



Java IO

Wang Yang
wyang@njnet.edu.cn



- Java I/O Introduction
- File and Directory
- Byte-stream and Character-stream
- Random Access File



Java I/O Introduction



Java I/O Introduction

- I/O Target
 - File
 - Console
 - Network Connection
- I/O Manner
 - Text-based (char) / Data-based(byte)
 - Sequential / Random Access



Java I/O Introduction

- java.io Package
 - general classes
 - filtered / buffered / piped streams
 - data streams
 - File
 - object serialization



File and Directory



File and Directory

- `java.io.File` - “A Path in a file system”
 - File
 - Directory
- File Construction

```
File file = new File("c:/Windows/explorer.exe");  
File file = new File("c:/Windows", "explorer.exe");  
File file = new File(".");
```



File and Directory

- File Method
 - `isFile()`, `isDirectory()`, `exists()`
 - `delete()`, `createNewFile()`, `mkdir()`
 - `list()`, `listFiles()`



File and Directory

```
private void testFile(){  
    File f = new File("/Users");  
    File[] filelist = f.listFiles();  
    for(File subf : filelist){  
        System.out.println(subf.getAbsolutePath());  
    }  
}
```

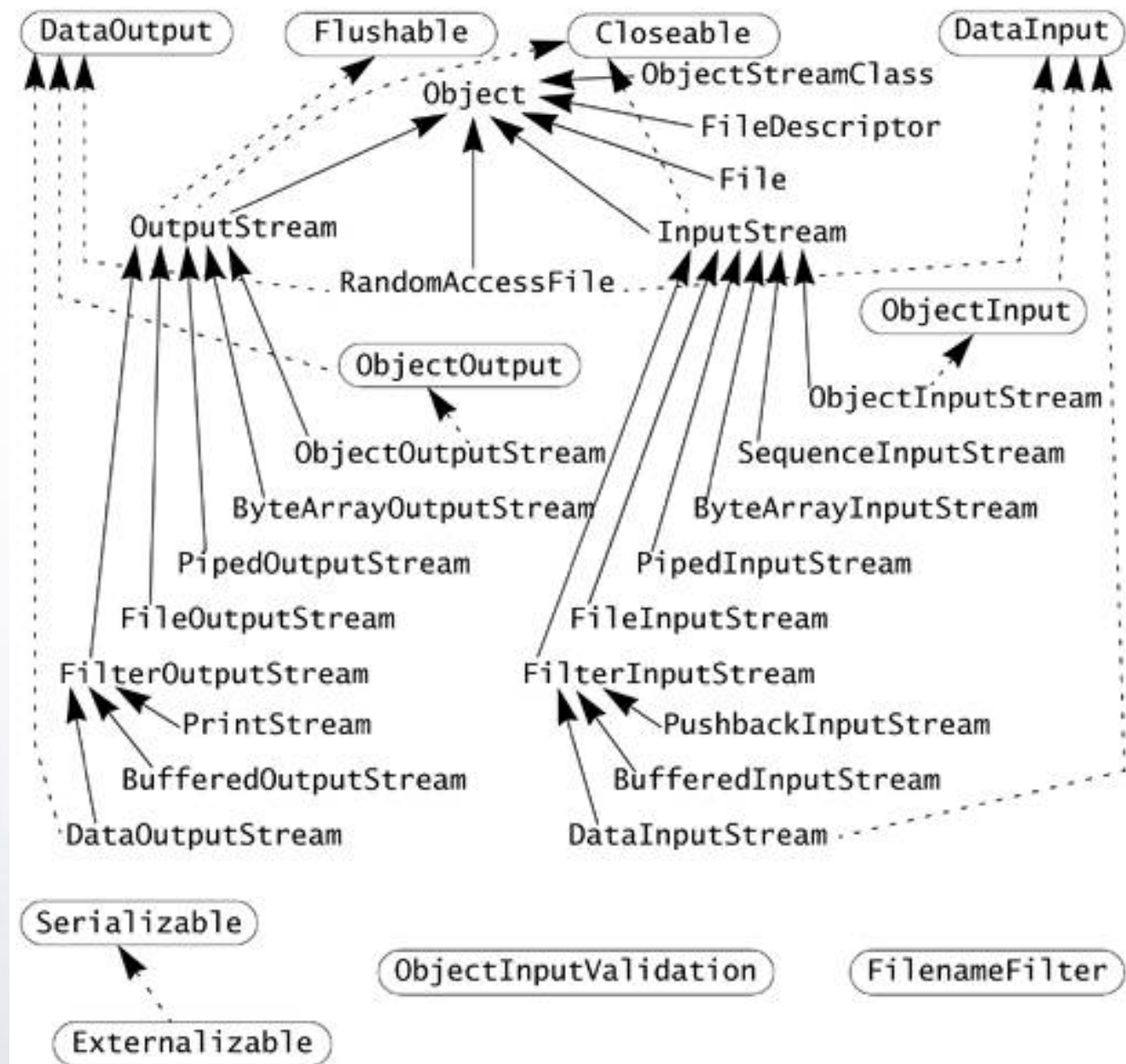


Byte-stream and Character-stream



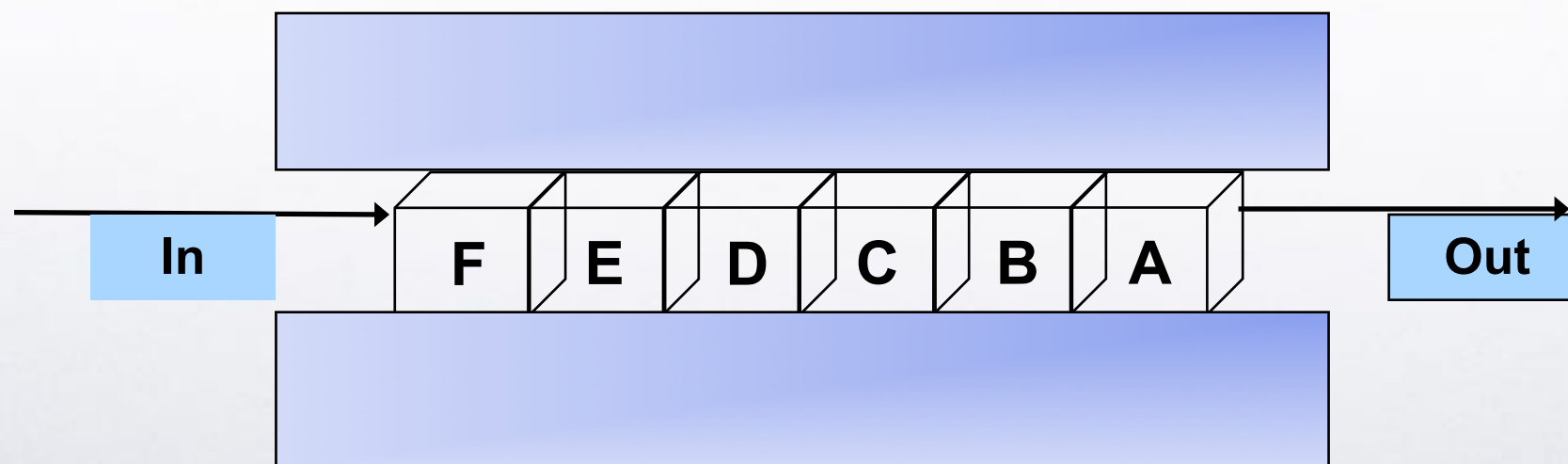
Byte Stream

- Complex Hierarchy of ByteStream





- The Notion of Stream
 - A **sequence** of flowing **byte / char**
 - A channel sending message in **FIFO**





Stream

- Classification of Stream

- Byte Stream

- **Byte** as the unit

10010011 01010010 10100101 01010100

- Used to read and write binary **data**

- Character Stream

- **Char** as the unit

Welcome to the CoSE!

- Used to read and write **text**



Stream

- **Abstract** Stream Class in java.io
 - Byte stream
 - java.io.InputStream
 - int read() //read a byte
 - java.io.OutputStream
 - write(int b) //write an int ??
 - Character stream
 - java.io.Reader
 - int read() //read a char
 - java.io.Writer
 - write(int b) //write an int ??



Stream

- Implemented Classes in java.io
 - Byte stream
 - FileInputStream、FileOutputStream
 - DataInputStream、DataOutputStream
 - ByteArrayInputStream、ByteArrayOutputStream
 - BufferedInputStream、BufferedOutputStream
 - ObjectInputStream、ObjectOutputStream
 - Character stream
 - FileReader、FileWriter
 - PipedReader、PipedWriter
 - BufferedReader、BufferedWriter
 - Bridge
 - InputStreamReader、OutputStreamWriter



A new thing

- IOException

```
try{  
  
} catch (IOException e) {  
    System.out.println(e);  
}
```




ByteStream

- initialize stream from

- File Object

- FileName

- Other Stream

- Remember **close()**

- release the system resource

```
try {  
    File f = new File("TEST.txt");  
    InputStream is = new FileInputStream(f);  
    // do some stuff  
    is.close();  
  
    OutputStream os = new FileOutputStream("TEST.txt");  
    // do some stuff  
    os.close();  
} catch (IOException e) {  
    System.out.println(e);  
}
```



Byte Stream

- FileInputStream
 - Read **bytes** from file system
 - Used to read image or **data**
- FileOutputStream
 - Write **bytes** to file system
 - Used to write image or data



...10110100 10111001...



Byte Stream

- Example:
 - Write following data into “c:\test.dat”
 - Read them out
 - byte 97
 - char ‘b’
 - String “好”



Byte Stream

```
public static void main(String[] args) {  
    // TODO Auto-generated method stub  
    try{  
        File f = new File("TEST.txt");  
        if(!f.exists()){  
            f.createNewFile();  
        }  
        FileOutputStream fos = new FileOutputStream(f);  
        fos.write(97);  
        fos.write('c');  
        fos.write("你好".getBytes());  
        fos.close();  
  
        FileInputStream fis = new FileInputStream(f);  
        int b = fis.read();  
        while(b != -1){  
            System.out.println(b);  
            b = fis.read();  
        }  
        fis.close();  
    } catch (IOException e){  
        System.out.println(e);  
    }  
}
```




Byte Stream

- Think
 - What is the result of this program?
 - If the parameter of `read()` and `write()` out of range of `Byte`, what will happen?



Byte Stream

- Output

97	97
99	'c'
228	“你”
189	
160	
229	“好”
165	
189	



Byte Stream

- Input

```
FileOutputStream fos = new FileOutputStream(f);  
fos.write(260);  
fos.write('你');  
fos.write('好');  
fos.write("你好".getBytes());  
fos.close();
```

|



Byte Stream

- Output

4	260
96	、你、
125	、好、
228	“你”
189	
160	
229	“好”
165	
189	



Byte Stream

- Why different “你”
 - Encoding Charset
 - Internal : Unicode (UTF-16) 2byte(79,96)
 - Output : System Default
 - GB 18030 (GBK) 2byte(125,196)
 - UTF-8 (UTF8) 3byte(228,189,160)



Byte Stream

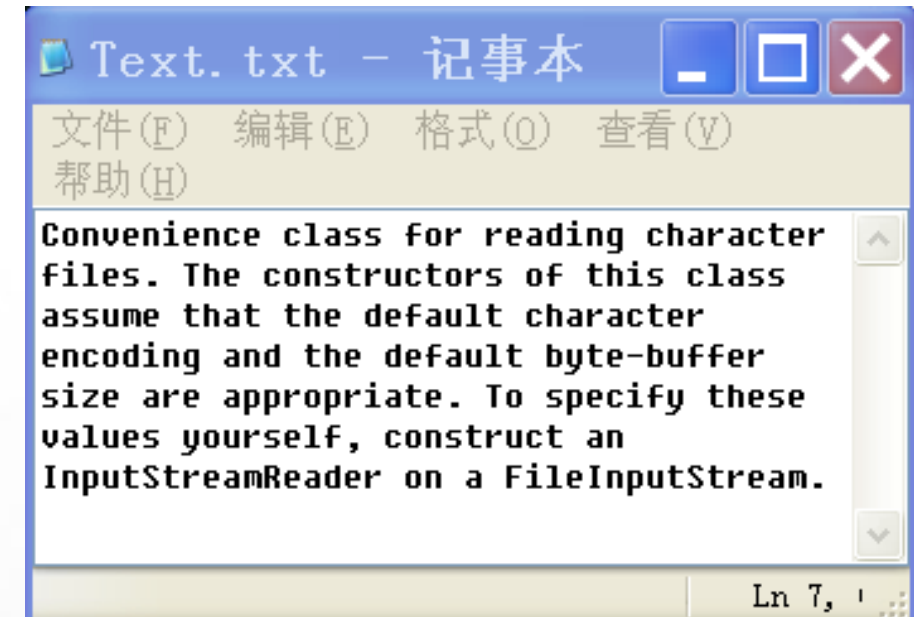
- Encoding
- `getBytes("encodingName")`

```
fos.write("你".getBytes("UTF-16"));  
fos.write("你".getBytes("UTF8"));  
fos.write("你".getBytes());|
```



Character Stream

- **FileReader**
 - Read char from file
- **FileWriter**
 - Write char to file
- **FileReader and FileWriter**
 - use system default encoding
- **Use other encodings:**
 - **InputStreamReader**
 - **OutputStreamWriter**

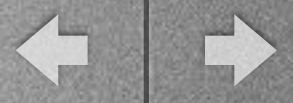


'C' 'o' 'n' 'v' 'e' 'n' 'i' 'e' 'n' 'c' 'e'



Character Stream

- Example:
 - Write following chars
 - Read them out
 - 'C' 'o' 'S' 'E'
 - '软' '件' '学' '院'



Character Stream

```
private void testCharStream() {  
    try {  
        File f = new File("TEST.txt");  
        if (!f.exists()) {  
            f.createNewFile();  
        }  
        FileWriter fos = new FileWriter(f);  
        fos.write("CoSE".toCharArray());  
        fos.write("软件学院".toCharArray());  
        fos.close();  
  
        FileReader fis = new FileReader(f);  
        int b = fis.read();  
        while (b != -1) {  
            System.out.println((char)b);  
            b = fis.read();  
        }  
        fis.close();  
    } catch (IOException e) {  
        System.out.println(e);  
    }  
}
```



Character Stream

- Output

C

O

S

e

软

件

学

院



Byte Stream and Character Stream

- Byte Stream
 - An `int` or `byte[]` can be written to an `OutputStream`;
 - An `int` or `byte[]` can be read from an `InputStream`;
- Character Stream
 - An `int` or `char[]` or `String` can be written to an `Writer`;
 - An `int` or `char[]` or `CharBuffer` can be read from a `Reader`



Byte Stream and Character Stream

- Think
 - How to input a student information into a file?
 - Student ID (int)
 - Name (String)
 - Age (short)
 - Sex (boolean)
 - How to read these information from file?
 - How to store these information in binary or text?



Byte Stream and Character Stream

- Right or Wrong

```
int id = 7111201;  
String name = "直树";  
short age = 34;  
boolean sex = true;  
  
fos.write(id);  
fos.write(name.getBytes());  
fos.write(age);  
fos.write(sex);
```




Byte Stream and Character Stream

- int : 4byte
- short : 2byte
- String : how do we know the length of String
 - no '\0' now
- boolean : convert to byte or String



Byte Stream and Character Stream

- Another Way
 - All convert into String

```
int id = 7111201;  
String name = "直树";  
short age = 34;  
boolean sex = true;  
fw.write(id + " " + name + " " + age + " " + sex + "\n");
```



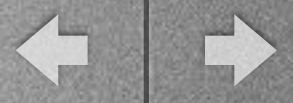
DataInputStream & DataOutputStream

- DataInputStream
 - read(byte [])
 - readBoolean
 - readByte
 - readShort
 - readChar
 - readInt
 - readLine : why not readString()



DataInputStream & DataOutputStream

- DataOutputStream
 - write(byte [])
 - writeBoolean
 - writeByte
 - writeShort
 - writeChar
 - writeChars (be careful)
 - writeInt
 - writeLine



DataInputStream & DataOutputStream

```
private void testWriteChars() {  
    try {  
        DataOutputStream dos = new DataOutputStream(new FileOutputStream("TEXT.txt"));  
        dos.writeChars("你好");  
        dos.write("你好".getBytes());  
        dos.close();  
  
        FileReader fr = new FileReader("TEXT.txt");  
        int b = fr.read();  
        while (b != -1) {  
            System.out.println((char)b);  
            b = fr.read();  
        }  
        fr.close();  
    } catch (Exception e) {  
        System.out.println(e);  
    }  
}
```



RandomAccessFile

- `java.io.RandomAccessFile`
 - `RandomAccessFile` is used for fixed length records
 - Using `seek(long position)` to locate
 - Nothing to do with `InputStream` and `OutputStream`
 - Like `DataInputStream` and `DataOutputStream`
 - Often used for building index of databases

