

# Java I/O

# Java I/O

Po co nam I/O?

- I/O – najważniejsza część programu
- Bez I/O program praktycznie nie istnieje
- HelloWorld:

```
1 package piotr_tutak;  
2  
3 public class HelloWorld {  
4     public static void main(String[] args) {  
5         System.out.println("Hello World!");  
6     }  
7 }  
8
```

# Java I/O

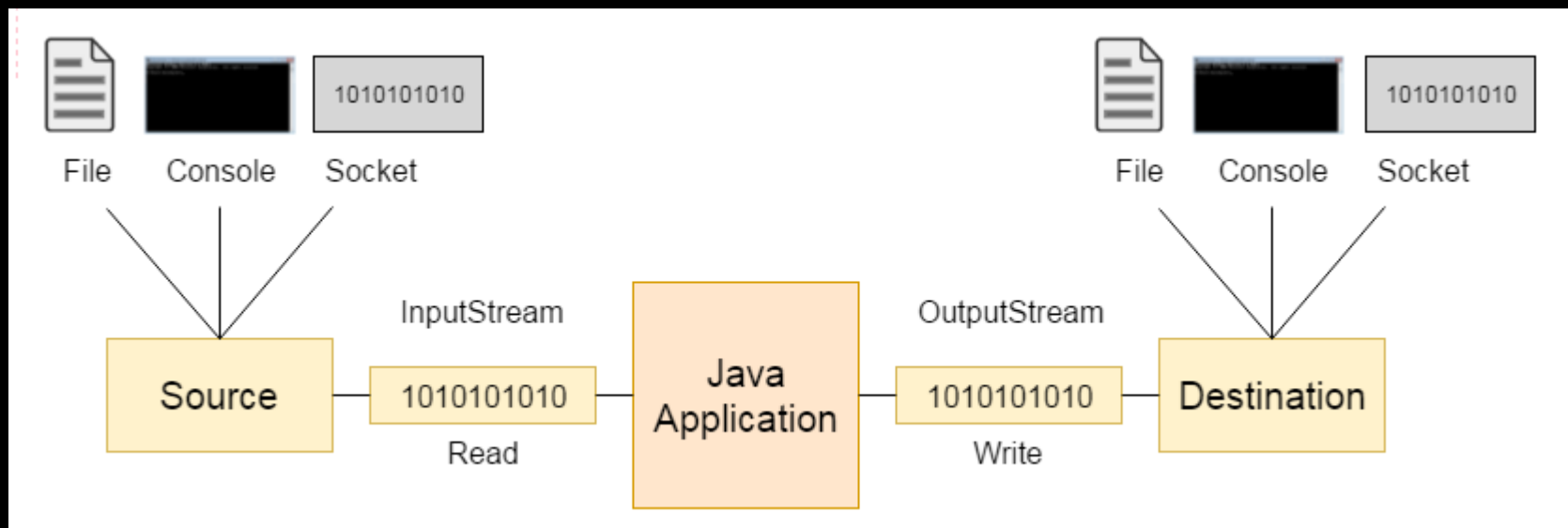
## Typowe źródła danych (Input/Output)

- Pliki
- Rury (Pipes)
- Połączenia sieciowe
- Bufory
- `System.in`, `System.out`, `System.error`

# Java I/O

Podstawowy koncept używany w Java I/O:

- IO Stream – strumień, teoretycznie nieskończone źródło danych wejściowych / wyjściowych z którego możemy czytać / zapisywać dane **sekwencyjnie**



# Java I/O

	Byte Based		Character Based	
	Input	Output	Input	Output
<b>Basic</b>	InputStream	OutputStream	Reader InputStreamReader	Writer OutputStreamWriter
<b>Arrays</b>	ByteArrayInputStream	ByteArrayOutputStream	CharArrayReader	CharArrayWriter
<b>Files</b>	FileInputStream RandomAccessFile	FileOutputStream RandomAccessFile	FileReader	FileWriter
<b>Pipes</b>	PipedInputStream	PipedOutputStream	PipedReader	PipedWriter
<b>Buffering</b>	BufferedInputStream	BufferedOutputStream	BufferedReader	BufferedWriter
<b>Filtering</b>	FilterInputStream	FilterOutputStream	FilterReader	FilterWriter
<b>Parsing</b>	PushbackInputStream StreamTokenizer		PushbackReader LineNumberReader	
<b>Strings</b>			StringReader	StringWriter
<b>Data</b>	DataInputStream	DataOutputStream		
<b>Data - Formatted</b>		PrintStream		PrintWriter
<b>Objects</b>	ObjectInputStream	ObjectOutputStream		

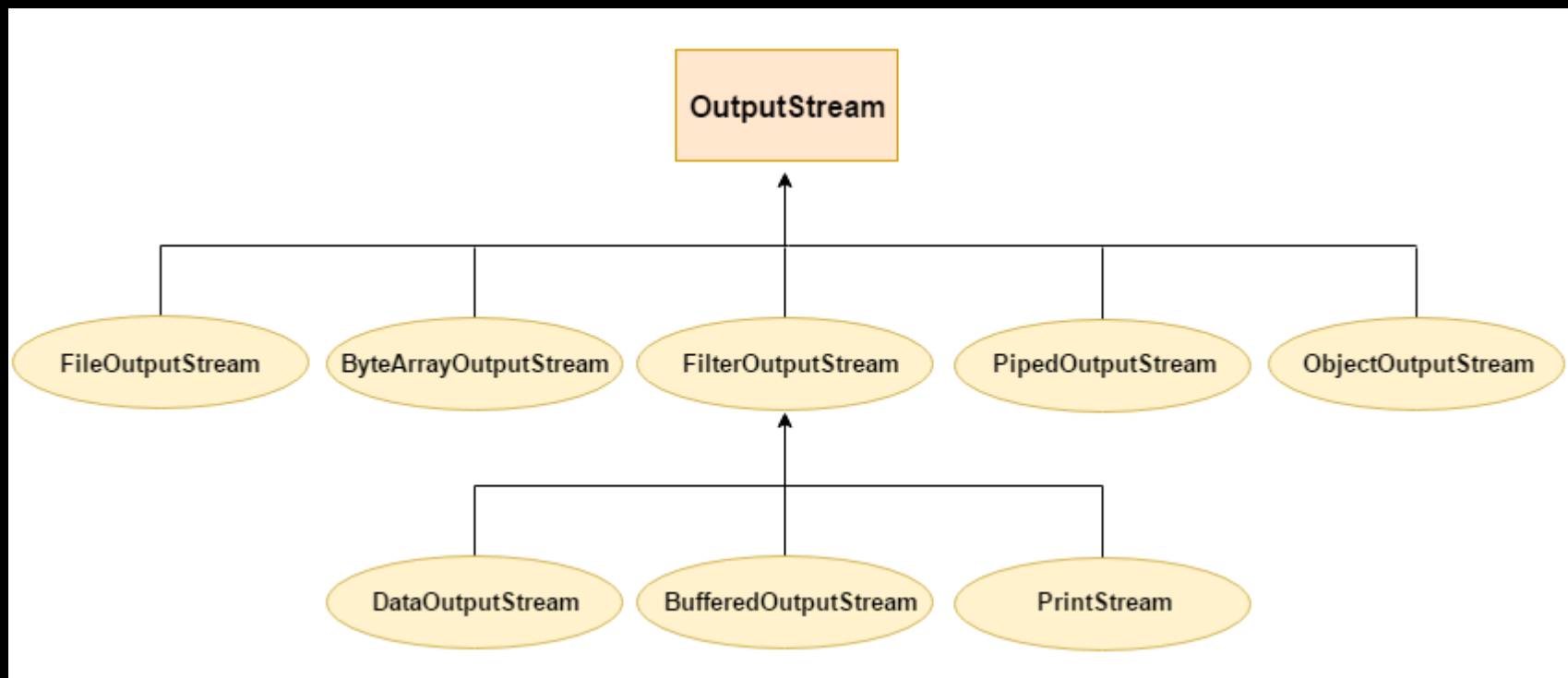
# Java I/O

## Reader/Writer a Input/Output Stream

- Reader/Writer to klasy operujące na znakach – wartość znaku zależy od kodowania i jest to liczba z zakresu [0 ,65535] – przeznaczone specjalnie do obsługi tekstu
- Input/Output Stream to klasy operujące na bajtach – czyli wartościach z zakresu [0,255] niezależnych od niczego

# Java I/O

## OutputStream:



# Java I/O

## InputStream:

