

Lesen und Schreiben in Dateien

(File IO)



Stephanie Böhning

Programmieren 2 - Media Systems

Hochschule für Angewandte Wissenschaften Hamburg

Motivation

- Bei Textbearbeitungsprogrammen wie Word möchte man den Text in einer Datei speichern, so dass der User jeder Zeit weiter arbeiten kann
- Bei Spielen soll der Spielstand gesichert werden
- Fast alle Programme haben Daten, die gesichert werden sollen

Übersicht der Schreibvarianten

Möglichkeiten zum Schreiben:

- FileWriter (zeichenorientiert)
- FileOutputStream (byteorientiert)
- BufferedWriter
- PrintWriter
- ObjectOutputStream (Objekte müssen serialisierbar sein!)

Übersicht der Schreibvarianten

Möglichkeiten zum Schreiben:

- FileWriter (zeichenorientiert)
- FileOutputStream (byteorientiert) **Warum? Beispiele?**
- BufferedWriter
- PrintWriter
- ObjectOutputStream (Objekte müssen serialisierbar sein!)

Übersicht der Schreibvarianten

Möglichkeiten zum Schreiben:

- FileWriter (zeichenorientiert)
- FileOutputStream (byteorientiert) : z.B. Mp3
- BufferedWriter
- PrintWriter
- ObjectOutputStream (Objekte müssen serialisierbar sein!)

Übersicht der Lesevarianten

Möglichkeiten zum Lesen:

- FileReader
- FileInputStream
- BufferedReader
- Scanner
- ObjectInputStream

FileWriter

- Ist ein „Low-Level“-Writer
- Schreibt zeichenweise in eine Datei
- Benötigt entweder den Pfad zu einer Datei
- Kann eine Datei überschreiben oder einfach Text in der Datei anhängen (append-Flag)

FileWriter

```
FileWriter out = null;  
try {  
    out = new FileWriter(file, false);  
    out.write("Huhu");  
    out.close();  
} catch (IOException e) {  
    // TODO: handle exception  
}
```



Achtung
Konstruktor wirft
IOException

FileOutputStream

- Ist ein „Low-Level“-Writer
- Schreibt byteweise in eine Datei
- Benötigt entweder den Pfad zu einer Datei
- Kann genauso wie der FileWriter eine Datei überschreiben oder einfach Text in der Datei anhängen (append-Flag)

FileOutputStream

```
try {  
    FileOutputStream fos = new  
        FileOutputStream(file, true);  
  
    fos.write(byteArray);  
    fos.flush();  
    fos.close();  
  
} catch (FileNotFoundException e) {  
    // TODO: handle exception  
}  
catch (IOException e) {  
    // TODO: handle exception  
}
```

BufferedWriter

Baut auf dem FileWriter auf

```
FileWriter out = new FileWriter(file);  
BufferedWriter writer = new  
                        BufferedWriter(out);  
  
writer.write("Huhu");  
writer.newLine();  
  
//beide Writer schließen
```


PrintWriter

Baut auf dem FileWriter auf

```
FileWriter out = new FileWriter(file);  
PrintWriter writer = new PrintWriter(out);  
  
writer.println("Huhu");  
  
writer.close();  
out.close();
```

ObjectOutputStream

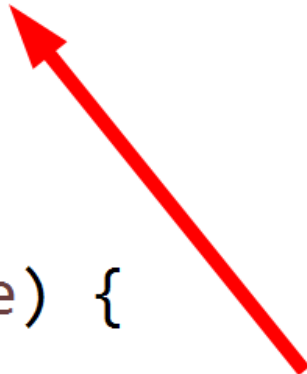
```
try {
```

```
    FileOutputStream fos = new FileOutputStream(file);  
    ObjectOutputStream oos = new ObjectOutputStream(fos);  
    Message msg = new Message("Stephi", "Alex", "Huhu");  
    oos.writeObject(msg);
```

```
} catch (Exception e) {  
    e.printStackTrace();  
}
```

FileReader

```
try {  
    FileReader in = new FileReader(file);  
    //read data here  
    in.close();  
} catch (FileNotFoundException e) {  
    // TODO: handle exception  
}
```



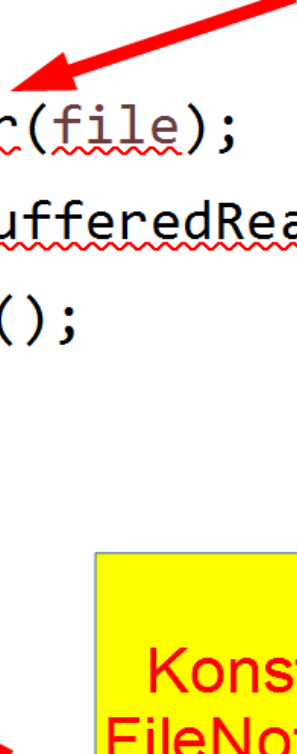
Achtung
Konstruktor wirft
FileNotFoundException

FileInputStream

```
try {  
    FileInputStream fis = new  
        FileInputStream("test.txt");  
    numberBytes = fis.read (byteArray);  
    fis.close();  
} catch (FileNotFoundException e) {  
    // TODO: handle exception  
} catch (IOException e) {  
    // TODO: handle exception  
}
```

BufferedReader

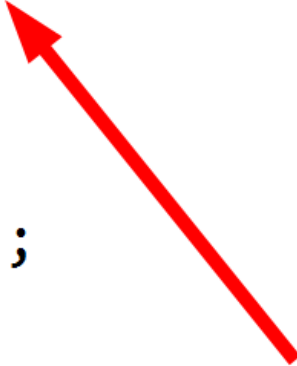
```
try {  
    FileReader in = new FileReader(file);  
    BufferedReader reader = new BufferedReader(in);  
    String line = reader.readLine();  
    while (line != null) {  
        text += line;  
        line = reader.readLine();  
    }  
    //close Reader  
} catch (FileNotFoundException e) {  
} catch (IOException e) {}
```



Konstruktor wirft die
FileNotFoundException
und
readLine() die
IOException

Scanner

```
try {  
    Scanner scan = new Scanner(file);  
  
    while (scan.hasNextLine()) {  
        text = scan.nextLine() + "\n";  
    }  
  
    scan.close();  
} catch (FileNotFoundException e) {  
    // TODO: handle exception  
}
```



Achtung
Konstruktor wirft
FileNotFoundException

ObjectInputStream

```
FileInputStream fis = new FileInputStream(file);  
ObjectInputStream ois = new ObjectInputStream(fis);
```

```
Object obj = ois.readObject();
```

//Typprüfung vor dem Casten, Achtung es gibt auch
Haker!

```
if (obj instanceof Message) {  
    Message msg = (Message) obj;  
    System.out.println(msg.toString());  
}
```

Gibt es Fragen?

