Java

Prof. Dr. Stefan Enderle
NTA Isny

4. Eingabe von Text und Zahlen

Eingabe mit Streams

- Für die Eingabe werden sogenannte **Streams** benutzt.
- Ein Stream ist eine FIFO-Queue.
- Der Stream, der von der Tastatur einliest nennt sich **`System.in`**.
- Bei jedem Aufruf von `read()` erhält man das nächste gespeicherte Objekt:

```
a = System.in.read();
```

- (Allerdings ruft man `read()` meistens nicht selbst auf...)

InputStreamReader

- `System.in` ist ein **Byte-Stream**,
  d.h. die gespeicherten Objekte sind einzelne Bytes.

- Um diese Bytes in Zeichen (mit jeweils 2 Bytes) umzuwandeln, benutzt man einen **`InputStreamReader`**.

- Beispiel:

```
InputStreamReader sr;
sr = new InputStreamReader(System.in);
```

- (Aber auch hier ruft man `read()` meistens nicht selbst auf...)

BufferedStreamReader

- `InputStreamReader` liefert jedoch nur einzelne Zeichen.

- Um ganze Strings einzulesen, benutzt man einen `BufferedReader`.

- Beispiel:

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

Eingabe von Zeichen

- Um ein einzelnes Zeichen einzulesen, kann die Methode `read()` von `BufferedReader` benutzt werden:

- Beispiel:

```
char zeichen;
zeichen = in.read();
```

Eingabe von Strings

- Um eine komplette Zeile einzulesen, kann die Methode `readline()` von `BufferedReader` benutzt werden:

- Beispiel:

```
String zeile;
zeile = in.readline();
```

Eingabe von Ganzzahlen

- Um einen `int` einzulesen, muss man einen eingelesenen String in einen int umwandeln.

- Hierzu kann die Klasse `Integer` benutzt werden, die einige Hilfsfunktionen rund um Integers enthält.

- Beispiel:

```
String zeile;
zeile = in.readline();
int i = Integer.parseInt(zeile);
```

Eingabe von Gleitkommazahlen

- Einen `float` einzulesen, geschieht analog zum `int`: String einlesen und in einen float umwandeln.

- Ganz analog kann hier die Klasse `Float` benutzt werden, die einige Hilfsfunktionen rund um Floats enthält.

- Beispiel:

```
String zeile;
zeile = in.readline();
float f = Float.parseFloat(zeile);
```

Exceptions

- Fehler, die zum Beispiel beim Einlesen von Zeichen oder Umwandeln eines Strings in int auftreten können, werden in JAVA mit ***Exceptions*** behandelt.

- Hierzu existiert ein sog. ***try/catch-Mechanismus***.

- Der eigentliche Code befindet sich im ***try-Block***.
- Tritt ein Fehler auf, wird in den ***catch-Block*** gesprungen.
- Allgemein:

```
try {
  // Anweisungen ...
}
catch(...) {
}
```

Beispiel 1

```
try {
  System.out.println( "String eingeben:" );
  BufferedReader in = new BufferedReader(
                 new InputStreamReader(System.in));

  String s = in.readLine();
  System.out.println("Der String ist: " + s);
}

catch (java.io.IOException e) {
}
```

Beispiel 2

```
try {
  System.out.println( "Zahl eingeben:" );
  BufferedReader in = new BufferedReader(
                 new InputStreamReader(System.in));
  String s = in.readLine();
  int    i = Integer.parseInt(s);
  System.out.println("Der int ist: " + i );
}

catch (java.io.IOException e) {
  System.out.println( "IOException");
}
catch (java.lang.NumberFormatException e) {
  System.out.println( "NumberFormatException");
}
```