

Objektorientierte Programmierung in Java

Vorlesung 2 - Imperative Konzepte

Emily Lucia Antosch

HAW Hamburg

06.10.2024

Inhaltsverzeichnis

1. Einleitung	3
2. Einfache Datentypen	7
3. License Notice	18

1. Einleitung

1.1 Wo sind wir gerade?

- In der Einführung habe ich Ihnen einen Überblick über die Themen der bevorstehenden Vorlesung gegeben.
- Sie haben außerdem Ihr erstes Programm in Java geschrieben!
- Heute geht es um **Imperative Konzepte**.

1.1 Wo sind wir gerade?

1. **Imperative Konzepte**
2. Klassen und Objekte
3. Klassenbibliothek
4. Vererbung
5. Schnittstellen
6. Graphische Oberflächen
7. Ausnahmebehandlung
8. Eingaben und Ausgaben
9. Multithreading (Parallel Computing)

- Wir sprechen über imperative Konzepte in der Programmierung mit Java.
- Sie verstehen die einfachen Datentypen in Java.
- Sie steuern den Programmfluss mit Kontrollstrukturen und Schleifen.
- Sie wenden den korrekten Coding Style an.

2. Einfache Datentypen

? Frage

Wie kann sich sein Program Zustand merken?

? Frage

Wie kann sich sein Program Zustand merken?

- Variablen, die den Zustand im Speicher des Computers speichern.
- Inhalt des Speichers auf dem Computer wird anhand des **Datentyps** interpretiert.

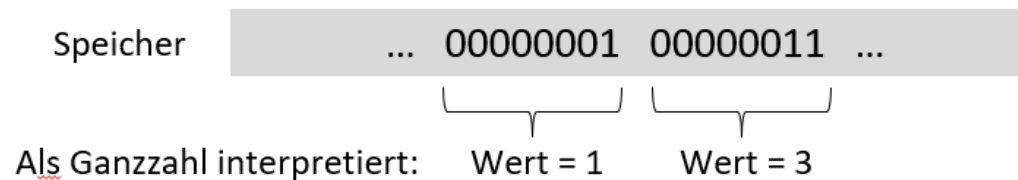


Abbildung 1: Speicher im Computer mit Werten aus dem Programm

? Frage

Welche Datentypen kennen Sie schon aus C?

? Frage

Welche Datentypen kennen Sie schon aus C?

- **int, char, float, double**
- **struct, enum, union**
- **void, bool**
- **Arrays mit [] und Zeiger mit ***

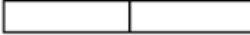
Folgende Datenstrukturen sind in Java verfügbar:

Wahrheitswert:


boolean (1 Bit) 


Ganzzahlen:

byte (1 Byte) 

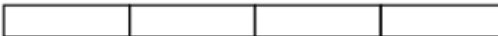
short (2 Byte) 

int (4 Byte) 

long (8 Byte) 

char (2 Byte) 

Gleitkommazahlen:

float (4 Byte) 


double (8 Byte) 

Abbildung 2: Datentypen in Java

2.2 Datentypen in Java

2. Einfache Datentypen

- Speichergrößen und die entsprechenden Wertebereiche:

Art	Datentyp	Größe	Werte	
Ganzzahl	byte	1 Byte	-2^7 bis $2^7 - 1$	entspricht -128 bis 127
	short	2 Byte	-2^{15} bis $2^{15} - 1$	entspricht -32.768 bis 32.767
	int	4 Byte	-2^{31} bis $2^{31} - 1$	
	long	8 Byte	-2^{63} bis $2^{63} - 1$	
(Zeichen)	char	2 Byte	0 bis $2^{16} - 1$	entspricht 0 bis 65.535
Fließkomma	float	4 Byte	$1,4 \cdot 10^{-45}$ bis $3,4 \cdot 10^{38}$	ungefährer Wertebereich
	double	8 Byte	$4,9 \cdot 10^{-324}$ bis $1,8 \cdot 10^{308}$	ungefährer Wertebereich
Wahrheit	boolean	1 Bit	true, false	

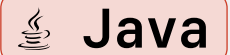
Abbildung 3: Wertebereiche der Datentypen in Java

! Merke

Variablen müssen deklariert werden, bevor sie benutzt werden können.

- Ein Datentyp wird vor dem Variablennamen geschrieben.
- Eine Deklaration könnte so aussehen:

```
1 int a;  
2 float b;  
3 char c;
```



! Merke

Im Anschluss an die Deklaration kann ein Wert zugewiesen werden. Das nennt man Initialisierung.

- Der Variable wird mittels des Zuweisungsoperators = ein Wert zugewiesen:

```
1  a = 5;  
2  b = 3.5;  
3  c = 'A';
```



! Merke

Die Deklaration und Initialisierung kann auch in einem Schritt erfolgen. Das wird dann als Definition bezeichnet.

- Beide Schritte werden direkt hintereinander geschrieben.
- Deklaration und Initialisierung (Definition):

```
1 int a = 5;  
2 float b = 3.5;  
3 char c = 'A';
```



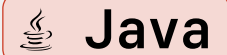
- Variablen haben einen Gültigkeitsbereich, der durch die geschweiften Klammern definiert wird.
- Variablen können an beliebiger Stelle im Code deklariert werden.
- Der Compiler verhindert die Verwendung von Variablen, die nicht initialisiert wurden.

2.7 Typkorrektheit

- Typen müssen korrekt sein, um Fehler zu vermeiden.
 - Anders als in C müssen Werte dem korrektem Datentyp zugewiesen werden.
 - Folgendes würde nicht funktionieren:

```
1 int a = 5;
```

```
2 float b = a;
```



Inkorrekter Typ

A blue arrow originates from the text "Inkorrekter Typ" and points upwards and to the left, ending at the variable 'a' in the second line of code, "float b = a;".

? Frage

Welche Unterschiede sehen Sie zwischen C und Java, wenn es um Datentypen geht?

3. License Notice

3.1 Attribution

- This work is shared under the CC BY-NC-SA 4.0 License and the respective Public License
- <https://creativecommons.org/licenses/by-nc-sa/4.0/>
- This work is based off of the work Prof. Dr. Marc Hensel.
- Some of the images and texts, as well as the layout were changed.
- The base material was supplied in private, therefore the link to the source cannot be shared with the audience.