

Objektorientierte Programmierung in Java

Vorlesung 1 - Organisation und Einführung

Emily Lucia Antosch

HAW Hamburg

04.10.2024

Inhaltsverzeichnis

1. Organisation	4
1.1 Das Ziel dieses Kapitels	5
1.2 Kurz zu mir	6
1.3 Vorlesungsablauf	7
1.4 Vorlesungsfokus	8
1.5 Vorraussetzungen	9
2. Einführung	10
2.1 Das Ziel dieses Kapitels	11
2.2 Themenübersicht: Grundlagen	12
2.3 Themenübersicht: Weiterführende Konzepte	13
2.4 Objekte und Klassen	14

2.5 Datenkapselung	20
2.6 Vererbung	24
2.7 Das erste Programm	26

1. Organisation

1.1 Das Ziel dieses Kapitels

1. Organisation

- Ich will mich bei Ihnen vorstellen und mit Ihnen den Ablauf dieses Moduls besprechen.
- Sie bekommen einen Überblick über die Voraussetzungen in diesem Modul und können diese erfüllen.
- Sie wissen, wie sich mich erreichen können.

- Emily Lucia Antosch, 24 Jahre alt
- Bachelor in Elektro- und Informationstechnik
- Zurzeit tätig als Anwendungsentwicklerin bei NVL
- Nächste Station: Masterstudium Informatik
- Mail: emilylucia.antosch@haw-hamburg.de

i Info

Ich mach das zum ersten Mal, seien Sie also bitte nachsichtig.

- Vorlesungen teilen sich in Termine am Dienstag und Donnerstag auf
 - Am Anfang sind sehr viele Termine, die Sie auf das Praktikum vorbereiten sollen
- Ich würde Sie bitten, sich an der Vorlesung aktiv zu beteiligen
- Es wird kleine Fragen und Aufgaben geben, die Sie live beantworten und mitprogrammieren können

! Merke

Falls Sie etwas nicht verstehen, fragen Sie bitte sofort! Ich wiederhole gerne Inhalte auf Deutsch oder Englisch!

- Wir wollen hier eine Brücke aus Ihren Vorkenntnissen bauen.
- Am Ende der Vorlesung sollten Sie in der Lage sein, in Java einfache Programme zu erstellen.
- Außerdem sollten Sie dann Objektorientierte Programmierung beherrschen und die Unterschiede zu anderen Paradigmen in der Programmierung herausstellen können.
- Den genauen Stoff können Sie außerhalb der Vorlesung auch im Modulhandbuch nachlesen

- Sie brauchen eine Installation des **Java SDK**.
 - Dafür habe ich Ihnen eine Anleitung geschrieben, die Sie im Moodle-Raum finden.
- Außerdem wird die Vorlesung mit dem Tool **JetBrains IntelliJ** sein.
 - Dies ist, wie ich finde, eine sehr gute und einfache IDE für den Anfang.
 - Auch hierfür finden Sie eine Anleitung im Moodle.

2. Einführung

- Sie sollen ihr bekanntes Wissen aus vorhergehenden Vorlesungen auf neue Inhalte anwenden.
- Sie kennen die grundlegenden Ideen der Objektorientierten Programmierung und kennen den Unterschied zu der Programmierung in C.
- Wir erstellen ein einfaches Programm in der Entwicklungsumgebung IntelliJ IDEA und führen dieses aus.

Die ersten Vorlesungen beziehen sich auf die folgenden Prinzipien:

1. Imperative Konzepte
2. Klassen und Objekte
3. Klassenbibliothek
4. Vererbung
5. Schnittstellen

2.3 Themenübersicht: Weiterführende Konzepte 2. Einführung

Aus den Grundlagen wollen wir dann weitere Konzepte ableiten:

6. Graphische Oberflächen
7. Ausnahmebehandlung
8. Eingaben und Ausgaben
9. Multithreading (Parallel Computing)

In der echten Welt werden oft Dinge über ihre Eigenschaften bestimmt und beschrieben:

- Ein Auto hat Eigenschaften wie
 - einen Hersteller
 - eine Farbe
 - einen Verbrauch



Idee

Mithilfe der Objektorientierten Programmierung können wir diesen intuitiven Ansatz auch bei der Programmierung anwenden!

? Frage

Über welche Eigenschaften könnten Sie zum Beispiel eine Person beschreiben? Wie passt das vielleicht in den Programmierungskontext?

? Frage

Über welche Eigenschaften könnten Sie zum Beispiel eine Person beschreiben? Wie passt das vielleicht in den Programmierungskontext?

- Für Studenten:
 - Name, Anschrift, Immatrikulationsnummer
- Für Programme/Webseiten:
 - Username, Passwort, Beitrittsdatum

- Um aus diesem ähnlichen Bauplan dann mehrere gleichartige Objekte zu erstellen, wird eine Klasse erstellt,
 - Sie enthält alle Eigenschaften, die wir gerade definiert haben, also zum Beispiel Variablen
 - Aus ihr lassen sich ganz verschiedene Objekte erstellen, die diese Eigenschaften unterschiedlich gefüllt haben



Beispiel

Aus der Klasse **Student** lassen sich zum Beispiel die beiden Studenten **Max** und **Ines** erstellen, die beide unterschiedlich heißen und eine eigene Immatrikulationsnummer haben.

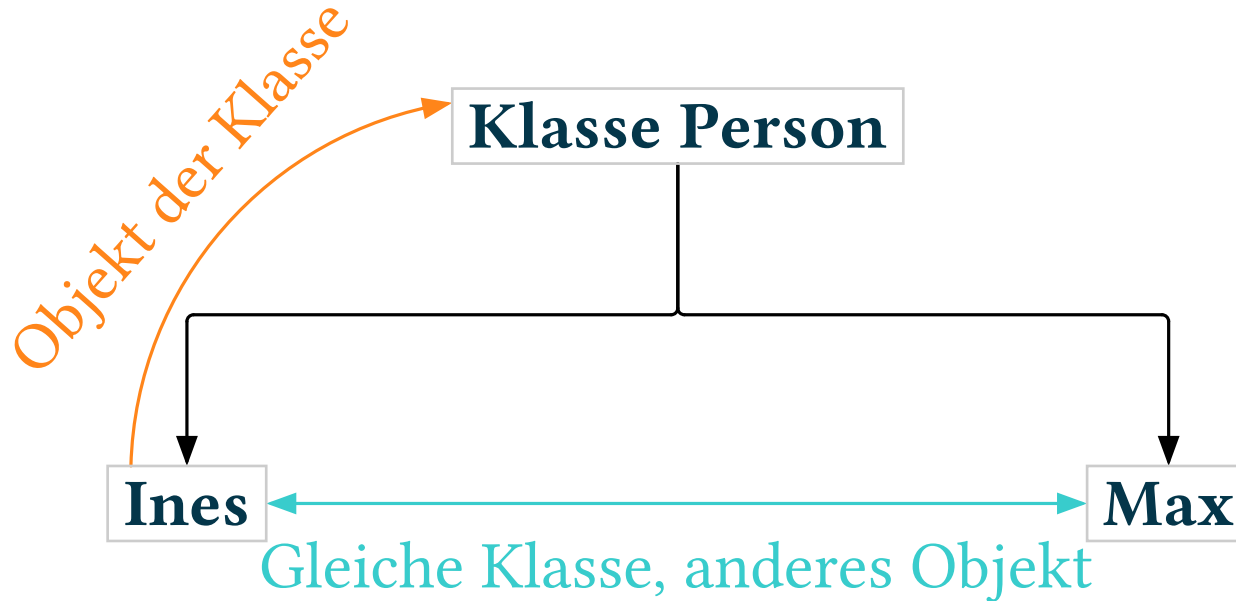


Abbildung 1: Relation zwischen Klassen und Objekten dieser Klasse

- Variablen und Funktionen werden also zu einer Klasse zusammengefasst.
 - Es wird eine Menge an Variablen definiert.
 - Für diese Variablen werden wiederum Funktionen eingeführt, die diese lesen und ändern können.

! Merke

- Variablen werden als **Attribute** bezeichnet.
- Die Werte dieser Variablen beschreiben den **Zustand** des Objekts.
- Funktionen werden als **Methoden** bezeichnet.

- Über sogenannte UML Klassendiagramme lassen sich Klassen mit ihren **Attributen** und **Methoden** beschreiben.

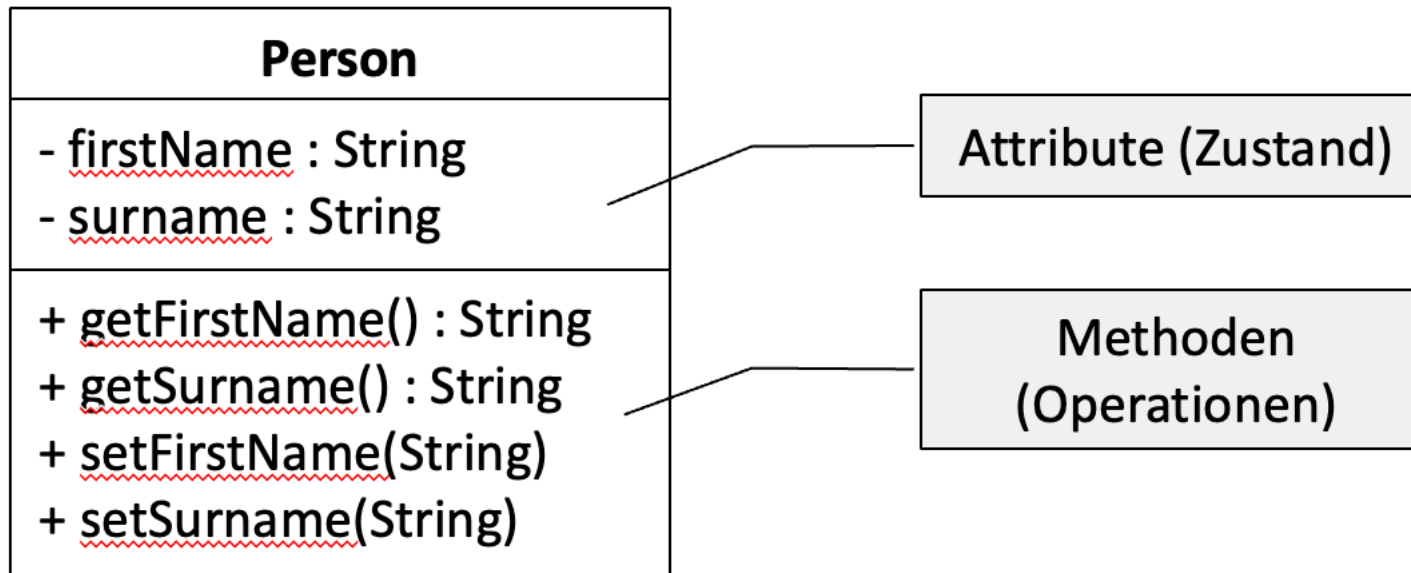


Abbildung 2: UML Klassendiagramm

- Mithilfe von Attributen und Methoden von Klassen lassen sich Daten kapseln.

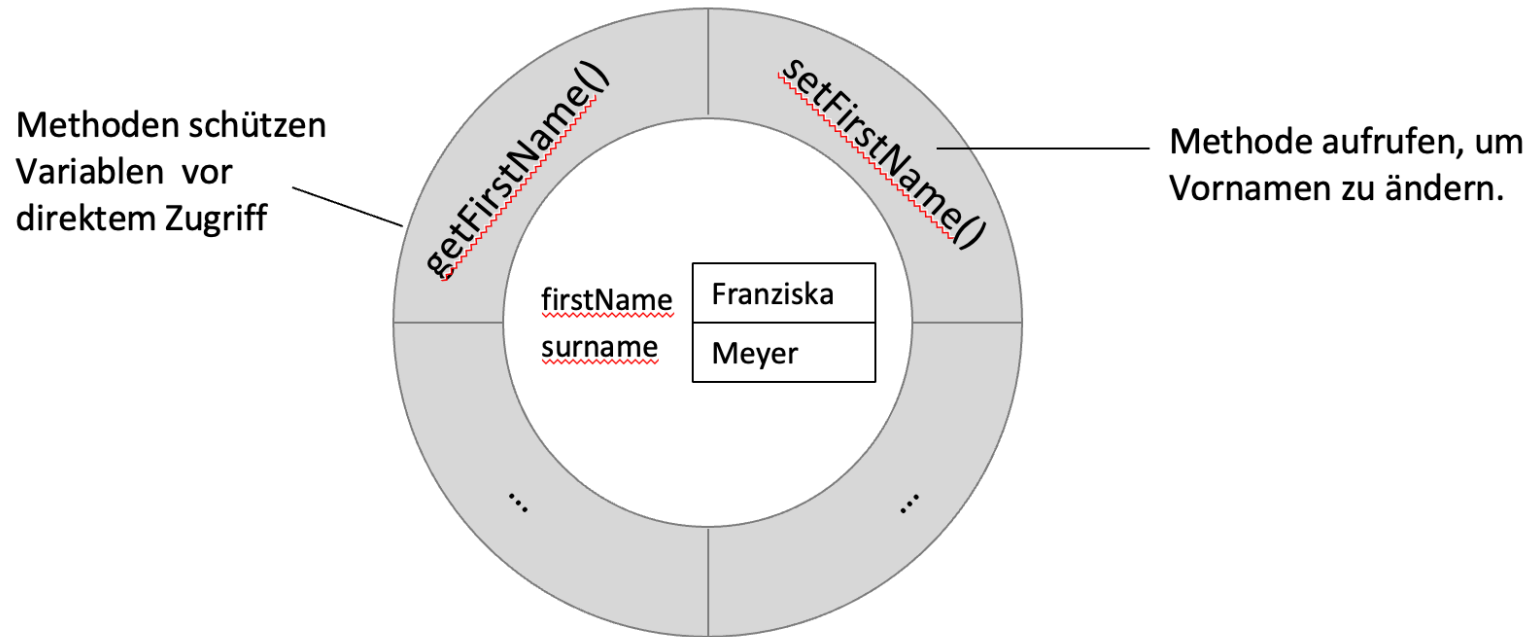
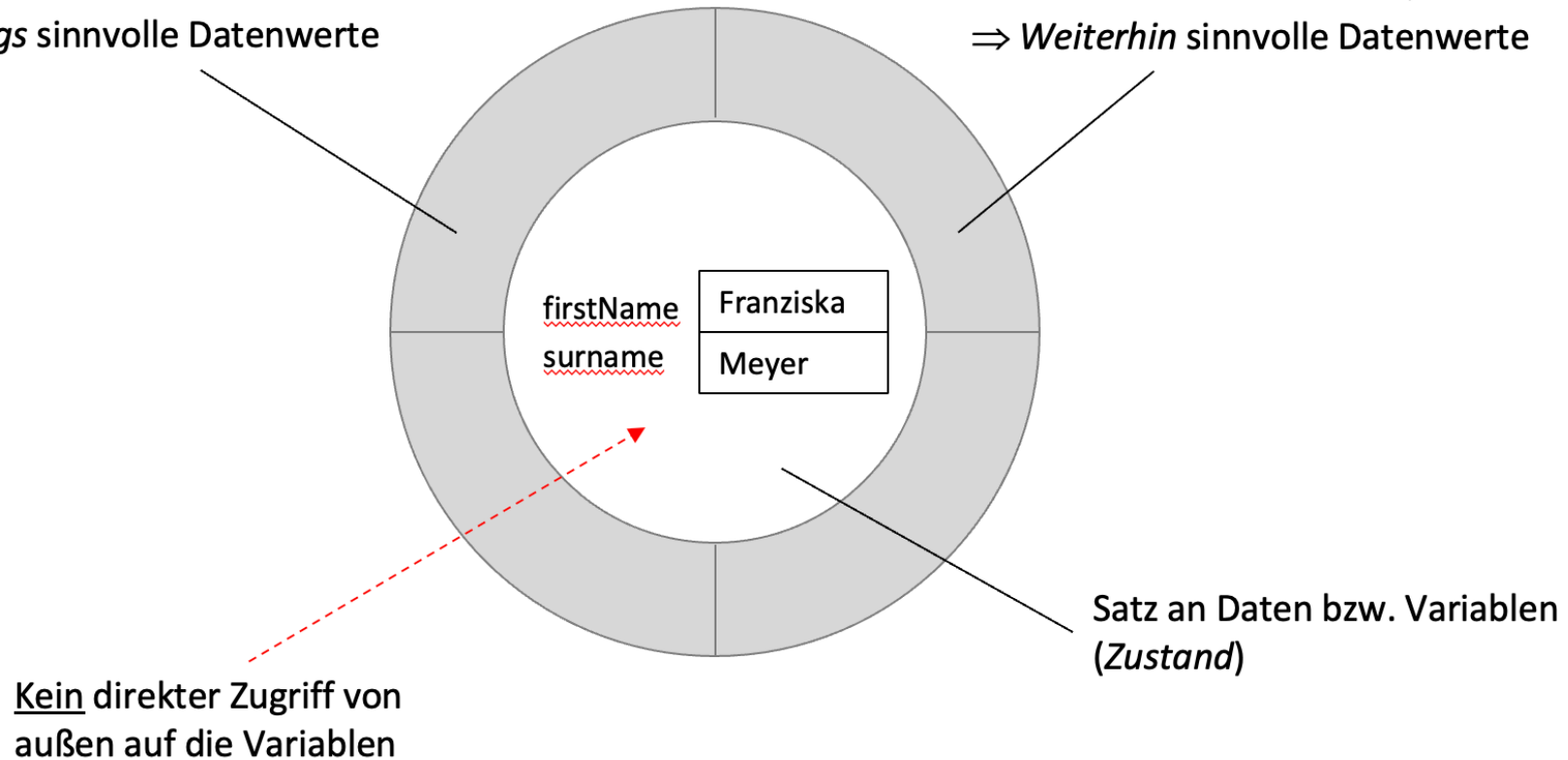


Abbildung 3: Datenkapselung durch Klassen

- Mithilfe von Attributen und Methoden von Klassen lassen sich Daten kapseln.
 - Auf gekapselte Daten können nicht alle Teile des Programms zugreifen, was die Sicherheit erhöht.
 - Außerdem lassen sich so Attribute vor fehlerhaften Werten schützen.

1. Über spezielle Methoden (Konstruktoren)
erzeugt und initialisiert
⇒ *Anfangs* sinnvolle Datenwerte

2. Kontrollierte Änderung der Datenwerte
nur über Methoden des Objektes
⇒ *Weiterhin* sinnvolle Datenwerte



? Frage

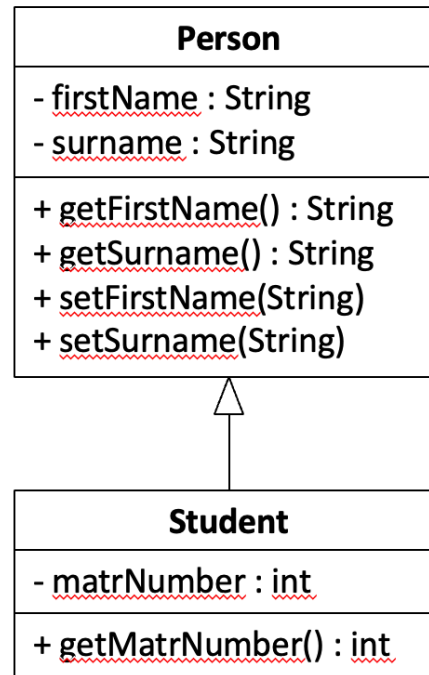
Wie ist das im Gegensatz dazu in C?

? Frage

Wie ist das im Gegensatz dazu in C?

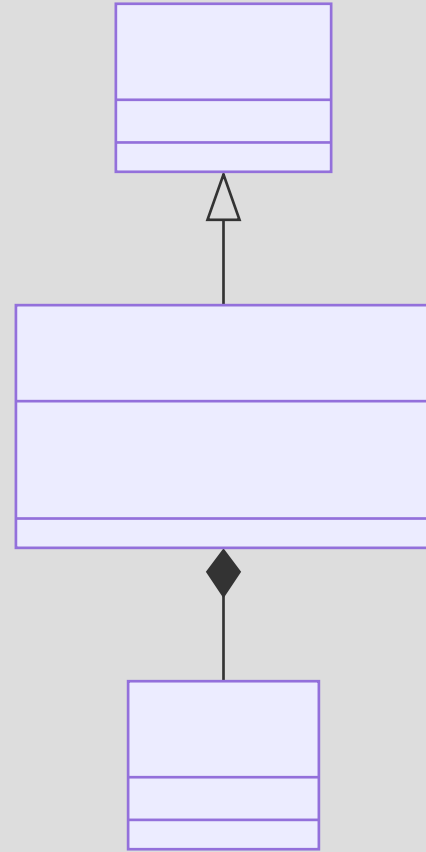
- Die Datenstruktur (also das **struct**) muss für den Zugriff auf die Elemente öffentlich gemacht werden.
- Die Daten werden nicht geschützt.
- Es gibt keine Zuordnung zwischen den Daten und den Funktionen.

- Durch Vererbung lassen sich neue Klassen aus anderen Klassen erzeugen.



- Durch Vererbung lassen sich neue Klassen aus anderen Klassen erzeugen.
- Dabei werden die Methoden und Attribute der Basisklasse übernommen und um weiteren Code erweitert
- Keine duplizierter Code

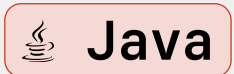
- Hallo Du



```
1 int main (){  
2     printf("Hello World!");  
3     return 0;  
4 }
```



```
1 public static void main(String[] args){  
2     System.out.println("Hello World!");  
3 }
```





Idee

Wenn Sie schon soweit sind, programmieren Sie gerne mit!