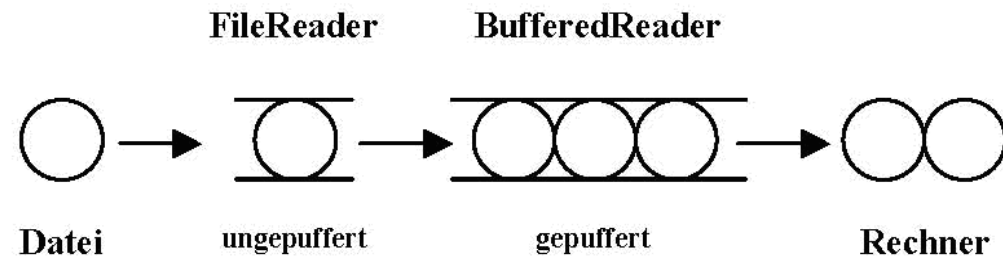
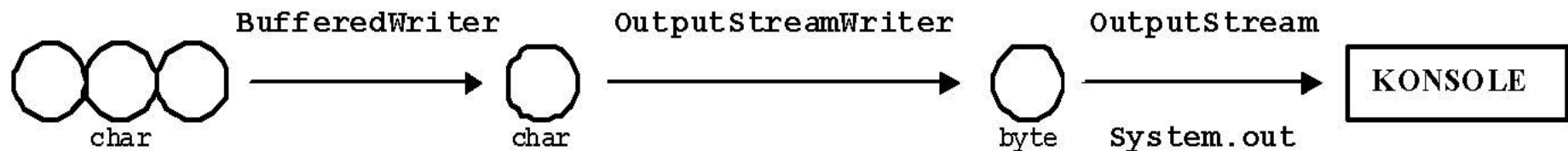


# Wrapping



```
BufferedReader inputStream = new BufferedReader(new FileReader("test.txt"));
```

```
BufferedReader inputStream = new BufferedReader(new InputStreamReader(System.in));
```



```
// Write to console.
```

```
BufferedWriter outputBuffer = new BufferedWriter(new OutputStreamWriter(System.out));
```

```
BufferedWriter outputStream = new BufferedWriter(new FileWriter("test.txt"));
```

## Flushing

- write out a buffer at critical points
- no effect, unless stream is buffered
- manual invocation through stream's "flush" method
- autoflush via optional constructor argument



## Summary

- stream: handles sequential input / output (i/o)
  - byte: InputStream, OutputStream
  - character: Reader, Writer
- unbuffered: direct handling by OS
- buffered: data storage in memory
- wrapping: buffered stream references unbuffered stream
- flush: method used to write out buffer content

# Contents

Introduction

Style Guide

Logging

File and Directory

Data Structure

Stream

Object Serialisation and Persistence

Object Cloning

Date and Time

XML Processing

Concurrency

Meta Programming and Reflexion

Language Binding

Java Native Interface (JNI)





# Example

```
import java.io.*;
public class Launcher {
    public static void main(String[] args) throws Exception {
        FileOutputStream fs = new FileOutputStream("readwrite/test1.ser");
        ObjectOutputStream os = new ObjectOutputStream(fs);
        Time time = new Time(10,20);
        os.writeObject(time);
        os.writeInt(123);
        os.writeObject("Hallo");
        os.writeObject(new Time(11, 25));
        os.close();
    }
}
```

```
public class Time implements Serializable {
    private int hour;
    private int minute;
    public Time(int hour, int minute) {
        this.hour = hour;
        this.minute = minute;
    }
    public String toString() {
        return hour + ":" + minute;
    }
}
```

```
FileInputStream fs = new FileInputStream("readwrite/test1.ser");
ObjectInputStream is = new ObjectInputStream(fs);
Time time = (Time) is.readObject();
System.out.println(time.toString());
System.out.println("" + is.readInt());
System.out.println((String) is.readObject());
Time t = (Time) is.readObject();
System.out.println(t.toString());
is.close();
}
```