

# **Propädeutikum:**

# **Programmierung in der Bioinformatik**

## Klassen und Objekte

Thomas Mauermeier

19.11.2019

Ludwig-Maximilians-Universität München

# Bisher: Prozedurale Programmierung

```
public class StudentenNaiv {  
    public static void main(String[] args) {  
        String s1_name = "Max";  
        int s1_semester = 1;  
        int s1_matrikel = 12345;  
        String s2_name = "Julia";  
        int s2_semester = 8;  
        int s2_matrikel = 87654;  
        String s3_name = "Alex";  
        int s3_semester = 4;  
        int s3_matrikel = 65432;  
        s1_semester++;  
        s3_name = "Alexandra";  
        boolean studiertLaenger = s1_semester > s3_semester;  
    }  
}
```

} Student 1

} Student 2

} Student 3

} Aktionen "auf" Studis

**Problem:** Viel Wiederholung, keine standardisierten Verfahren um mit Studierenden umzugehen

# Idee: Wir brauchen einen "Bauplan"

```
public class StudentenNaiv {  
    public static void main(String[] args) {  
        String s1_name = "Max";  
        int s1_semester = 1;  
        int s1_matrikel = 12345;  
        String s2_name = "Julia";  
        int s2_semester = 8;  
        int s2_matrikel = 87654;  
        String s3_name = "Alex";  
        int s3_semester = 4;  
        int s3_matrikel = 65432;  
        s1_semester++;  
        s3_name = "Alexandra";  
        boolean studiertLaenger = s1_semester > s3_semester;  
    }  
}
```

Was *hat* jeder Student?

- Name
- Semester
- Matrikelnummer

"Bauplan"

Was *kann* jeder Student?

- Semester erhöhen
- Namen ändern
- Studienzeitsvergleich

# Idee: Wir brauchen einen “Bauplan”

Was *hat* jeder Student?

## Attribute:

- Name
- Semester
- Matrikelnummer

Was *kann* jeder Student?

## Methoden:

- Semester erhöhen
- Namen ändern
- Studienzeitsvergleich

definiert

definiert

“Bauplan” = **Klasse** Student

# Bauplan in Java: Klasse Student

```
public class Student {
    String name;
    int semester;
    int matrikel;

    public Student(String name, int semester, int matrikel) {
        this.name = name;
        this.semester = semester;
        this.matrikel = matrikel;
    }
    public void naechstesSemester() {
        this.semester++;
    }
    public void setName(String name) {
        this.name = name;
    }
    public boolean studiertLaengerAls(Student s) {
        return this.semester > s.semester;
    }
    // main-Methode rauskommentiert
}
```

# Bauplan in Java: Klasse Student

```
public class Student {  
    String name;  
    int semester;  
    int matrikel;  
  
    public Student(String name, int semester, int matrikel) {  
        this.name = name;  
        this.semester = semester;  
        this.matrikel = matrikel;  
    }  
    public void naechstesSemester() {  
        this.semester++;  
    }  
    public void setName(String name) {  
        this.name = name;  
    }  
    public boolean studiertLaengerAls(Student s) {  
        return this.semester > s.semester;  
    }  
    // main-Methode rauskommentiert  
}
```

## Attribute:

- Name
- Semester
- Matrikelnummer

# Attribute: Was hat das Objekt

```
public class Student {  
    String name;  
    int semester;  
    int matrikel;  
}
```

Beispiel:

Student
name = "Julia" semester = 8 matrikel = 87654

Student
name = "Max" semester = 1 matrikel = 12345

- Variablen aus denen das Objekt “bestehen” soll (deswegen auch manchmal *Objektvariablen* genannt)
- Man deklariert **einmal** die Variablen, die dann **jedes** Objekt haben soll
- Jedes erzeugte Objekt besitzt dann eine **eigene Instanz** der deklarierten Attribute
  - d.h. diese werden **nicht** mit anderen Objekten des selben Typs geteilt

# Bauplan in Java: Klasse Student

```
public class Student {
    String name;
    int semester;
    int matrikel;

    public Student(String name, int semester, int matrikel) {
        this.name = name;
        this.semester = semester;
        this.matrikel = matrikel;
    }
    public void naechstesSemester() {
        this.semester++;
    }
    public void setName(String name) {
        this.name = name;
    }
    public boolean studiertLaengerAls(Student s) {
        return this.semester > s.semester;
    }
    // main-Methode rauskommentiert
}
```

## Methoden:

- Semester erhöhen
- Namen ändern
- Studienzeitsvergleich



# Methoden: Was kann das Objekt

```
public void naechstesSemester() {  
    this.semester++;  
}  
public void setName(String name) {  
    this.name = name;  
}  
public boolean studiertLaengerAls(Student s) {  
    return this.semester > s.semester;  
}
```

Beispiel:

Student
name = "Max" semester = 1 matrikel = 12345
naechstesSemester setName studiertLaengerAls

- Deklaration der Methoden die mit dem Objekt verknüpft sind
- Man deklariert **einmal** die Methoden, die dann **jedes** Objekt haben soll
- Bei Aufruf der Attribute mit **this**, wird mit den Variablen des Objekts gearbeitet, **das die Methode aufruft** (→ Punktoperator)

# Punktoperator

Angenommen unser Student-Objekt ist in Variable s1 abgelegt:

**Wie greife ich jetzt auf die Attribute/Methoden des Objekts zu?**

**Punktoperator:**

objekt.**.**methode();  
objekt.**.**variable;

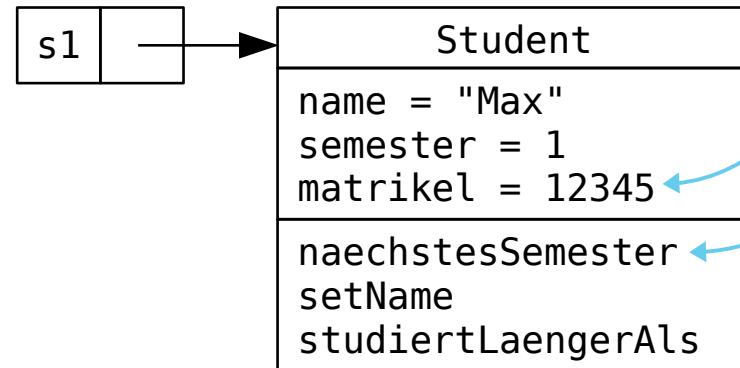


Lesbar als:

“Von **objekt**, Zugriff auf dessen **methode/variable**”

Beispiel:

```
s1.naechstesSemester();  
s1.matrikel;
```



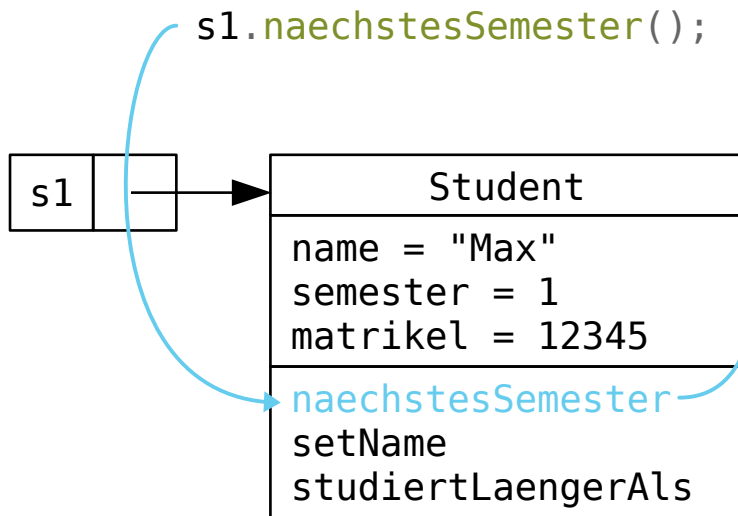
# this-Referenz

## Selbstreferenz **this**:

Bezieht sich in Objektmethoden und Konstruktoren auf das Objekt *selbst*.

Wird erst ein konkreter Wert, wenn auf ein bestimmtes Objekt zugegriffen wird.

Beispiel:



```
public void naechstesSemester() {
    this.semester++;
}
```

da Aufruf "von s1 aus"

```
public void naechstesSemester() {
    s1.semester++;
}
```

# Bauplan in Java: Klasse Student

```
public class Student {
    String name;
    int semester;
    int matrikel;

    public Student(String name, int semester, int matrikel) {
        this.name = name;
        this.semester = semester;
        this.matrikel = matrikel;
    }
    public void naechstesSemester() {
        this.semester++;
    }
    public void setName(String name) {
        this.name = name;
    }
    public boolean studiertLaengerAls(Student s) {
        return this.semester > s.semester;
    }
    // main-Methode rauskommentiert
}
```

## **Konstruktor:**

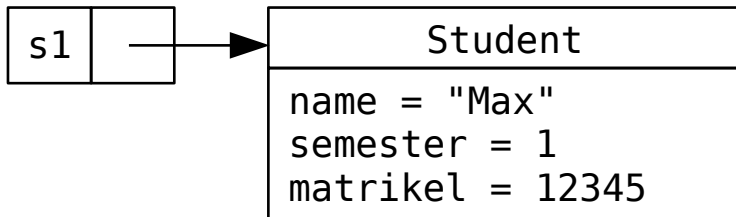
Besondere Methode zum Erzeugen von Objekten

# Konstruktor: Erzeugen eines Objekts

```
public Student(String name, int semester, int matrikel) {  
    this.name = name;  
    this.semester = semester;  
    this.matrikel = matrikel;  
}
```

Beispiel:

```
Student s1 = new Student("Max", 1, 12345);
```



- Besondere Methode: Definiert wie Attribute “befüllt” werden, also wie Objekt *konstruiert* wird
- **Aufruf** eines Konstruktors mit Keyword “new”, erzeugt **Instanz** einer Klasse, also ein Objekt
- **Unterschiede** zu normalen Methoden:
  - Name der Methode **muss** Klassenname sein
  - Es wird **kein** Rückgabewert angegeben (gibt aber offensichtlich das Objekt zurück)

# Idee: Wir brauchen einen "Bauplan"

Was *hat* jeder Student?

## Attribute:

- Name
- Semester
- Matrikelnummer

Was *kann* jeder Student?

## Methoden:

- Semester erhöhen
- Namen ändern
- Studienzeitsvergleich

definiert

definiert

"Bauplan" = **Klasse** Student

erzeugt mit

**Konstruktor**

**Objekt** vom **Typ** Student

```
public class Student {
    String name;
    int semester;
    int matrikel;
    public Student(String name, int semester, int matrikel) {
        this.name = name;
        this.semester = semester;
        this.matrikel = matrikel;
    }
    public void naechstesSemester() {
        this.semester++;
    }
    public void setName(String name) {
        this.name = name;
    }
    public boolean studiertLaengerAls(Student s) {
        return this.semester > s.semester;
    }
    public static void main(String[] args) {
        Student s1 = new Student("Max", 1, 12345);
        Student s2 = new Student("Julia", 8, 87654);
        Student s3 = new Student("Alex", 4, 65432);
        s1.naechstesSemester();
        s3.setName("Alexandra");
        s1.studiertLaengerAls(s3);
    }
}
```

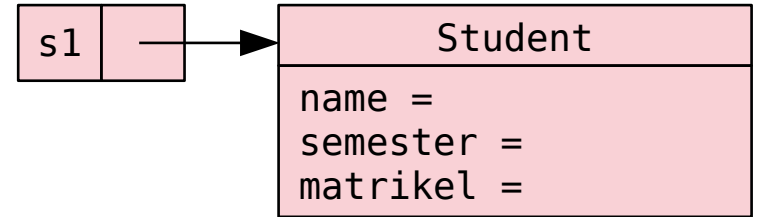
```
public class Student {
    String name;
    int semester;
    int matrikel;
    public Student(String name, int semester, int matrikel) {
        this.name = name;
        this.semester = semester;
        this.matrikel = matrikel;
    }
    public void naechstesSemester() {
        this.semester++;
    }
    public void setName(String name) {
        this.name = name;
    }
    public boolean studiertLaengerAls(Student s) {
        return this.semester > s.semester;
    }
    public static void main(String[] args) {
        Student s1 = new Student("Max", 1, 12345);
        Student s2 = new Student("Julia", 8, 87654);
        Student s3 = new Student("Alex", 4, 65432);
        s1.naechstesSemester();
        s3.setName("Alexandra");
        s1.studiertLaengerAls(s3);
    }
}
```



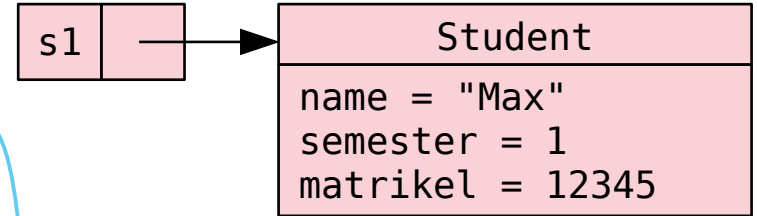
Einstieg



```
public class Student {
    String name;
    int semester;
    int matrikel;
    public Student(String name, int semester, int matrikel) {
        this.name = name;
        this.semester = semester;
        this.matrikel = matrikel;
    }
    public void naechstesSemester() {
        this.semester++;
    }
    public void setName(String name) {
        this.name = name;
    }
    public boolean studiertLaengerAls(Student s) {
        return this.semester > s.semester;
    }
    public static void main(String[] args) {
        Student s1 = new Student("Max", 1, 12345);
        Student s2 = new Student("Julia", 8, 87654);
        Student s3 = new Student("Alex", 4, 65432);
        s1.naechstesSemester();
        s3.setName("Alexandra");
        s1.studiertLaengerAls(s3);
    }
}
```



```
public class Student {
    String name;
    int semester;
    int matrikel;
    public Student(String name, int semester, int matrikel) {
        this.name = name;
        this.semester = semester;
        this.matrikel = matrikel;
    }
    public void naechstesSemester() {
        this.semester++;
    }
    public void setName(String name) {
        this.name = name;
    }
    public boolean studiertLaengerAls(Student s) {
        return this.semester > s.semester;
    }
    public static void main(String[] args) {
        Student s1 = new Student("Max", 1, 12345);
        Student s2 = new Student("Julia", 8, 87654);
        Student s3 = new Student("Alex", 4, 65432);
        s1.naechstesSemester();
        s3.setName("Alexandra");
        s1.studiertLaengerAls(s3);
    }
}
```

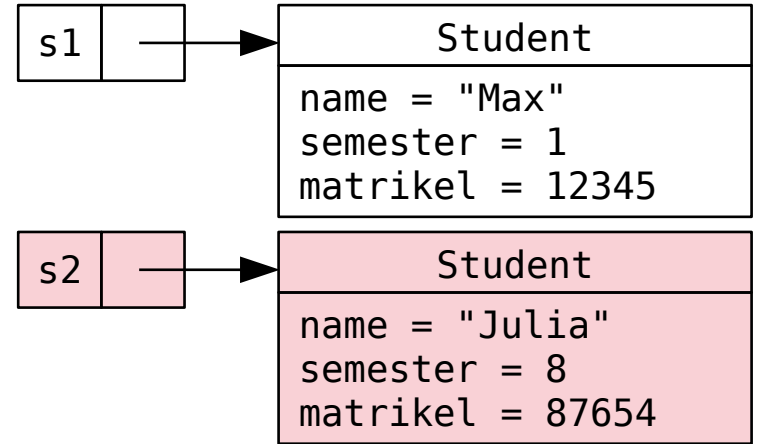


**this** bezieht sich hier auf **s1**

```

public class Student {
    String name;
    int semester;
    int matrikel;
    public Student(String name, int semester, int matrikel) {
        this.name = name;
        this.semester = semester;
        this.matrikel = matrikel;
    }
    public void naechstesSemester() {
        this.semester++;
    }
    public void setName(String name) {
        this.name = name;
    }
    public boolean studiertLaengerAls(Student s) {
        return this.semester > s.semester;
    }
    public static void main(String[] args) {
        Student s1 = new Student("Max", 1, 12345);
        Student s2 = new Student("Julia", 8, 87654);
        Student s3 = new Student("Alex", 4, 65432);
        s1.naechstesSemester();
        s3.setName("Alexandra");
        s1.studiertLaengerAls(s3);
    }
}

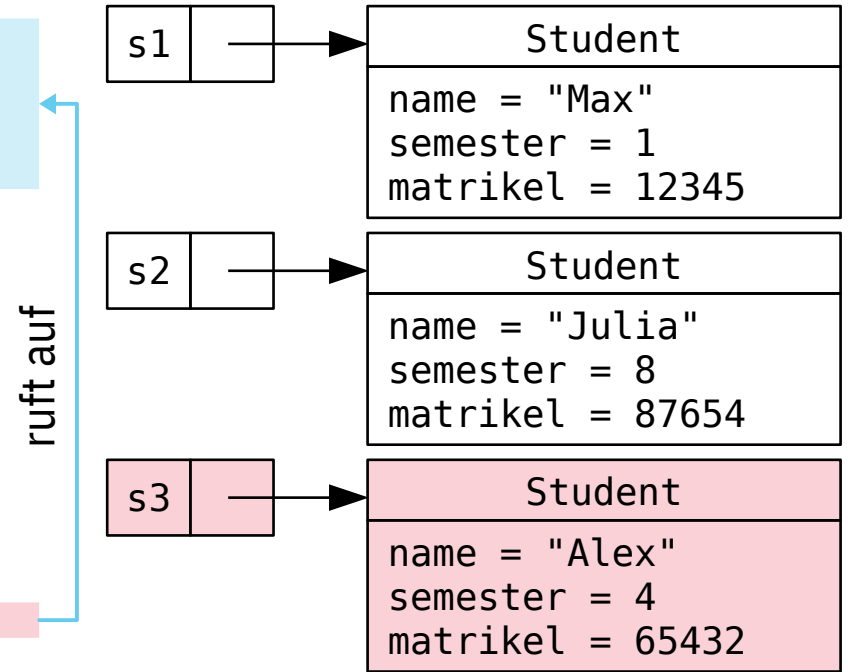
```



```

public class Student {
    String name;
    int semester;
    int matrikel;
    public Student(String name, int semester, int matrikel) {
        this.name = name;
        this.semester = semester;
        this.matrikel = matrikel;
    }
    public void naechstesSemester() {
        this.semester++;
    }
    public void setName(String name) {
        this.name = name;
    }
    public boolean studiertLaengerAls(Student s) {
        return this.semester > s.semester;
    }
    public static void main(String[] args) {
        Student s1 = new Student("Max", 1, 12345);
        Student s2 = new Student("Julia", 8, 87654);
        Student s3 = new Student("Alex", 4, 65432);
        s1.naechstesSemester();
        s3.setName("Alexandra");
        s1.studiertLaengerAls(s3);
    }
}

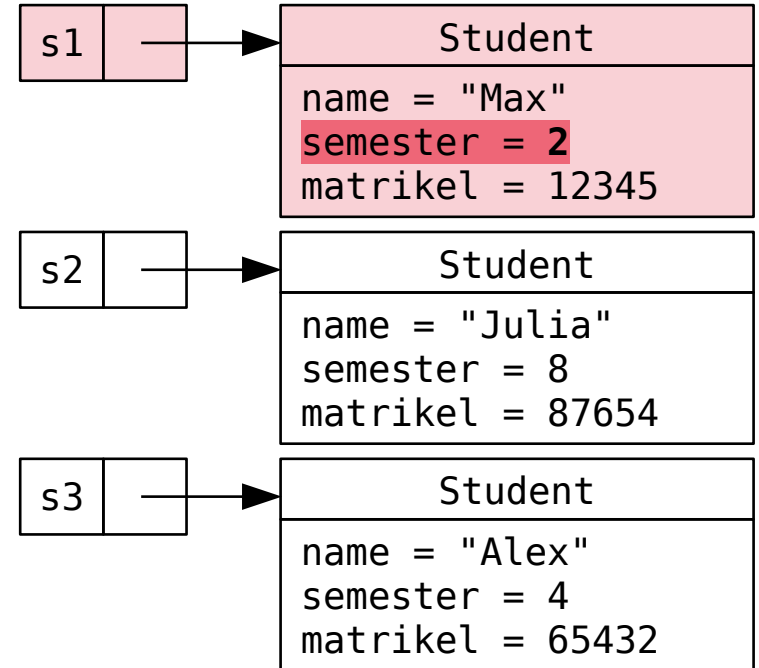
```



```

public class Student {
    String name;
    int semester;
    int matrikel;
    public Student(String name, int semester, int matrikel) {
        this.name = name;
        this.semester = semester;
        this.matrikel = matrikel;
    }
    public void naechstesSemester() {
        this.semester++;
    }
    public void setName(String name) {
        this.name = name;
    }
    public boolean studiertLaengerAls(Student s) {
        return this.semester > s.semester;
    }
    public static void main(String[] args) {
        Student s1 = new Student("Max", 1, 12345);
        Student s2 = new Student("Julia", 8, 87654);
        Student s3 = new Student("Alex", 4, 65432);
        s1.naechstesSemester();
        s3.setName("Alexandra");
        s1.studiertLaengerAls(s3);
    }
}

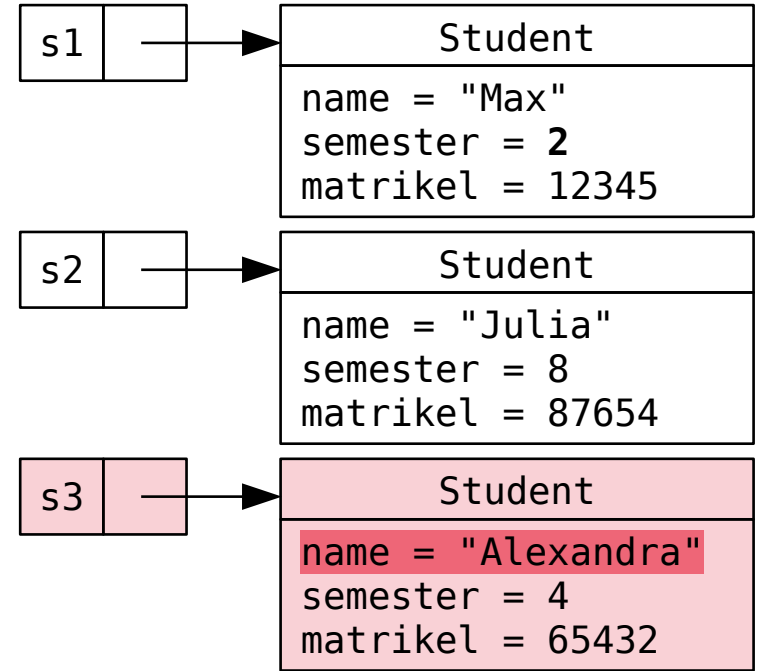
```



```

public class Student {
    String name;
    int semester;
    int matrikel;
    public Student(String name, int semester, int matrikel) {
        this.name = name;
        this.semester = semester;
        this.matrikel = matrikel;
    }
    public void naechstesSemester() {
        this.semester++;
    }
    public void setName(String name) {
        this.name = name;
    }
    public boolean studiertLaengerAls(Student s) {
        return this.semester > s.semester;
    }
    public static void main(String[] args) {
        Student s1 = new Student("Max", 1, 12345);
        Student s2 = new Student("Julia", 8, 87654);
        Student s3 = new Student("Alex", 4, 65432);
        s1.naechstesSemester();
        s3.setName("Alexandra");
        s1.studiertLaengerAls(s3);
    }
}

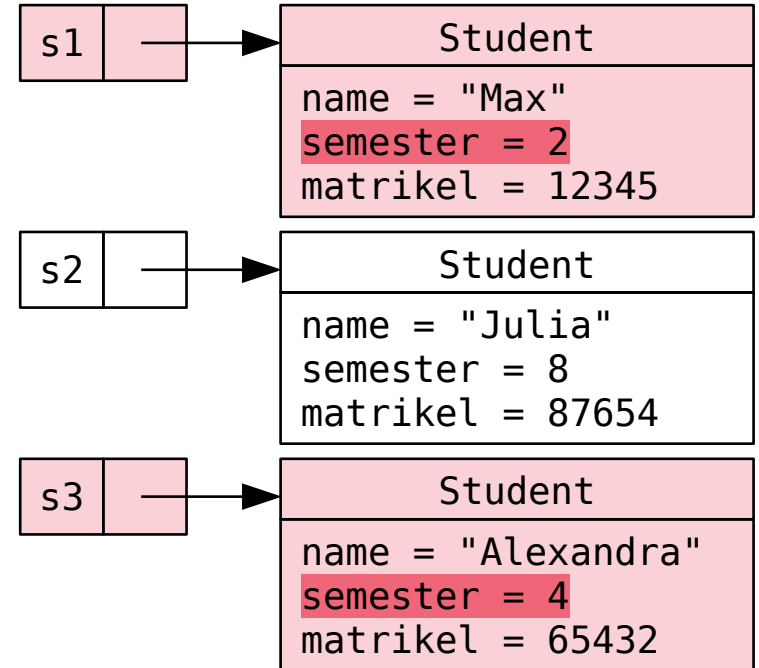
```



```

public class Student {
    String name;
    int semester;
    int matrikel;
    public Student(String name, int semester, int matrikel) {
        this.name = name;
        this.semester = semester;
        this.matrikel = matrikel;
    }
    public void naechstesSemester() {
        this.semester++;
    }
    public void setName(String name) {
        this.name = name;
    }
    public boolean studiertLaengerAls(Student s) {
        return this.semester > s.semester;
    }
    public static void main(String[] args) {
        Student s1 = new Student("Max", 1, 12345);
        Student s2 = new Student("Julia", 8, 87654);
        Student s3 = new Student("Alex", 4, 65432);
        s1.naechstesSemester();
        s3.setName("Alexandra");
        s1.studiertLaengerAls(s3);
    }
}

```



ruft auf

**this.semester** → s1.semester  
**s.semester** → s3.semester  
 ⇒ 2 > 4? **false**

# Objektorientierte Programmierung (OOP)

## Attribute:

- Was *hat* das Objekt?
- Variablen die das Objekt definieren

## Methoden:

- Was *kann* das Objekt?
- Methoden zur Arbeit “auf den Objekten”

definiert

definiert

**Klasse**

erzeugt mit  
**Konstruktor**

**Objekt** vom **Typ** den die Klasse definiert



# **Backup-Folien**

# Klasse Student

```
public class Student {
    String name;
    int semester;
    int matrikel;

    public Student(String name, int semester, int matrikel) {
        this.name = name;
        this.semester = semester;
        this.matrikel = matrikel;
    }
    public void naechstesSemester() {
        this.semester++;
    }
    public void setName(String name) {
        this.name = name;
    }
    public boolean studiertLaengerAls(Student s) {
        return this.semester > s.semester;
    }
    // main-Methode rauskommentiert
}
```

# this-Referenz bei Konstruktoren

```
public Student(String name, int semester, int matrikel) {  
    this.name = name;  
    this.semester = semester;  
    this.matrikel = matrikel;  
}
```

Beispiel:

```
Student s1 = new Student("Max", 1, 12345);  
|-----|-----|  
Aufrufendes      Konstruktoraufruf  
Objekt
```

```
this.name = name;  
this.semester = semester;  
this.matrikel = matrikel;
```

Aufruf von Objekt s1 aus

```
s1.name = name;  
s1.semester = semester;  
s1.matrikel = matrikel;
```