

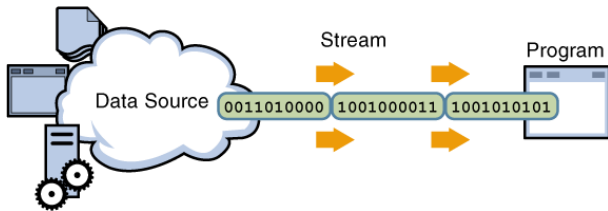
File In- and Output

Sebastian Hollizeck

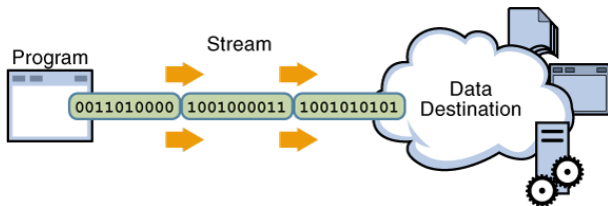
LMU

14. Dez 2010

What is in- and output?



What is in- and output?



IO as Streams in Java

In **JAVA** File input is dealt with like a continuous stream.
The kind of Stream one chooses depends on the purpose.

- ▶ ByteStream raw binaries

IO as Streams in Java

In **JAVA** File input is dealt with like a continuous stream.
The kind of Stream one chooses depends on the purpose.

- ▶ `ByteStream` raw binaries
- ▶ `CharacterStream` translates binaries to char

IO as Streams in Java

In **JAVA** File input is dealt with like a continuous stream.
The kind of Stream one chooses depends on the purpose.

- ▶ ByteStream raw binaries
- ▶ CharacterStream translates binaries to char
- ▶ BufferedStreams wrapper for other streams

IO as Streams in Java

In **JAVA** File input is dealt with like a continuous stream.
The kind of Stream one chooses depends on the purpose.

- ▶ ByteStream raw binaries
- ▶ CharacterStream translates binaries to char
- ▶ BufferedStreams wrapper for other streams
- ▶ Scanner/Format for a formatted text

IO as Streams in Java

In **JAVA** File input is dealt with like a continuous stream.
The kind of Stream one chooses depends on the purpose.

- ▶ ByteStream raw binaries
- ▶ CharacterStream translates binaries to char
- ▶ BufferedStreams wrapper for other streams
- ▶ Scanner/Format for a formatted text
- ▶ CommandLine

IO as Streams in Java

In **JAVA** File input is dealt with like a continuous stream.
The kind of Stream one chooses depends on the purpose.

- ▶ ByteStream raw binaries
- ▶ CharacterStream translates binaries to char
- ▶ BufferedStreams wrapper for other streams
- ▶ Scanner/Format for a formatted text
- ▶ CommandLine
- ▶ DataStream primitive data

IO as Streams in Java

In **JAVA** File input is dealt with like a continuous stream. The kind of Stream one chooses depends on the purpose.

- ▶ ByteStream raw binaries
- ▶ CharacterStream translates binaries to char
- ▶ BufferedStreams wrapper for other streams
- ▶ Scanner/Format for a formatted text
- ▶ CommandLine
- ▶ DataStream primitive data
- ▶ ObjectStream ObjectIO

IO as Streams in Java

In **JAVA** File input is dealt with like a continuous stream.
The kind of Stream one chooses depends on the purpose.

- ▶ ByteStream raw binaries
- ▶ CharacterStream translates binaries to char
- ▶ BufferedStreams wrapper for other streams

- ▶ CommandLine

ByteStream

The byte stream is the most basic stream java provides. Example for input:

Require: file f which contains data

```
while f!=empty do
```

```
    int i ← f.read
```

```
    store i
```

```
end while
```

```
return store
```

CharacterStream

The `CharacterStream` is the one to choose, if you want to read from a text file.

Require: file `f` which contains chars

```
while f!=empty do
```

```
    char c ← f.read
```

```
    print c
```

```
end while
```

BufferedStreams

BufferedStreams are wrapper for other in or outputstreams to:

- ▶ simplify the usage
- ▶ increase the speed
- ▶ provide better observability

The commandline can be used for input or output as well:

Require: someone reacting to the program

while no reaction **do**

do nothing

end while

$x \leftarrow$ reaction.read

do whatEver

Excercises

- ▶ create a program which reads the *xanadu.txt* and prints it to the STDOUT
- ▶ extend the program so that it creates another file named *outPut.txt* which looks exactly like *xanadu.txt*
- ▶ extend the program so that it creates another file named *seperated.txt* which has one word from *xanadu.txt* in each line
- ▶ create a program which downloads the script of this lecture from the homepage.
- ▶ create a program which changes your password via console