Streams

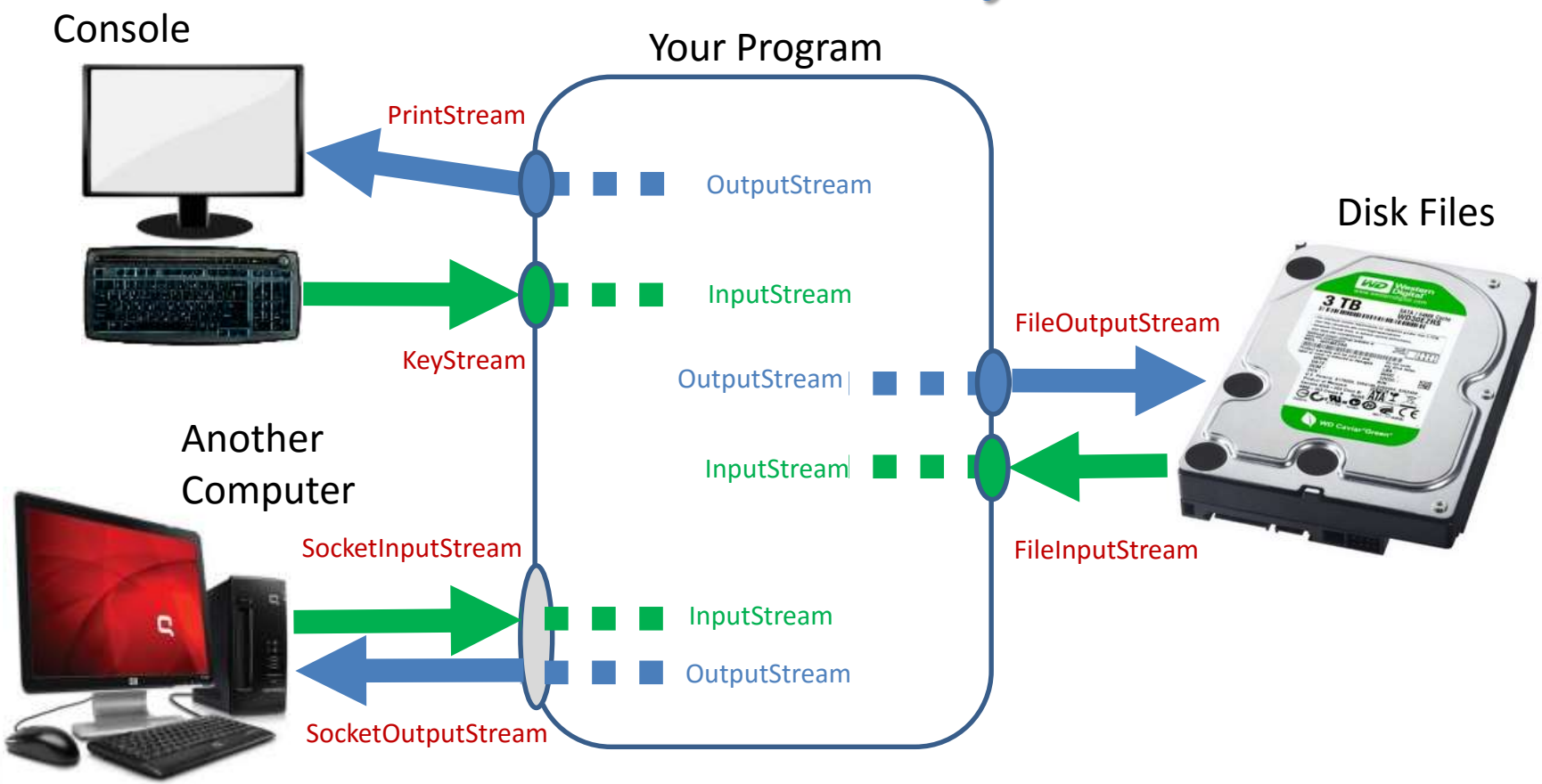
- Bytes
- Characters
- Readers and Writers
- Chaining Streams

See Also

https://docs.oracle.com/javase/tutorial/essential/io/

http://tutorials.jenkov.com/java-io/streams.html

Streams of Bytes



InputStream

```
import java.io.InputStream;

public class Tinker {

    public static void main(String [] args)
{

        InputStream is = System.in;
        is.


    }

}
```

- read(byte[] b, int off, int len) : int - InputStream - 7%
- close() : void - InputStream - 5%
- read() : int - InputStream - 4%
- available() : int - InputStream - 2%
- read(byte[] b) : int - InputStream - 2%
- reset() : void - InputStream - 1%
- equals(Object obj) : boolean - Object
- getClass() : Class<?> - Object
- hashCode() : int - Object
- mark(int readlimit) : void - InputStream
- markSupported() : boolean - InputStream
- ...ject
- ... Object
- ... - InputStream
- ... - Object
- ...ect
- ...) : void - Object
- ..., int nanos) : void - Object

Returns an estimate of the number of bytes that can be read (or skipped over) from this input stream without blocking by the next invocation of a method for this input stream. The next invocation might be the same thread or another thread. A single read or skip of this many bytes will not block, but may read or skip fewer bytes.

Note that while some implementations of `InputStream` will return the total number of bytes in the stream, many will not. It is never correct to use the return value of this method to allocate a buffer intended to hold all data in this stream.

InputStream

```
InputStream is = new FileInputStream("Test.txt");

while(is.available()>0) {
    int i = is.read();
    System.out.println(i);
}

int i = is.read();
System.out.println(i);

is.close();
```

Open

Use

Close

Console | Javadoc | P

<terminated> Tinker (4) [Java Applic

```
111
109
32
74
97
118
97
33
-1
```

*Tinker.java | Test.txt | Test.txt

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	ASCII
00000000:	48	65	6C	6C	6F	20	57	6F	72	6C	64	0D	0A	66	72	6F	Hello World..fro
00000010:	6D	20	4A	61	76	61	21										m Java!

OutputStream

```
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.OutputStream;

public class Tinker {

    public static void main(String [] args) throws IOException {

        OutputStream os = new FileOutputStream("Test2.txt");

        for(int x=0;x<10;++x) {
            os.write(x+65);
        }

        os.flush();
        os.close();

    }

}
```

Open

Use

Close

Test2.txt

1 ABCDEFGHIJ

Test2.txt

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	ASCII
00000000:	41	42	43	44	45	46	47	48	49	4A							ABCDEFGHIJ

Readers and Writers

```
Reader is = new FileReader("Test.txt");

char [] c = new char[5];
is.read(c);

System.out.println(Arrays.toString(c));

is.close();
```

Console | Javadoc | Probler

<terminated> Tinker (4) [Java Application]

[¡, H, a, s, t]

```
InputStream is = new FileInputStream("Test.txt");

byte [] c = new byte[5];
is.read(c);

System.out.println(Arrays.toString(c));

is.close();
```

Console | Javadoc | Problems

<terminated> Tinker (4) [Java Application] C:\Pro

[-95, 72, 97, 115, 116]

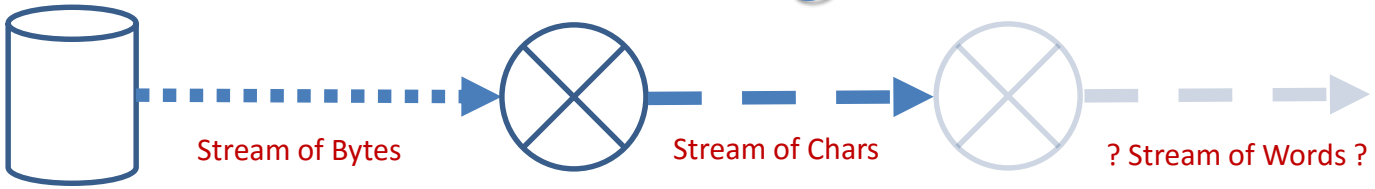
Test.txt

1 ¡Hasta mañana!
2 See you tomorrow

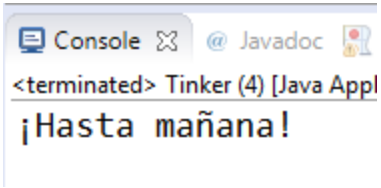
Test.txt

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	ASCII
00000000:	A1	48	61	73	74	61	20	6D	61	F1	61	6E	61	21	0D	0A	◆Hasta ma◆ana!..
00000010:	53	65	65	20	79	6F	75	20	74	6F	6D	6F	72	72	6F	77	See you tomorrow

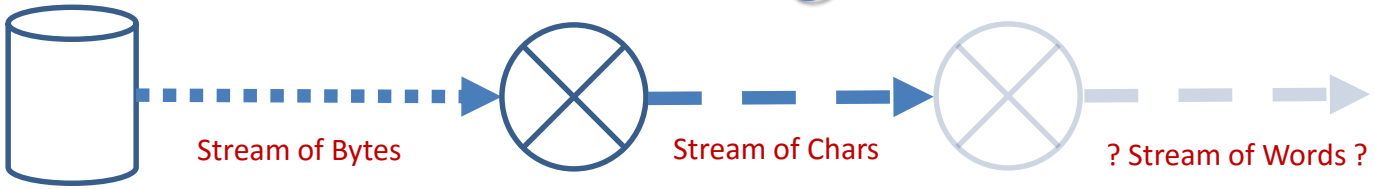
Chaining Streams




```
Reader is = new FileReader("Test.txt");
BufferedReader br = new BufferedReader(is);

String g = br.readLine();
System.out.println(g);

g = br.readLine();
g = br.readLine(); // null

br.close(); // Closes its source
```



Chaining Streams




```
BufferedReader br = new BufferedReader(new FileReader("Test.txt"));
...
br.close(); // Closes its source
```

```
WordReader wr = new WordReader(new BufferedReader(new FileReader("Test.txt")));
...
wr.close(); // Closes its source
```

Readers and Writers

Test.txt

1 ¡Hasta mañana!
2 明天见
3 See you tomorrow
4 להתראות מחר
5

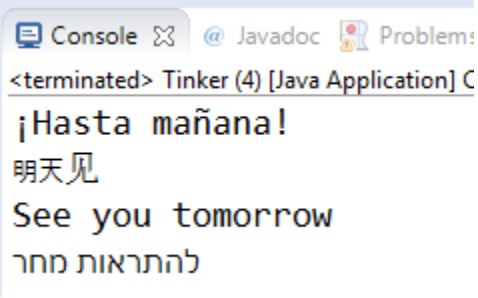
UTF-8

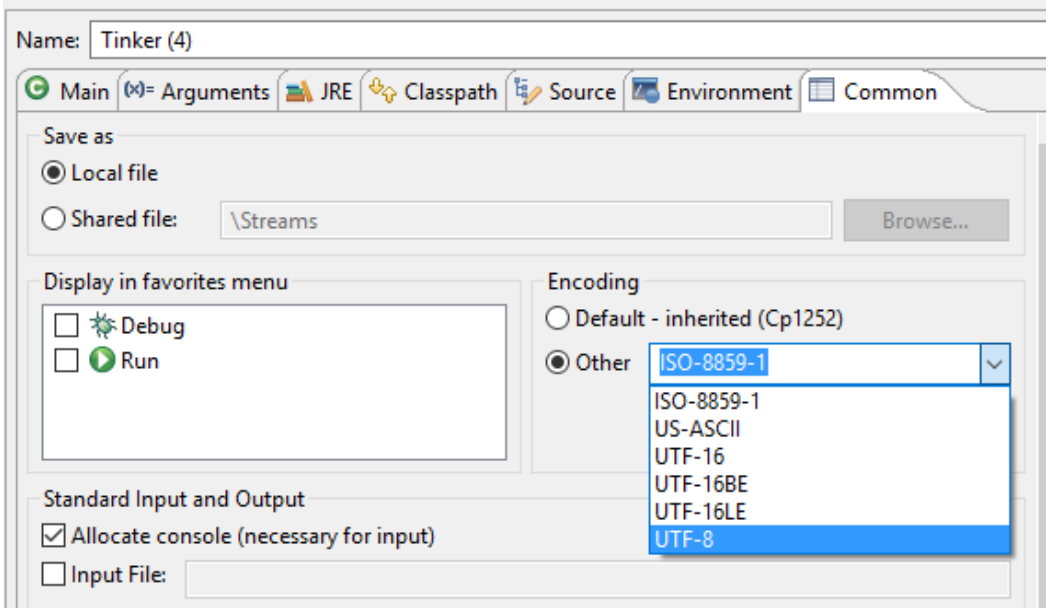
Test.txt

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	ASCII
00000...	C2	A1	48	61	73	74	61	20	6D	61	C3	B1	61	6E	61	21	••Hasta ma••ana!
00000...	0D	0A	E6	98	8E	E5	A4	A9	E8	A7	81	0D	0A	53	65	65	..•••••••••..See
00000...	20	79	6F	75	20	74	6F	6D	6F	72	72	6F	77	0D	0A	D7	you tomorrow..•
00000...	9C	D7	94	D7	AA	D7	A8	D7	90	D7	95	D7	AA	20	D7	9E	•••••••••••••• ••
00000...	D7	97	D7	A8	0D	0A											•••••..

Readers and Writers

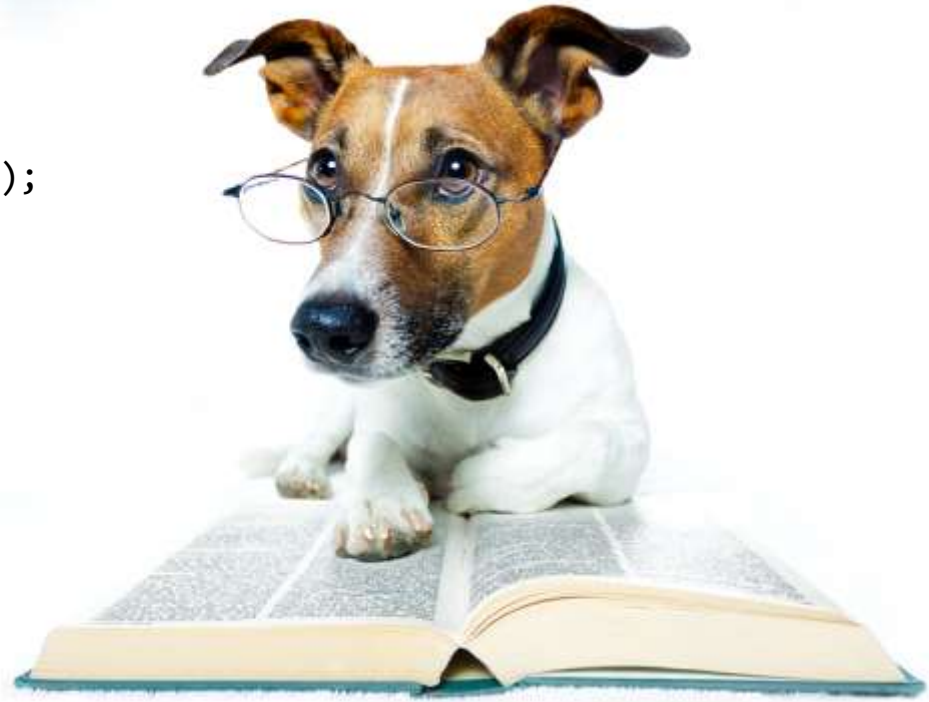
```
InputStream fis = new FileInputStream("TestUTF8.txt"); // Read bytes
Reader r = new InputStreamReader(fis, "UTF8"); // bytes (UTF-8) to chars
BufferedReader br = new BufferedReader(r); // chars to lines

System.out.println(br.readLine());
System.out.println(br.readLine());
System.out.println(br.readLine());
System.out.println(br.readLine());
```





Readers and Writers

```
OutputStream os = new FileOutputStream("Test2.txt");
PrintStream ps = new PrintStream(os);
ps.println("Hello World");
ps.flush();
ps.close();

Writer w = new FileWriter("Test3.txt");
PrintWriter pw = new PrintWriter(w);
pw.println("Hello World");
pw.flush();
pw.close();
```



Tinkering

- Write a program to read a text file and write its capitalized version to another file.
- Handle exceptions without using the try-with-resource from Java7.