFoundException` – signals that an attempt to open the file denoted by a specified pathname has failed
  - `IOException` – signals that an I/O failure of some sort has occurred

© 1992-2007 Pearson Education, Inc. All rights reserved.

Demo: Line number / total lines

Open a text file and output it to the console such that each line is prefixed by its line number and the total number of lines in the file.

Can use the following I/O classes:

- `FileReader` – character reader from a file
- `BufferedReader` – buffered reader to obtain lines
- `System.out` – standard output to console

© 1992-2007 Pearson Education, Inc. All rights reserved.

Demo: Zip up the input stream

Store the standard input stream to a zipped file (gzip format), until we input Ctrl-D (EOF).

Can use the following I/O classes:

- `System.in` – **standard input stream**
- `InputStreamReader` – **go from reader**
- `BufferedReader` – **buffered reader will give lines**

- `FileOutputStream` – **byte output to file**
- `GZIPOutputStream` – **zip up the stream**
- `PrintStream` – **byte output line by line**

© 1992-2007 Pearson Education, Inc. All rights reserved.

Demo: Output URL to console

Open a URL input stream and output its contents to the console.

Can use the following I/O and network classes:

- `java.net.URL` **– Internet URL support**
- `InputStreamReader` **– bridge from bytes to chars**
- `BufferedReader` **– buffered reader to obtain lines**
- `System.out` **– standard output stream**

© 1992-2007 Pearson Education, Inc. All rights reserved.

Demo: Image I/O and JPanel

Open a JPEG image and display it on a panel.

Can use the following classes:

- `javax.imageio.ImageIO` **– image I/O support**
- `java.awt.image.BufferedImage` **– image buffer**
- `Graphics.drawImage` **– draw image on a panel**

© 1992-2007 Pearson Education, Inc. All rights reserved.