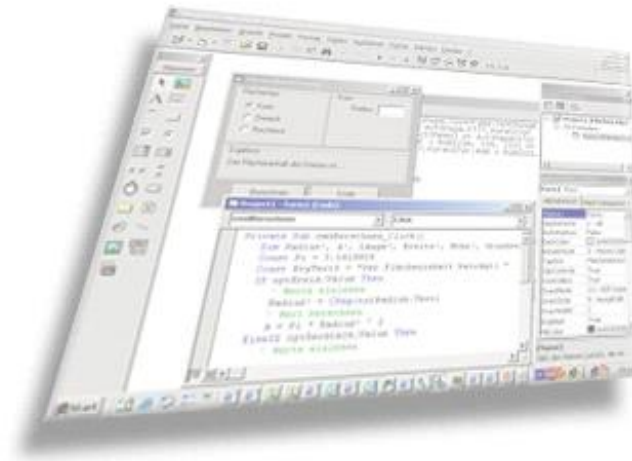
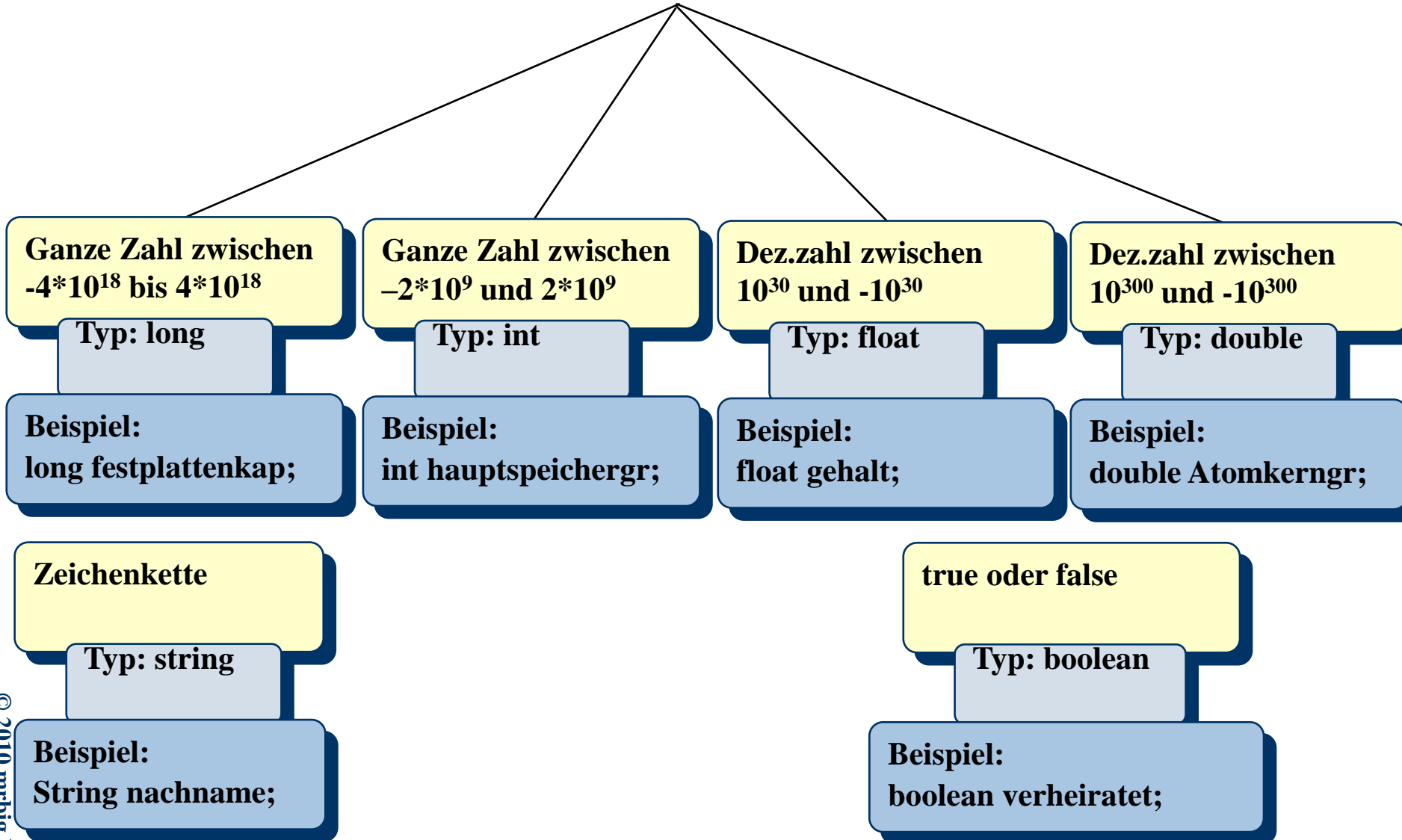


Datentypen & Zeichenketten



Datentypen I





Deklaration von Variablen

Beispiel:

```
String Name;  
float Gehalt;  
int Schuhgroesse;  
long Kontostand;  
boolean verheiratet;
```

```
Name = "Hello World!";  
Gehalt = 5600.45f;  
Schuhgroesse = 34;  
Kontostand = 125000;  
verheiratet = true;
```



Regeln für Variablennamen

- ✓ Maximal 255 Zeichen
- ✓ Müssen mit einem Buchstaben beginnen
- ✓ Dürfen **keine Leerzeichen** enthalten
- ✓ Dürfen **nicht Schlüsselwörter** der Prog.sprache sein (z.B. System, print usw.)
- ✓ Dürfen **keinen Punkt** enthalten

Beispiele:

Gehalt, Konto78, test_45, MainIt, PublicClass, public

Falsch wäre: 5test, Garfield&diePeanuts, Peter Pan,
Mr.Big, Public, Static

Verbinden von Zeichenketten

Zeichenketten: "Hallo!", "3", "3XYZ"

```
String c,d,e;  
d="3";  
c="2";  
e=c+d;  
System.out.println("c+d:"+e);
```

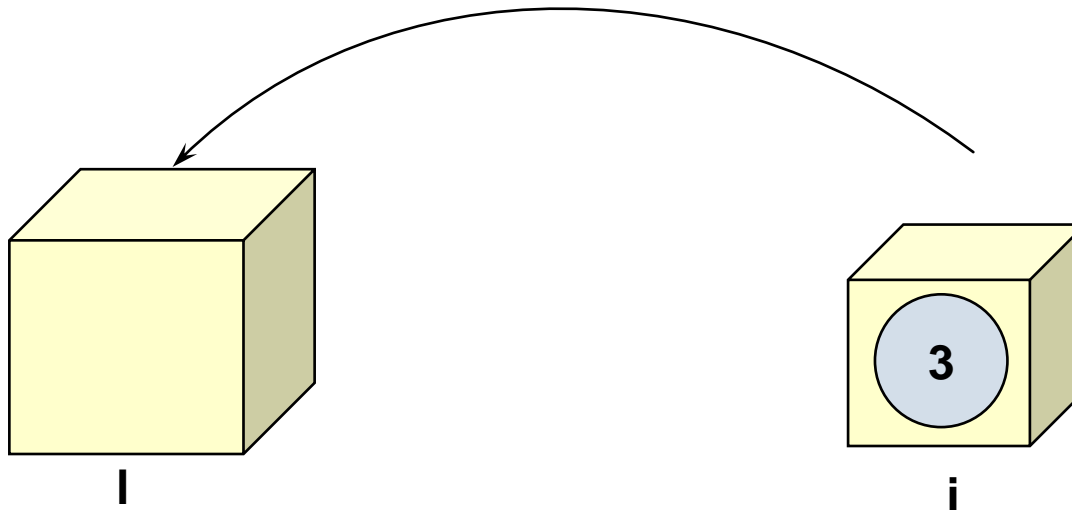


A screenshot of a Windows command prompt window. The title bar reads "C:\WINDOWS\system32\cmd.exe". The window content shows the output of a Java program: "c+d:23". Below this, the current directory is shown as "C:\Documents and Settings\MrBig\workspace\Fir" and the prompt "Press any key to continue . . ." is displayed.

Datentypkonvertierung – Zahlen I

```
int i=3; long l=4;  
l= i;
```

Automatisch möglich, da eine Variable vom Datentyp „long“ problemlos jeden Wert von einer Variable von dem Datentyp „int“ aufnehmen kann!



Automatische Konvertierung in dieser Richtung möglich:
byte -> short -> int -> long -> float -> double

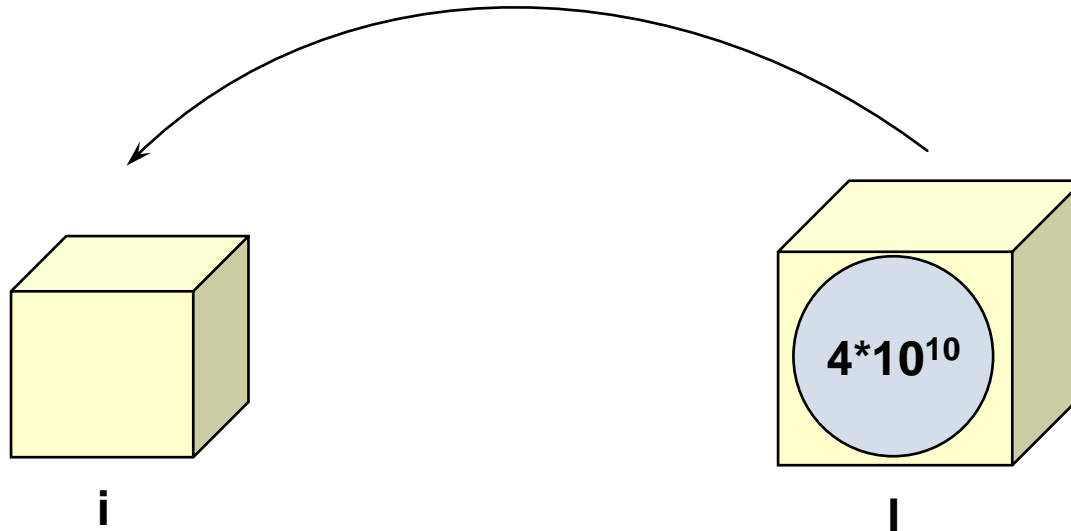


Datentypkonvertierung – Zahlen II

```
int i=3; long l=400000000;
```

```
i = (int) l;
```

Automatisch **nicht** möglich, da eine Variable vom Datentyp „long“ **nicht** jeden Wert von einer Variable von dem Datentyp „int“ aufnehmen kann!





Datentypkonvertierung – Zahlen III

```
int big = 1234567890;  
float approx = big;  
big= (int) approx;  
System.out.println(big);
```

Ausgabe: 1234567936

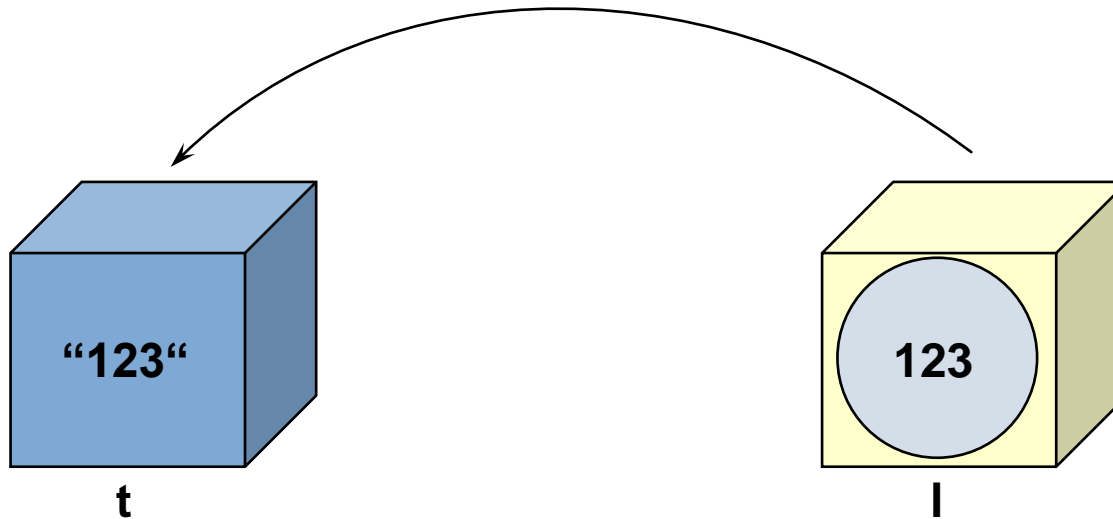
Differenz: 46

Grund: Ungenaue Konvertierung von int nach float

(Float hat nicht genug Nachkommastellen!)

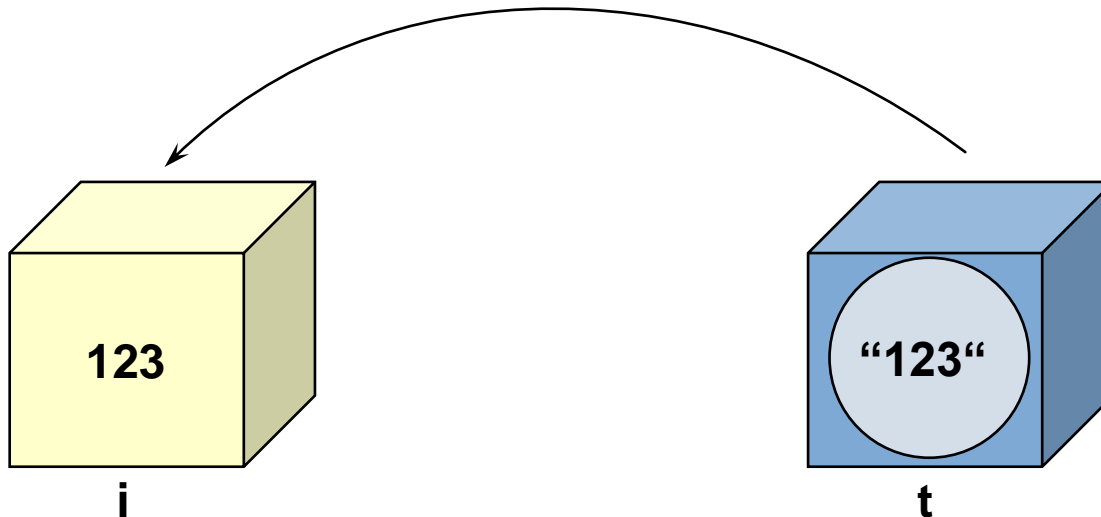
Datentypkonvertierung I

```
String t;  
int i=123;  
t =Integer.toString(i);  
// alternativ:  
t =String.valueOf(i);
```



Datentypkonvertierung II

```
int i;  
String t="123";  
i =Integer.parseInt(t);
```



Datentypkonvertierung II

```

1 import java.lang.Integer;
2
3 public class test {
4
5     public static void main(String[] args) {
6         int Zahl;
7         String Text="NotANumber!";
8
9         Zahl=Integer.parseInt (Text);
10        System.out.print ("Zahl:"+Zahl);
11    }

```



Lösung: Abfangen des Fehlers mit try.....catch!

Laufzeitfehler, wenn sich der Text nicht in eine Zahl konvertieren läßt!

```

1 import java.lang.Integer;
2
3 public class test {
4
5     public static void main(String[] args) {
6         int Zahl;
7         String Text="NotANumber!";
8         try {
9             Zahl=Integer.parseInt (Text);
10        }
11        catch (NumberFormatException e) {
12            Zahl=0;
13        }
14        System.out.print ("Zahl:"+Zahl);
15    }
16 }
17

```



Schlüsselwörter

```
public static void main(String[] args) {  
    int Zahl;  
    String Text="123";  
    Zahl=Integer.parseInt(Text);  
    System.out.print("Zahl:"+Zahl);  
}
```

Die markierten Wörter sind sogenannte **Schlüsselwörter**; diese müssen genauso geschrieben werden (bis auf Groß-/Kleinschreibung) und in der gleichen Syntax (Reihenfolge) verwendet werden.



Die Klasse String

```
String s1 = "Hello World!";
```

Weist der Variablen s1 den Wert „Hello World!“ zu

```
int i = s1.length();
```

Weist der Variablen i den Wert der Länge der Zeichenkette s1 zu:12

```
char c = s1.charAt(6);
```

Weist der Variablen c das 6.te Zeichen von der Zeichenkette s1 zu

```
i = s1.indexOf('e',1);
```

Bestimmt die Stelle, wo das Zeichen „e“ in s1 vorkommt

```
String s3 = "abcdef".substring(2, 5);
```

Ermittelt die Zeichenkette vom 3. bis 5. Zeichen in „abcdef“

```
String s4 = s3.concat("f");
```

Verbindet die Zeichenkette in s3 mit „f“

```
String s5 = String.valueOf(i);
```

Konvertiert den Inhalt von der Variablen i in eine Zeichenkette



Die Methode length

```
int Länge;  
String Text="Peter Pan";  
Länge =Text.length();  
System.out.print("Länge:"+ Länge);
```

Länge:9



Die Methode substr

```
String Text="Peter Pan";  
String TextAuswahl;  
TextAuswahl =Text.substring(6,9);  
System.out.print("7.-9.Zeichen:"+TextAuswahl);
```

7.-9.Zeichen:Pan

Zählweise: Es wird mit „0“ begonnen!

Die Methode indexOf

```
String Text="Peter Pan";  
int LeerZeichenPos;  
LeerZeichenPos =Text.indexOf(' ');  
System.out.println("Pos vom Leerzeichen vom Start:"+LeerZeichenPos);  
LeerZeichenPos =Text.indexOf(" ",6);  
System.out.println("Pos vom Leerzeichen von Pos. 7:"+LeerZeichenPos);
```

1 Leerzeichen!

Pos vom Leerzeichen vom Start:5
Pos vom Leerzeichen von Pos. 7:-1



Vergleiche Ia

```
1 public class Aufgabe {  
2  
3     public static void main(String[] args) {  
4         String JaOderNein;  
5         System.out.println("Ja oder Nein?(j/n):");  
6         JaOderNein =Keyboard.stringInput();  
7         if (JaOderNein=="j")  
8             System.out.println("Jaaaaa!");  
9         else  
10            System.out.println("Nöööööö!");  
11     }  
12 }  
13
```

Funktioniert nicht, da hier die Adressen von JaOderNein und "Ja" verglichen werden!

```
C:\WINDOWS\system32\cmd.exe  
Ja oder Nein?(j/n):  
j  
Nöööööö!  
  
C:\Documents and Settings\MrBig\workspace\Firs  
Press any key to continue . . .
```



Vergleiche Ib

```
1 public class Aufgabe {  
2  
3     public static void main(String[] args) {  
4         String JaOderNein;  
5         System.out.println("Ja oder Nein?(j/n):");  
6         JaOderNein =Keyboard.stringInput();  
7         if (JaOderNein.equals("j"))  
8             System.out.println("Jaaaaa!");  
9         else  
10            System.out.println("Nöööööö!");  
11     }  
12 }  
13
```

```
C:\WINDOWS\system32\cmd.exe  
Ja oder Nein?(j/n):  
j  
Jaaaaa!  
  
C:\Documents and Settings\MrBig\workspace\Firs  
Press any key to continue . . . _
```