14. File Input and Output

12 Nov 2015

Objectives

- Using File object to represents file and directory.
- Include a JFileChooser object in your program to let the user specify a file.
- Write text data to a file using PrintWriter.
- Read a text file using Scanner.

File Objects

- In this section, we introduce Java standard classes for reading data from or writing data to a file.
- Before we can start reading data from file or list files in directory, we must first create a File object (in java.io package).
- File object can be representation of file and directory.

File Objects

- Using File object as a directory: By passing path of directory to File's constructor
  - isDirectory() method: Use to check is it a directory or not.

```
File dir1 = new File("/Users/tomm/Documents/data/");

if(dir1.isDirectory()){

    System.out.println("dir1 is directory.");
}
```

  - Output:

```
dir1 is directory.
```

File Objects

- Using File object as a file: By passing path and file's name to File's constructor
  - isFile() method: Use to check is it a file or not.

```
File file1 = new File("/Users/tomm/Documents/data/test.data");
if(file1.isFile()){
    System.out.println("file1 is file.");
}
```

  - Output:

```
file1 is file.
```

File Objects

- We can check if a File object is associated to an existing file or directory by using exists method
  - exists() method: Use to check is it exists or not.

```
File dir1 = new File("/Users/tomm/Documents/data/");

if(dir1.exists()){

    System.out.println("dir1 exists");
}
```

  - Output:

```
dir1 exists
```

File Objects

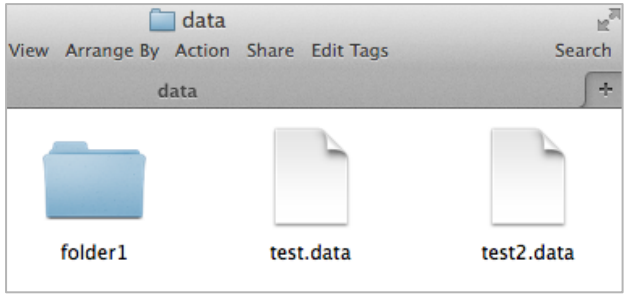
- If File object is a directory, we can list the contents of the directory by using list method
  - list() method: Use to get all contents' name.

```
File dir1 = new File("/Users/tomm/Documents/data/");
String[] names = dir1.list();

for(String name : names)
    System.out.println(name);
```

  - Output:

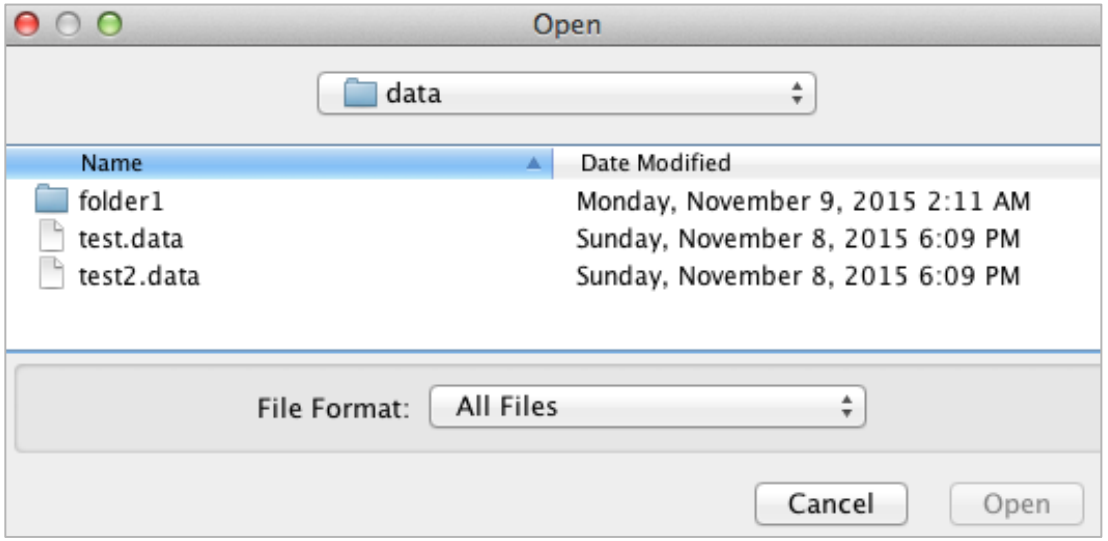
```
folder1
test.data
test2.data
```



JFileChooser Objects

- We can use a javax.swing.JFileChooser object to let the user select a file easily.

```
JFileChooser chooser =
        new JFileChooser("/Users/tomm/Documents/data/");
chooser.showOpenDialog(null);
```



JFileChooser Objects

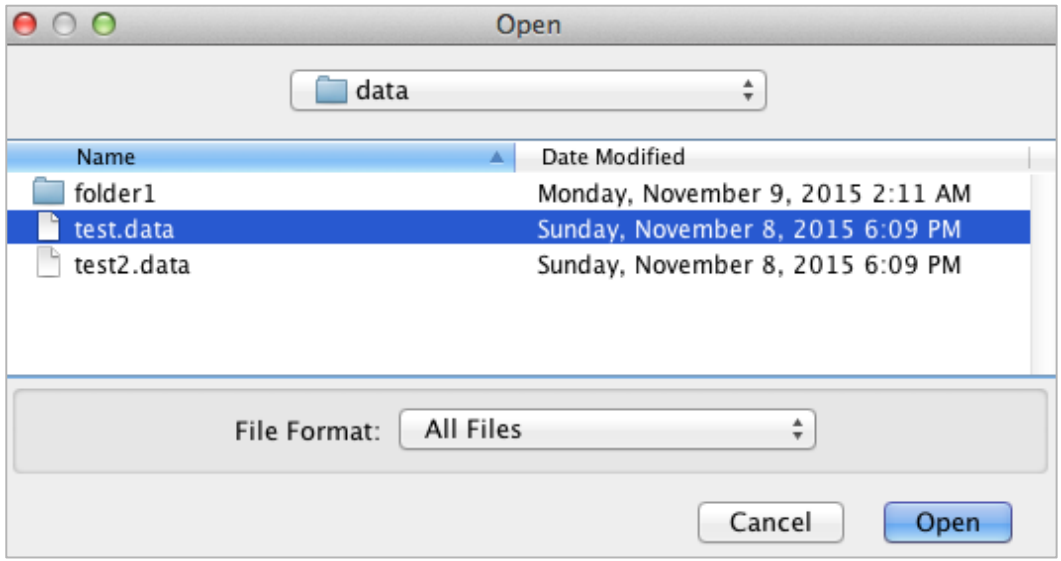
- To check whether the user has clicked on the Open or Cancel button, we test the return value from the showOpenDialog() method.

```
JFileChooser chooser =
        new JFileChooser("/Users/tomm/Documents/data/");

int status = chooser.showOpenDialog(null);
if (status == JFileChooser.APPROVE_OPTION) {
    System.out.println("Open is clicked");
}
else if (status == JFileChooser.CANCEL_OPTION){
    System.out.println("Cancel is clicked");
}
```

JFileChooser Objects

– Output:




```
Open is clicked
```

JFileChooser Objects

- Once we determine the Open button is clicked, we can retrieve the selected file using getSelectedFile() method.

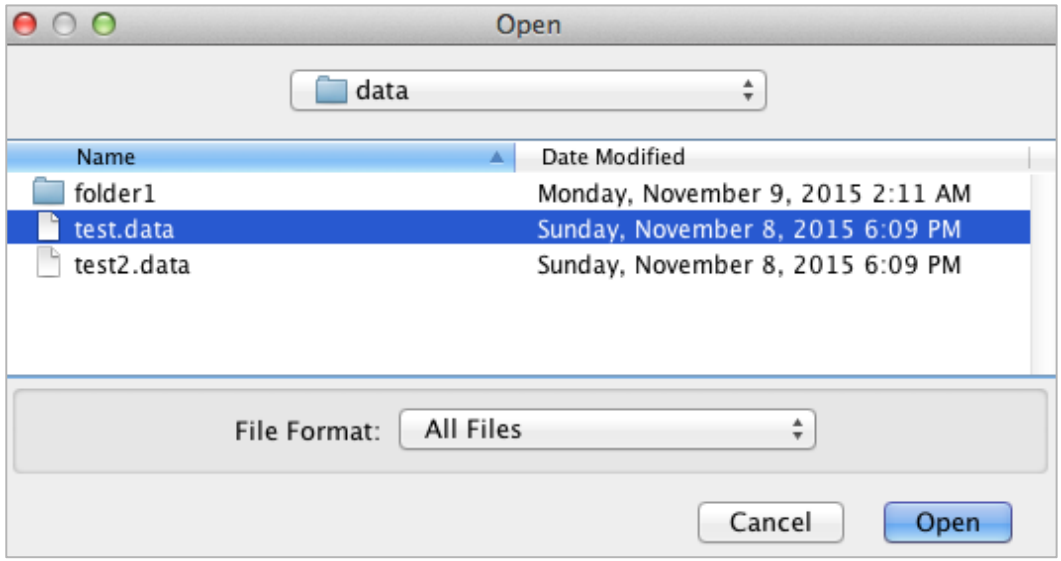
```
JFileChooser chooser =
        new JFileChooser("/Users/tomm/Documents/data/");

int status = chooser.showOpenDialog(null);
if (status == JFileChooser.APPROVE_OPTION) {
    File file = chooser.getSelectedFile();

    System.out.println("Selected File: \t" + file.getName());
    System.out.println("Full path: \t" + file.getAbsolutePath());
}
```

JFileChooser Objects

– Output:




```
Selected File:  test.data
Full path:      /Users/tomm/Documents/data/test.data
```

File Input and Output

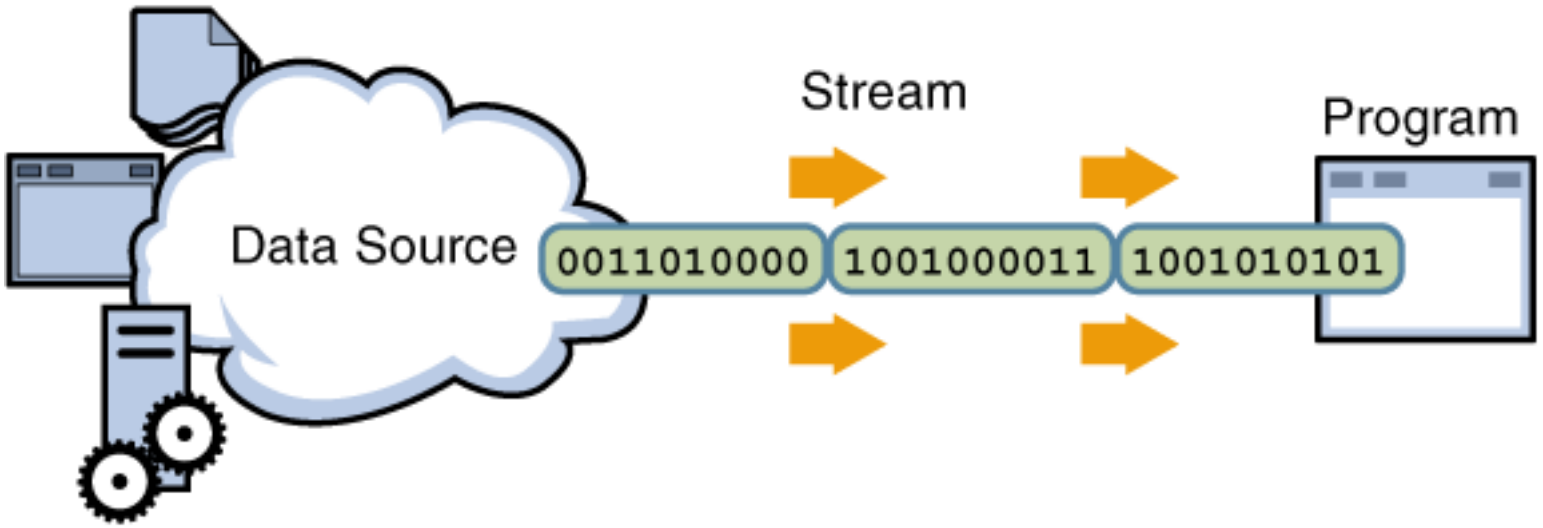
- In Java, we can read - write both binary and text file, in this section will focus on text file only.
- To read data from or write data to a file, we must create Java stream objects and attach it to the file.

File Input and Output

- A stream is simply a sequence of data items, usually 8 bits per item.
- Java has two types of streams:
  - Input stream
  - Output stream

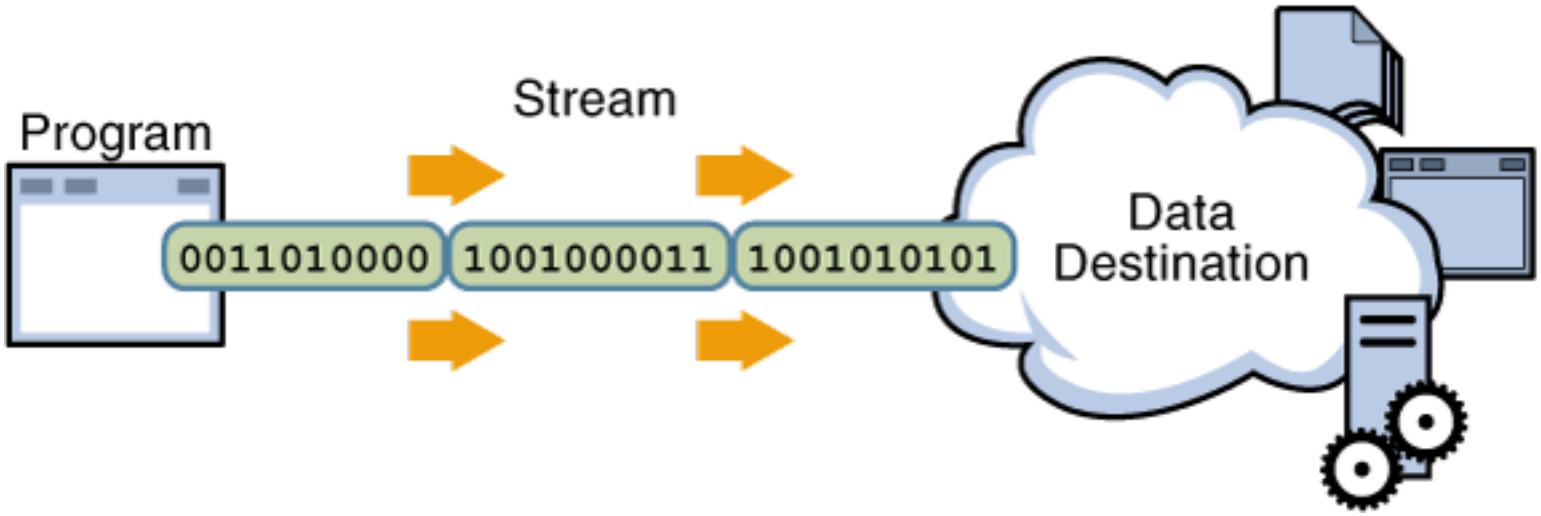
File Input and Output

- Input stream: A program uses an input stream to read data from a source, one item at a time.



File Input and Output

- Output stream: A program uses an output stream to write data to a destination, one item at time.



File Input and Output

- To write the text file, we use three objects: File, FileInputStream, and PrintWriter.

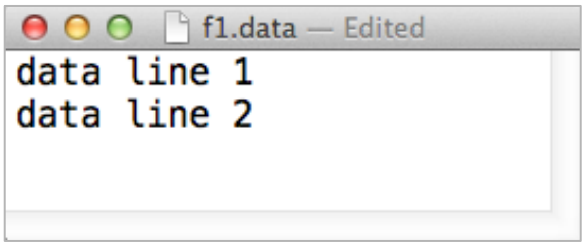
```
File outFile = new File("/Users/tomm/Documents/data/f1.data");

FileOutputStream outFileStream = new FileOutputStream(outFile);

PrintWriter outStream = new PrintWriter(outFileStream);

outStream.println("data line 1");
outStream.println("data line 2");
outStream.close();
```

  - Output:



File Input and Output

- To read the data from the file, since Java 5.0, we can use the Scanner object to read data from a text file easily. We use 2 objects: File and Scanner.

```
File inFile = new File("/Users/tomm/Documents/data/f1.data");

Scanner scanner = new Scanner(inFile);

while(scanner.hasNext()){
    System.out.println(scanner.nextLine());
}

scanner.close();
```

  - Output

```
data line 1
data line 2
```

Summary

- A File object represents a file or a directory.
- An instance of the JFileChooser class is a file dialog that lets the user select a file to read data from or save data to.
- Various input and output stream classes are defined in the java.io package.

Summary

- With text I/O, data are read and saved as strings.
- PrinterWriter class is used for create text file.
- The Scanner class can be used to input data from a text file.

Reference

- C. Thomas Wu, An Introduction to Object-Oriented Programming with Java, 5th Edition
  - Chapter 12: File Input and Output

Question?