

Datentypen

- Lokale und globale Variablen
- Deklaration & Definition
- Typkonvertierung

Deklaration & Definition

- Deklaration
 - `int a;`
- Definition
 - `a=10;`
- Deklaration und Definition
 - `int a=10;`

Schachtelung von Bodies

```
public class InnerOuterBody
```

```
{
```

```
    public static void main(String[] args)
```

```
    {
```

```
        for (int i=0;i<10;i++)
```

```
        {
```

```
            if (true)
```

```
            {
```

```
                //tue irgendwas
```

```
            }
```

```
        }
```

```
    }
```

```
}
```

Äußerster Body

Innerster Body

Variablennamen: Lokal vor Global

```
public class Something {
    static int a=12;

    public static void f()
    {
        int a=10;
        System.out.println(a);
    }
    public static void g()
    {
        System.out.println(a);
    }
}
```

```
public class Something {
    static int a=12;

    public static void f()
    {
        a=10;
        System.out.println(a);
    }
    public static void g()
    {
        System.out.println(a);
    }
}
```

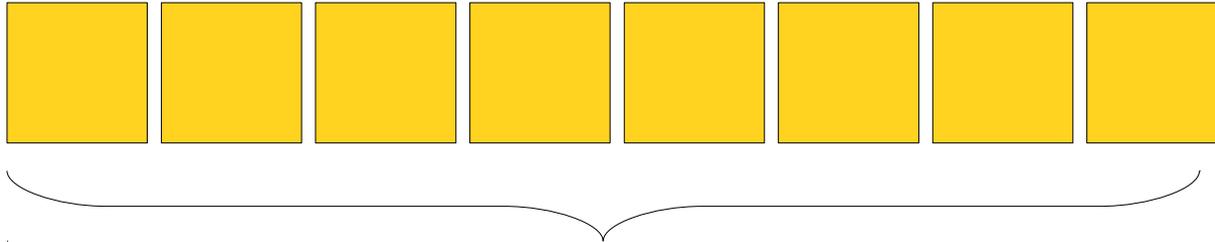
Bei Aufruf von f() und g() aus der Main

- Ausgabe ist 10 und 12, da bei gleichnamigen Variablen die innere Vorrang hat
- Ausgabe ist zweimal 10, da in der inneren Schleife keine neue Variable erzeugt wird, sondern die globale genutzt wird

Ganzzahlen Datentypen

- Beliebig befüllbar mit 8 mal 2 Zuständen = $2*2*2*2*2*2*2*2=256$ Gesamtzustände

1 bit



1 byte=8 bit

Zahlen-Datentypen (Ganzzahl)

- byte 8 Bit $-2^7 .. 2^7-1$ Bereich ± 128
- short 16 bit $2^{15} .. 2^{15}-1$ Bereich ± 32768
- int 32 bit $-2^{31} .. 2^{31}-1$ Bereich ± 2147483648
- long 64 bit $-2^{63} .. 2^{63}-1$ Bereich ± 9223372036854775808
 - Wird verwendet für Datumangaben
`System.currentTimeMillis()`
- boolean 1 bit

Task

- Finden Sie heraus, welche Zahl unter `Integer.MAX_VALUE` gespeichert ist und was passiert, wenn zu dieser Zahl 1 hinzuaddiert wird

Fließkomma-Datentypen

- float 32 bit einfache Genauigkeit
Bereich $\pm 3.4028235 \cdot 10^{38}$
 - Floats im Sourcecode mit Suffix „f“
- double 64 bit doppelte Genauigkeit
Bereich $\pm 1.7976931348623157 \cdot 10^{308}$
 - Doubles im Sourcecode mit Suffix „d“

Task Zufallszahlengenerator

- Erstellen Sie in Pseudocode eine Funktion, die eine Zufallszahl zwischen 2 angegebenen Grenzen erzeugt
 - `Math.random()` erzeugt Zufallszahl in $[0..1]$
- Input: minZahl,maxZahl:int
- Output: int
- Implementieren und testen Sie diese Funktion

Typkonvertierungen

- Datentypen werden in Java nicht automatisch konvertiert, d.h. folgendes geht nicht:
 - Gegeben `int i`; falscher Aufruf `i="hallo"`
 - Gegeben `float f`; falscher Aufruf `f=3.14d`;
- Implizite Konvertierung erfolgt nur bei verlustlosen Zuweisungen z.B. `double d= 3.14f`;
- Typkonvertierungen zwischen primitiven datentypen
 - Syntax (`<datentyp>`) `<ausdruck>`
 - Beispiel `(int) 1.8` konvertiert den double-Wert „1.8“ in einen ganzzahligen Int-Wert durch Abschneiden

Task float versus double

- Verwenden Sie folgende Vorschrift zur iterativen Berechnung einer Zahl mit Hilfe einer Summe:

$$\sum_{i=0}^{1000000} \frac{(-1)^i}{2i+1}$$

- Verwenden Sie zum Speichern der Summe
 - einen float Datentyp und danach einen double Datentyp
 - was ergibt sich als Summe?
 - um wieviel unterscheidet sich das Ergebnis?