

Teil 4: Java-Sprachkonstrukte

Datentypen und Variable

Modul „Grundlagen der objektorientierten Programmierung mit Java“

Prof. Dr. Cornelia Heinisch

Agenda

- Datentypen und Variable
- Referenzvariable
- this-Referenz

Information und ihre Darstellung



Geschwindigkeit:

Ganzzahl

mit Werten von 0, 1, ..., 220

Datentyp: int

Tankfüllstand:

Kommazahl

mit Werten von 0.00, 0.05, ... 1.00

Datentyp: float

Fahrerhinweis:

Zeichenkette

mit Länge (hier 13)

mit Werten in Form von Zeichen (Buchstaben, Ziffern, Sonderzeichen)

Datentyp: "String"

Was ist ein Datentyp und was ist eine Variable?

Datentyp = Bauplan (Blaupause) für Variable



- Anzahl der benötigten Speicherstellen
- Bedeutung (Interpretation) der einzelnen Speicherstellen
- zulässiger Wertebereich

Bauplan Datentyp `int`:

- Anzahl der benötigten Speicherstellen: 4 Byte = 32 Bit
- Bedeutung (Interpretation) der einzelnen Speicherstellen: Zweierkomplementdarstellung, höchstes Bit gibt das Vorzeichen an.
- zulässiger Wertebereich: Ganzzahl von -2^{31} bis $+2^{31} - 1$

Variable des Datentyps `int`:

- Anlegen einer Variablen mit Namen `i` vom Datentyp `int` mit `int i;`
- Zuweisen eines Wertes aus dem Wertebereich mit `int i = 0;`



Aufgabe 1: zwei Zahlen addieren und ausgeben

- Projekt `teil_4` anlegen, Klasse `TestAddieren` mit `main()`-Methode anlegen.
- Lege eine Variable mit dem Namen `wert1` vom Datentyp `int` an.
- Lege eine Variable mit dem Namen `wert2` vom Datentyp `int` an.
- Weise den Variablen zulässige Werte aus dem Wertebereich zu.
- Lege eine Variable mit dem Namen `summe` vom Datentyp `int` an.
- Berechne die Summe der Variablen `wert1` und `wert2` und speichere das Ergebnis in der Variablen `summe`.
- Gib die berechnete Summe auf der Konsole aus.
- Lege eine Variable mit dem Namen `produkt` vom Datentyp `int` an.
- Berechne das Produkt der Variablen `wert1` und `wert2` und speichere das Ergebnis in der Variablen `produkt`.
- Gib das berechnete Produkt auf der Konsole aus.



Aufgabe 2: zwei Zahlen einlesen, addieren und ausgeben

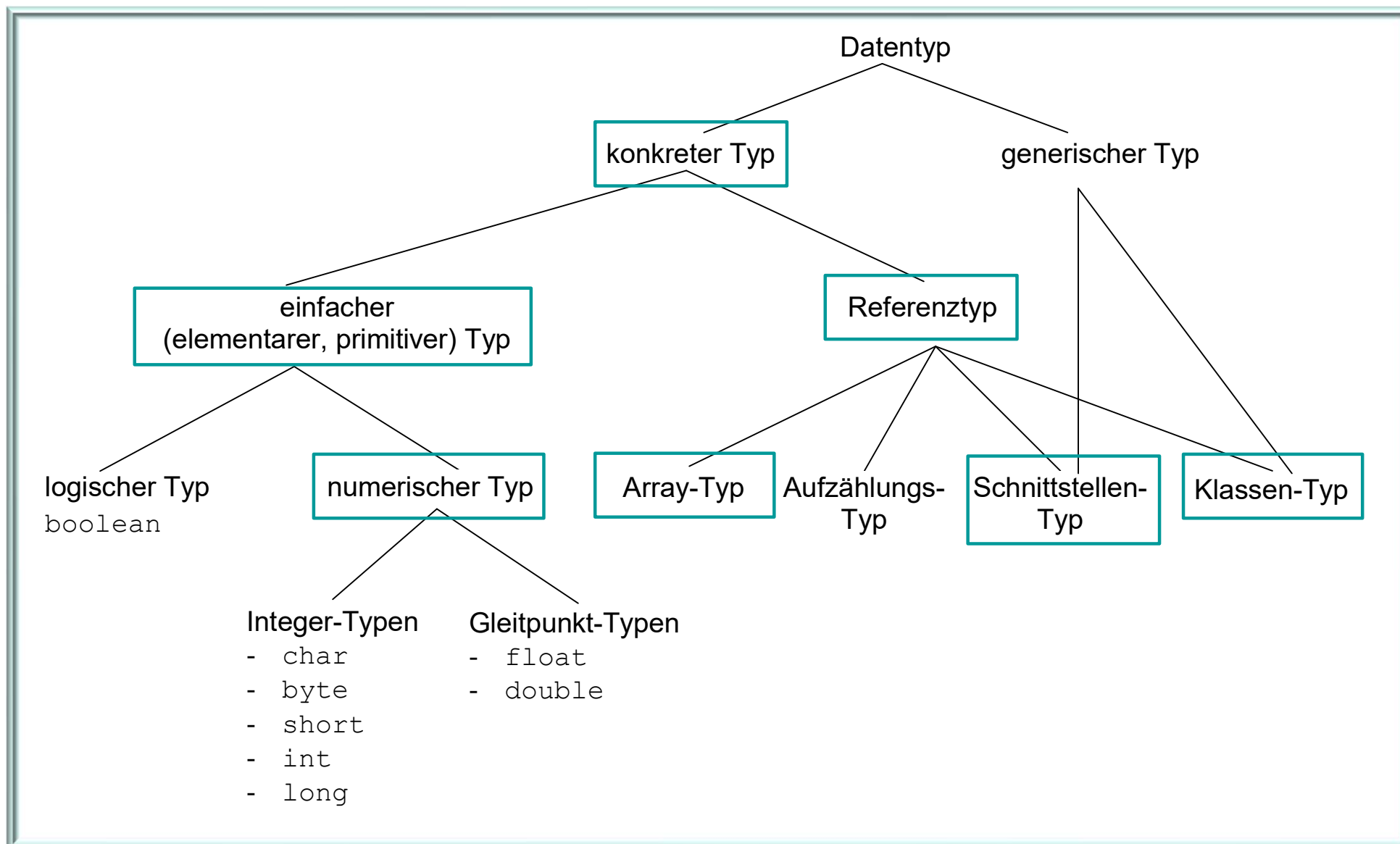
- Lege eine neue Klasse `TestAddierenEinlesen` mit einer `main()`-Methode an.
- Lege eine Variable mit dem Namen `wert1` vom Datentyp `int` an.
- Lege eine Variable mit dem Namen `wert2` vom Datentyp `int` an.
- Programmiere eine Ausgabe, die den Benutzer auffordert einen Wert einzugeben.
- Lies einen Wert von der Tastatur in die Variable `wert1` ein.
- Programmiere eine Ausgabe, die den Benutzer auffordert einen Wert einzugeben.
- Lies einen Wert von der Tastatur in die Variable `wert2` ein.
- Lege eine Variable mit dem Namen `summe` vom Datentyp `int` an.
- Berechne die Summe der Variablen `wert1` und `wert2` und speichere das Ergebnis in der Variablen `summe`.
- Gib die berechnete Summe auf der Konsole aus.
- Lege eine Variable mit dem Namen `produkt` vom Datentyp `int` an.
- Berechne das Produkt der Variablen `wert1` und `wert2` und speichere das Ergebnis in der Variablen `produkt`.
- Gib das berechnete Produkt auf der Konsole aus.



Hilfestellung Aufgabe 2:

- Für das Einlesen von der Konsole benötigst Du die Klasse `Scanner` aus dem Paket `java.util`.
- Du musst diese Klasse importieren mit
`import java.util.Scanner;`
- Du musst ein Objekt von der Klasse `Scanner` erzeugen:
`Scanner scan = new Scanner (System.in);`
- Du kannst die Methode `nextInt()` der Klasse `Scanner` verwenden, um eine Ganzzahl einzulesen:
`int i = scan.nextInt();`
- Wenn du das `Scanner`-Objekt nicht mehr benötigst, rufst Du die Methode `close()` auf:
`scan.close();`

Alle Datentypen (Typen) auf einen Blick



Agenda

- Datentypen und Variable
- Referenzvariable
- this-Referenz

Einführung Referenzvariablen

Wir wissen bislang Folgendes:

- eine Klasse ist ein Datentyp
- ein Objekt ist eine Variable
- ein Objekt kann in Java nur mit Hilfe des `new`-Operators erzeugt werden

Mit

```
new Person();
```

wird ein Objekt der Klasse `Person` angelegt.



Damit man mit einem Objekt in Java etwas machen kann, benötigt man einen „Zeiger“ auf dieses Objekt. Dieser Zeiger wird in Java **Referenz** bzw. **Referenzvariable** genannt.



Wo befinden sich im Programm `TestPerson.java` Referenzvariablen?

Anlegen und Verwenden von Referenzvariablen

Eine Referenzvariable wird beispielsweise angelegt durch:

```
Person p1;
```

Die Referenzvariable `p1` vom Datentyp `Person` – man sagt vereinfacht „vom Typ `Person`“ – kann nun auf ein beliebiges Objekt vom Typ `Person` zeigen.

Mit

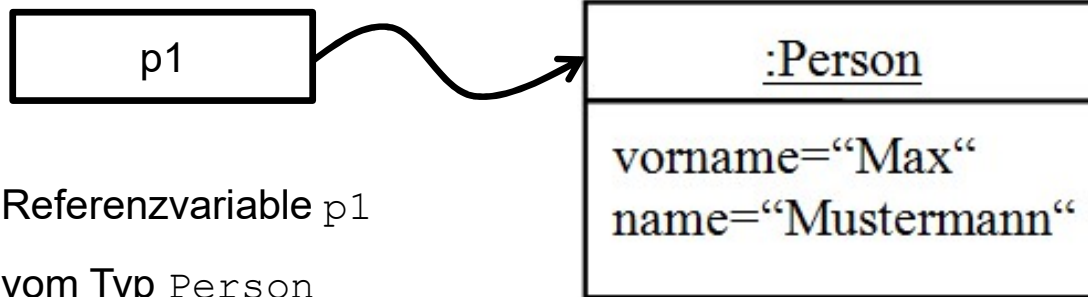
```
p1 = new Person();
```

wird der Referenzvariablen `p1` als Wert die Adresse zugewiesen, an welcher das Objekt vom Typ `Person` durch den `new`-Operator im Speicher angelegt wurde.



In Java wird eine Referenzvariable benötigt, um auf Objekte zuzugreifen, die mit Hilfe des `new`-Operators im Speicher angelegt wurden. In einer Referenzvariablen wird als Wert die Adresse abgelegt, an welcher sich das Objekt im Speicher befindet.

Grafische Darstellung einer Referenzvariablen



Referenzvariable `p1`

vom Typ `Person`

zeigt auf ein Objekt

vom Typ `Person`



Ein Objekt hat in Java keinen Namen. Auf ein Objekt kann nur zugegriffen werden, wenn die Adresse (wo das Objekt durch den `new`-Operator erzeugt wurde) einer Referenzvariablen zugewiesen wird.



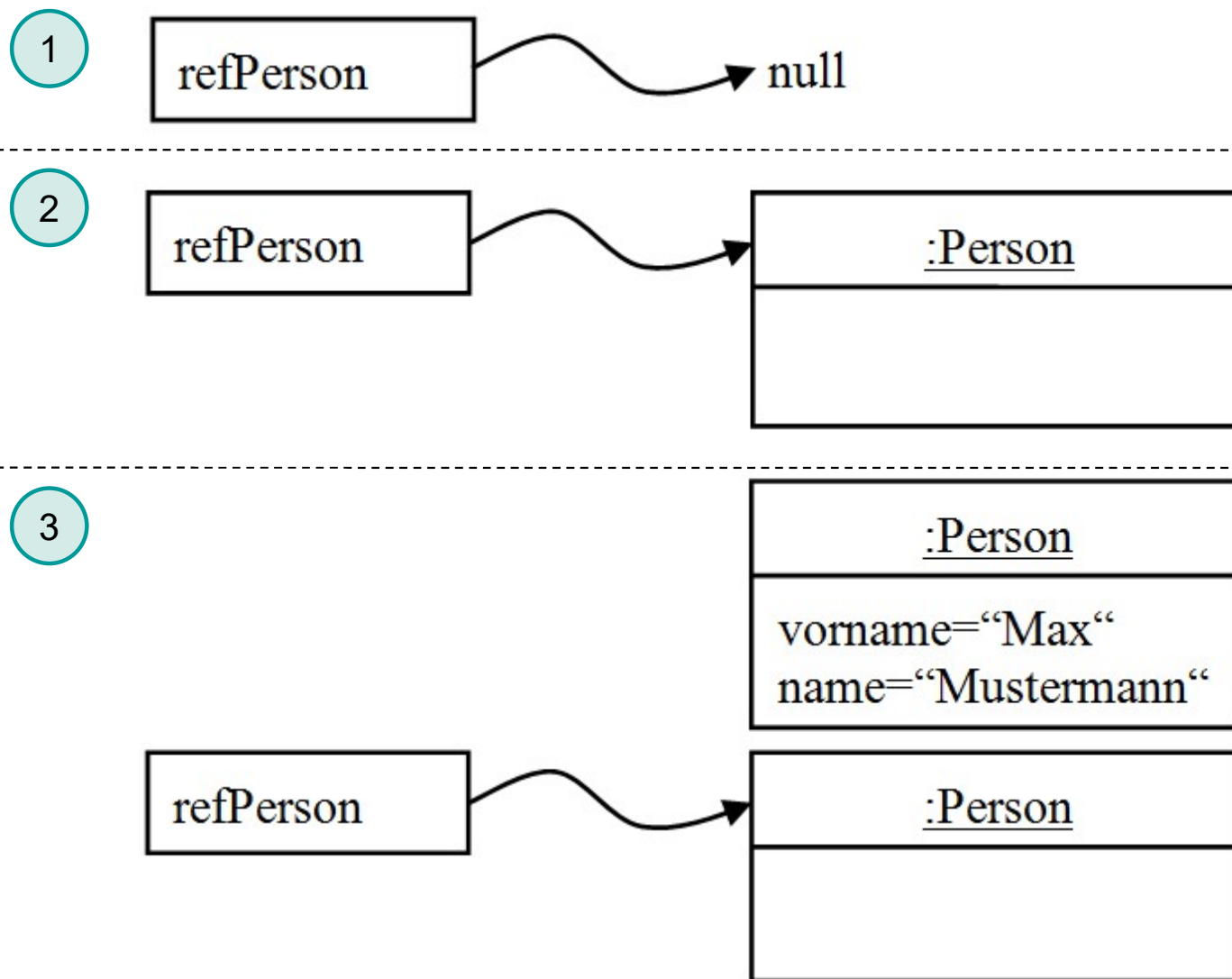
Aufgabe 3

- Klassen `Person` und `TestPerson` aus `teil_3` kopieren und in `teil_4` einfügen.
- Die Klasse `TestPerson` soll derart umgeschrieben werden, dass nur eine einzige Referenzvariable `refPerson` benötigt wird.
- Die Ausgabe des Programmes soll identisch bleiben.

```
// Datei: TestPerson.java
public class TestPerson
{
    public static void main (String[] args)
    {
        // Referenzvariable refPerson anlegen und mit null initialisieren.
        ① Person refPerson = . . . . .;
        ② refPerson = . . . . .; //refPerson soll auf erstes Objekt vom Typ Person zeigen.
        refPerson.setName ("Mustermann");
        refPerson.setVorname ("Max");
        System.out.println (refPerson.getName() + " " + refPerson.getVorname());

        // Referenzvariable refPerson soll auf das zweite Objekt vom Typ Person zeigen.
        ③ refPerson = . . . . .;
        refPerson.setName ("Meister");
        refPerson.setVorname ("Ralf");
        System.out.println (refPerson.getName() + " " + refPerson.getVorname());
    }
}
```

Visualisierungen zur Aufgabe 1





Aufgabe 4: Personendaten von der Tastatur einlesen

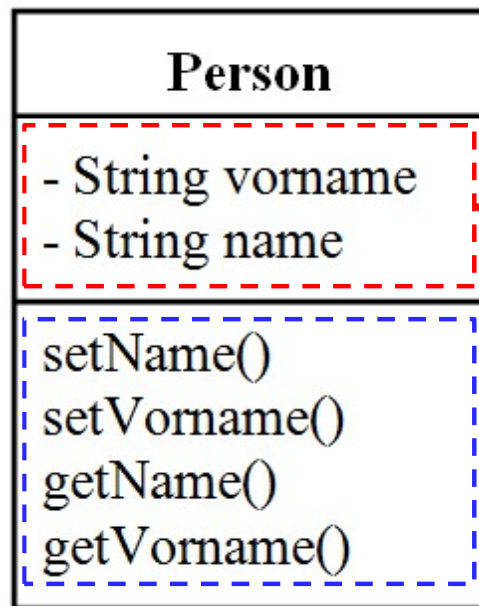
- Erstelle eine neue Klasse `TestPersonEinlesen` mit einer `main()`-Methode.
- Lege zwei Personen-Objekte an und lies den Vornamen und den Nachnamen mit Hilfe der Methode `nextLine()` der Klasse `Scanner` von der Konsole ein.
- Gib die eingelesenen Personendaten zu Testzwecken auf der Konsole aus.

Agenda

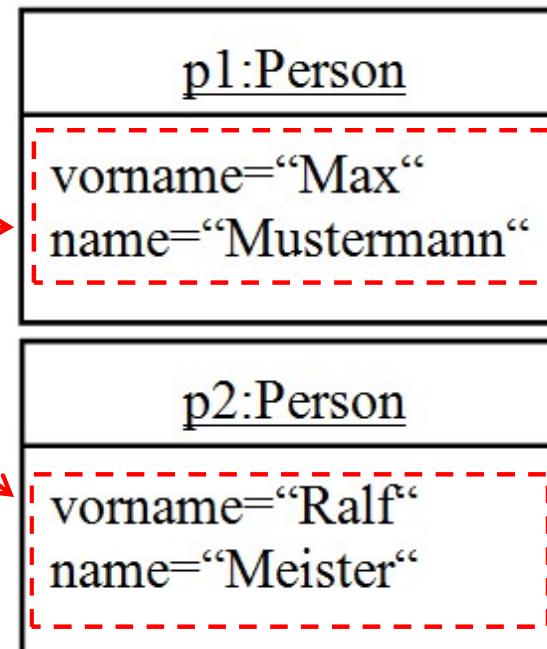
- Datentypen und Variable
- Referenzvariable
- this-Referenz

Zugriff auf die Instanzvariablen

Klasse = Bauplan:



beliebig viele Objekte:



Jedes erzeugte Objekt hat seine eigenen Instanzvariablen, denen Objekt-individuelle Werte zugewiesen werden!

Methodenimplementierung existiert nur einmal für alle Objekte gemeinsam!



Woher weiß eine Instanzmethode, auf welchen Objekt-individuellen Instanzvariablen sie arbeiten soll?



Aufgabe 5: this-Referenz in Klasse Person verwenden

- Schreibe die Klasse `Person` derart um, dass die Übergabeparameter der `set()`-Methoden den gleichen Namen haben, wie die zugehörige Instanzvariable.
- Verwende für den Zugriff auf die dann verdeckte Instanzvariable die `this`-Referenz.