ABSTRAKTE KLASSE

Übersicht

- Abstrakte Klasse
- Beispiel Personen

Abstrakte Klasse

Situation

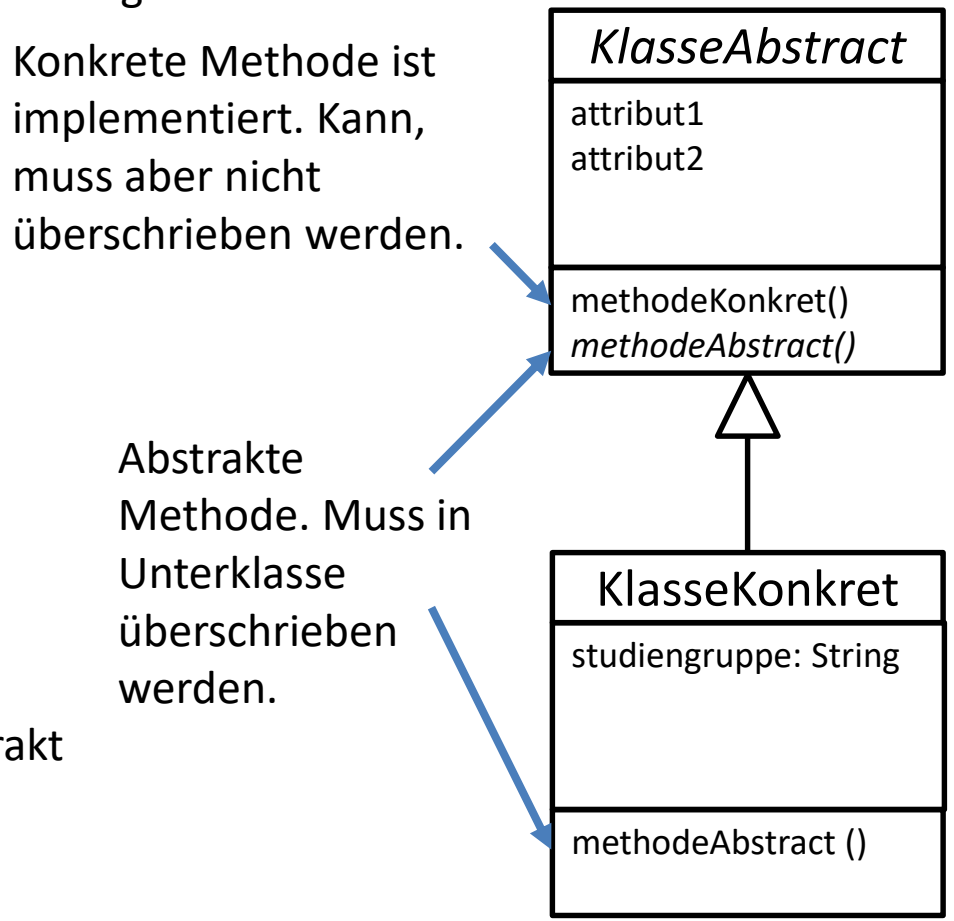
- nicht immer ist Instanz sinnvoll
- Oberklasse könnte unvollständig sein, nur durch Unterklasse vollständig

Konzept Abstrakte Klasse

- Abstraktion verbietet Instanziierung (Compilerfehler)
- muss für Nutzung Unterklasse haben
- Schlüsselwort **`abstract`**
  - für eine Klasse
  - zusätzlich möglich für Methode

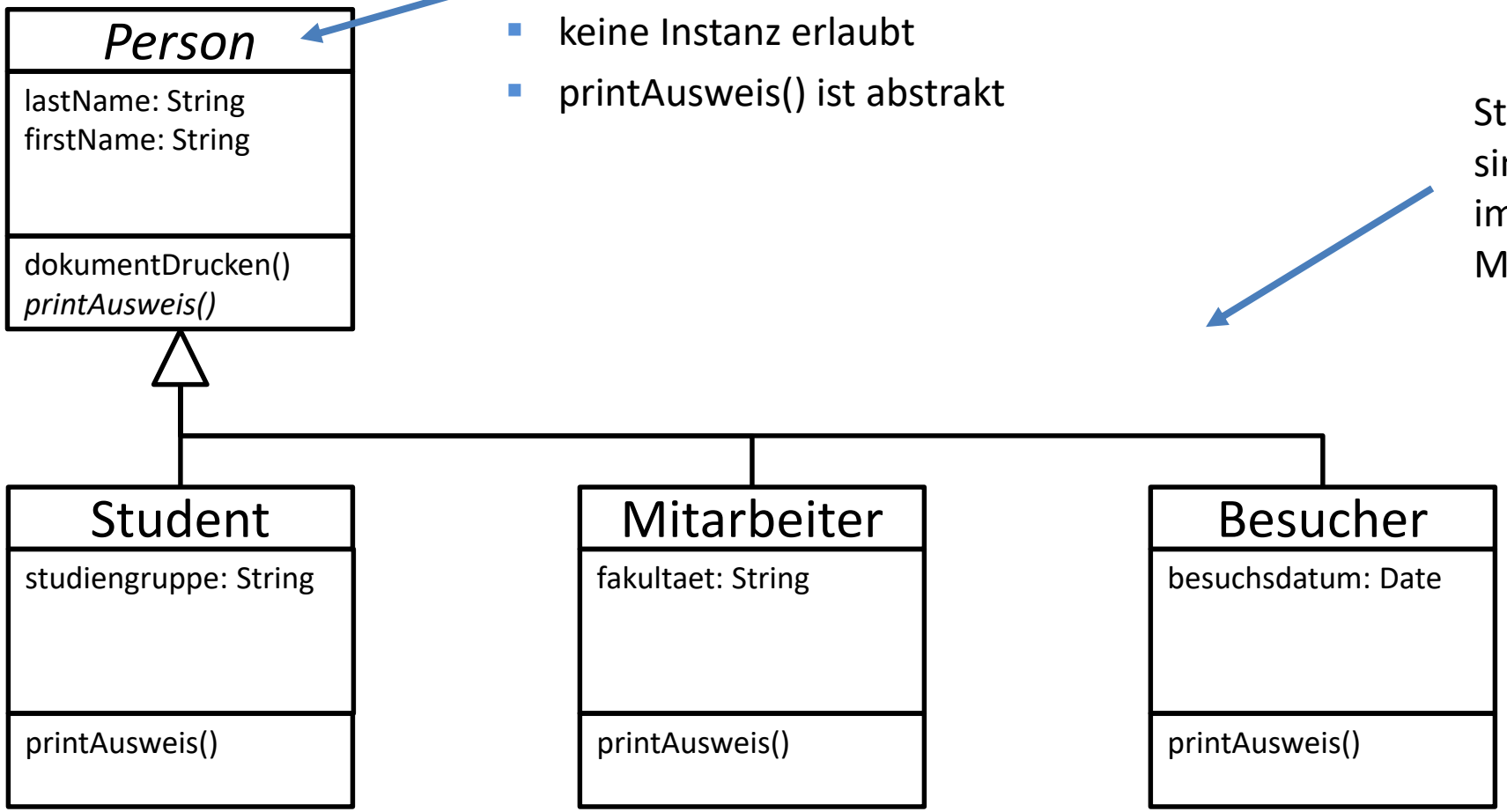
Abstrakte Methode

- muss in Unterklasse implementiert sein
- Klassen mit mind. einer abstrakten Methode sind ebenfalls abstrakt



Beispiel Abstrakte Klasse /1

demo005.personAbstract



Beispiel Abstrakte Klasse /2

demo005.personAbstract

Schlüsselwort zur Kennzeichnung der abstrakten Klasse Person

```
public abstract class Person {
     protected String firstName;
     protected String lastName;

     protected Person(String firstName, String lastName) {
          this.firstName=firstName;
          this.lastName=lastName;
     }

     public void dokumentDrucken() {
          System.out.println("====================");
          printAusweis();
          System.out.println("====================");
     }

     protected abstract void printAusweis();
}
```

Abstrakte Methode wird aufgerufen und führt zur Laufzeit zur konkreten Implementierung der Unterklasse

Methode ist abstrakt und muss in Unterklasse implementiert werden

Beispiel Abstrakte Klasse /3

demo005.personAbstract

```
public class Student extends Person {

     private String studiengruppe;

     public Student(String firstName, String lastName, String studiengruppe) {
          super(firstName, lastName);
          this.studiengruppe=studiengruppe;
     }

     @Override
     protected void printAusweis() {
          System.out.println("Studierendenausweis:");
          System.out.println("Name: "+lastName);
          System.out.println("Last Name: "+lastName);
          System.out.println("Studiengruppe: "+studiengruppe);
     }

}
```

Annotation @override kennzeichnet das Überschreiben der abstrakten Methode

Konkrete Implementierung der abstrakten Methode

Motivation für Abstrakte Klasse

Warum könnte man Abstrakte Klasse verwenden wollen?

Gemeinsame Funktionalität

- in einer Oberklasse bündeln
- durch extends in Unterklassen einbinden
- Verwendung erfordert eine Unterklasse

API (Application Programming Interface)

- teilweise definiert in Oberklasse
- implementiert in Unterklasse
- Ggf. Erweiterung in Unterklasse

Vorgabe von Verarbeitungsschritten

- Basis der Implementierung in abstrakter Klasse
- differenzierte Funktion in Unterklasse
- zusammenführen in Oberklasse
- erzwingen von Implementierung in Unterklasse

```
public abstract class ProzessAbstract {
    public void process() {
        einVerarbeitungsschritt();
        aktion();
        weitererSchritt();
    }

    public void einVerarbeitungsschritt() {
        // Implementierung hier in Oberklasse
        // hier ist Realisierung schon bekannt
    }

    // wird in Unterklasse implementiert, weil
    // hier die Realisierung nicht bekannt ist
    public abstract void aktion();

    public void weitererSchritt() {
        // Implementierung hier in Oberklasse
        // hier ist Realisierung schon bekannt
    }
}
```

Zusammenfassung

- Abstrakte Klasse
  - keine Instanz
  - abstrakte Methoden möglich
- Beispiel Personen