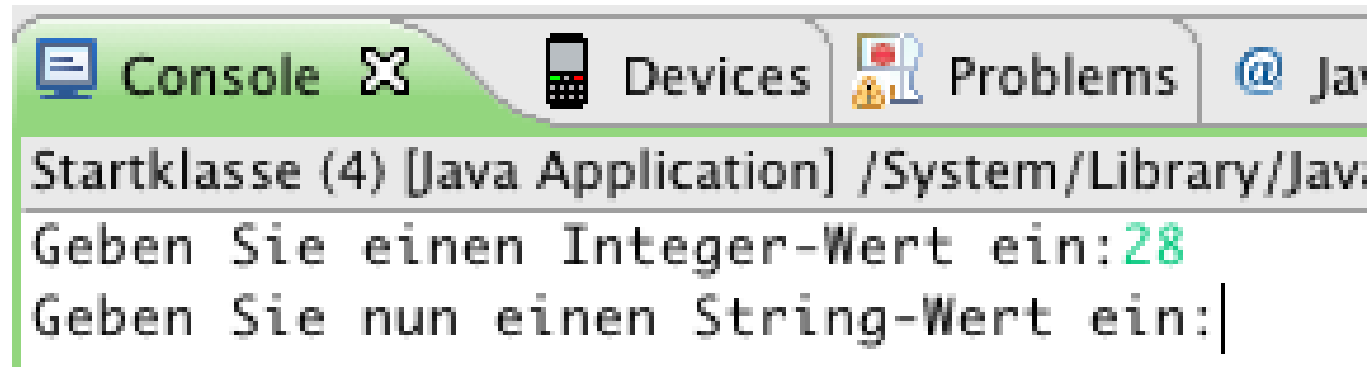


Java: Konsoleneingabe mit der Klasse Scanner

Klasse "Scanner"

Nimmt einfache Texteingaben (auch aus Files) entgegen.



The screenshot shows the Java IDE's console window. The title bar includes tabs for 'Console', 'Devices', 'Problems', and 'Java'. The 'Console' tab is active and highlighted in green. The console content shows the following text: 'Startklasse (4) [Java Application] /System/Library/Java...', 'Geben Sie einen Integer-Wert ein: 28', and 'Geben Sie nun einen String-Wert ein:'. The number '28' is entered in green text, and a cursor is visible at the end of the second prompt line.

```
Startklasse (4) [Java Application] /System/Library/Java...
Geben Sie einen Integer-Wert ein: 28
Geben Sie nun einen String-Wert ein: |
```

Vorgehen: Neues Objekt der Klasse Scanner erzeugen

Vorgehen:

1. Importieren mit `import java.util.Scanner;`
2. Objekt der Klasse Scanner erzeugen;
Parameter ist in unserem Fall `System.in` (Konsoleneingabe)

```
Scanner s = new Scanner(System.in);
```

Methoden der Klasse Scanner

Klasse Scanner stellt zahlreiche Methoden zur Verfügung:

```
Scanner s = new Scanner(System.in);
```

```
s.
```

- **close() : void – Scanner**
- delimiter() : Pattern – Scanner
- equals(Object obj) : boolean – Object
- findInLine(Pattern pattern) : String – Scanner
- findInLine(String pattern) : String – Scanner
- findWithinHorizon(Pattern pattern, int horizon) : String – Scanner
- findWithinHorizon(String pattern, int horizon) : String – Scanner
- getClass() : Class<?> – Object
- hashCode() : int – Object
- hasNext() : boolean – Scanner

Press '^Space' to show Template Proposals

close

```
public void close()
```

Closes this scanner.

If this scanner has not yet been closed, its underlying [readable](#) [Closeable](#) interface the `close` method will be invoked. If the scanner is already closed then invoking this method has no effect.

Press 'Tab' from proposal

(Screenshot mit Autovervollständigungsvorschlägen von Eclipse nach Strg+Space)

Methoden der Klasse Scanner

Erst einmal genügen:

next ()

// liest eine Zeichenkette ein

nextInt ()

// liest einen Integer-Wert ein

Anwendungsbeispiel:

Int-Wert einlesen

```
Scanner s = new Scanner(System.in);  
int eineZahl = s.nextInt();
```

int eineZahl ---- lokale Variable!

Anwendungsbeispiel: Zeichenkette einlesen

```
Scanner s = new Scanner(System.in);  
String eineZeichenkette = s.next();
```

String eineZeichenkette ---- lokale Variable!

Anwendungsbeispiel:

Werte einlesen und verarbeiten

```
Scanner s = new Scanner(System.in);  
System.out.print("Ihr Name: ");  
String name = s.next();  
System.out.print("Ihr Alter: ");  
int alter = s.nextInt();  
System.out.println("Sie sind "+alter+  
    " Jahre alt und heißen "+name+"!");
```


Anwendungsbeispiel:

2 Zahlen einlesen und addieren

```
Scanner s = new Scanner(System.in);  
System.out.print("Zahl 1: ");  
int z1 = s.nextInt();  
System.out.print("Zahl 2: ");  
int z2 = s.nextInt();  
int ergebnis = z1+z2;  
System.out.println(z1+" "+z2+"="+  
    ergebnis+" - richtig?");
```

Zusammenfassung

1. Klasse Scanner importieren (übernimmt i.d.R. IDE)
2. Neues Objekt d. Klasse Scanner anlegen
3. Methoden `next()` bzw. `nextInt()` verwenden.

ACHTUNG!!!!!!

Die Klasse Scanner gibt es schon.

Sie sollten also keine Klasse
„Scanner“ nennen. Lieber
„Eingabe“ oder „ScannerEingabe“
o.ä.

Anwendungsübung Scanner

Schreiben Sie ein Programm, wo man über die Konsole seinen Namen und seinen Kontostand eingeben kann. Wenn der Kontostand negativ ist, erhält man eine Zahlungsaufforderung ...

„Heini, bezahlen Sie 2500 Euro zurück!!!“

(Math.abs für absoluten Betrag)

... ansonsten wird man freundlich mit Name und Kontostand begrüßt:

„Heini, guten Tag! Viel Spaß mit Ihren 2500 Euro.“