



Java Einführung Objektorientierung

Kapitel 11

Inhalt

- Methoden in Klassen
- Aufruf der Methoden
- Konstruktoren
- Statische Komponenten

Klassen

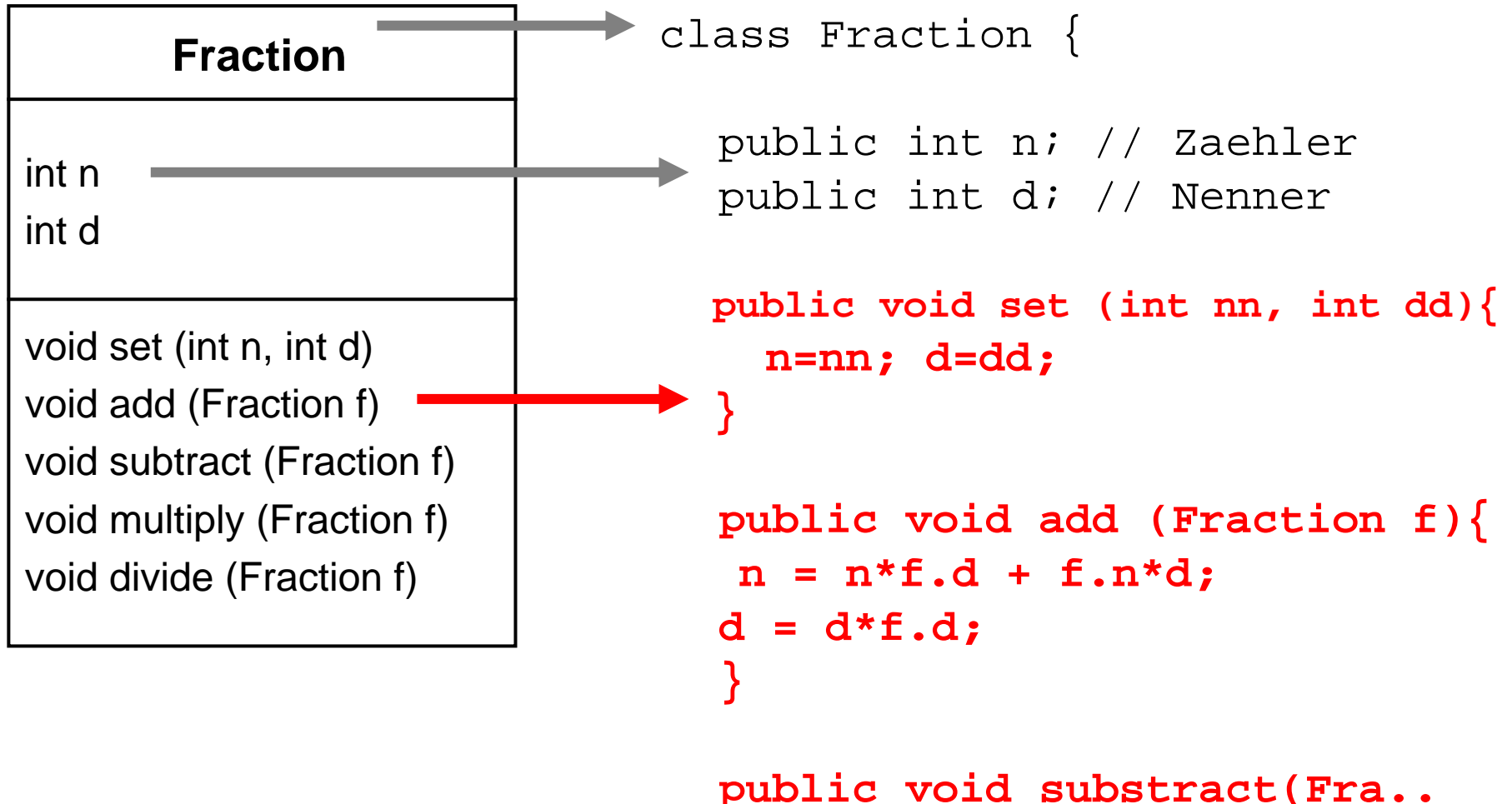
- Klassen fügen
 - **Attribute** und
 - **Methoden**

zu einer logischen Einheit zusammen.

- Klassen sind die Grundbausteine der objektorientierten Programmierung.

Name
Attribute
Methoden

Methoden in Klassen



Aufruf von Methoden in Klassen I

```
Fraction a = new Fraction();  
a.set(1,2);
```

Instanz a

```
...  
void set (int nn, int dd) {  
    n = nn;  
    d = dd;  
}...
```

Instanzvariablen können in den Methoden auch mit `this` angesprochen werden. Z.B. kann die Methode `set` so aussehen:

```
void set (int n, int d) {  
    this.n = n; this.d = d;  
}
```

Aufruf von Methoden in Klassen II

```
Fraction a = new Fraction();
```

```
a.n=1; a.d=2;
```

```
Fraction b = new Fraction();
```

```
b.n=3; b.d=5;
```

```
@.add(b);
```

Instanz a

```
...  
void add (Fraction f) {  
    n = n*f.d + f.n*d;  
    d = d*f.d;  
}...
```

Die Instanz *a* enthält nun das Ergebnis der Addition.

Konstrukturen

- Wenn ein Instanz erzeugt wird, werden alle Klassenvariablen automatisch erzeugt und erhalten folgende Werte:
 - Zahlen: `0` oder `0.0`
 - Referenzen (Strings, Arraynamen, Instanznamen): `null`
 - Boolean: `false`
- Bei der Erzeugung wird eine spezielle Methode aufgerufen (Konstruktor, gleicher Name wie die Klasse, kann Argumente haben). In dieser Methode können die Werte der Instanzvariablen initialisiert werden.

Definition von Konstruktoren

```
class Fraction{
    int n,d;
    Fraction () { // Default-Konstruktor
        n=0; d=1;
    }
    Fraction (int n, int d) {
        this.n = n; // this bezeichnet die Klassenvariablen
        this.d = d;
    }
    Fraction (Fraction toCopy) { // Copy-Konstruktor*
        this.n = toCopy.n;
        this.d = toCopy.d;
    }
}
```

...

* Java unterstützt die Methode `.clone()` der Klasse `Object`

```
}
```

Verwendung von Konstruktoren

- Aufruf: Default Konstruktor

```
Fraction c = new Fraction(); // 0/1
```

- Aufruf: Konstruktor mit Argumenten

```
Fraction d = new Fraction(3,7); // 3/7
```

Statische Komponenten

- Variablen und Methoden die mit dem Schlüsselwort **static** deklariert werden:
 - Existieren nur 1x (pro Klasse pro Programm)
 - Existieren auch ohne dass eine Instanz der Klasse erzeugt werden muss
 - Können auch über den Klassennamen angesprochen werden `Class.field` bzw. `Class.method()`

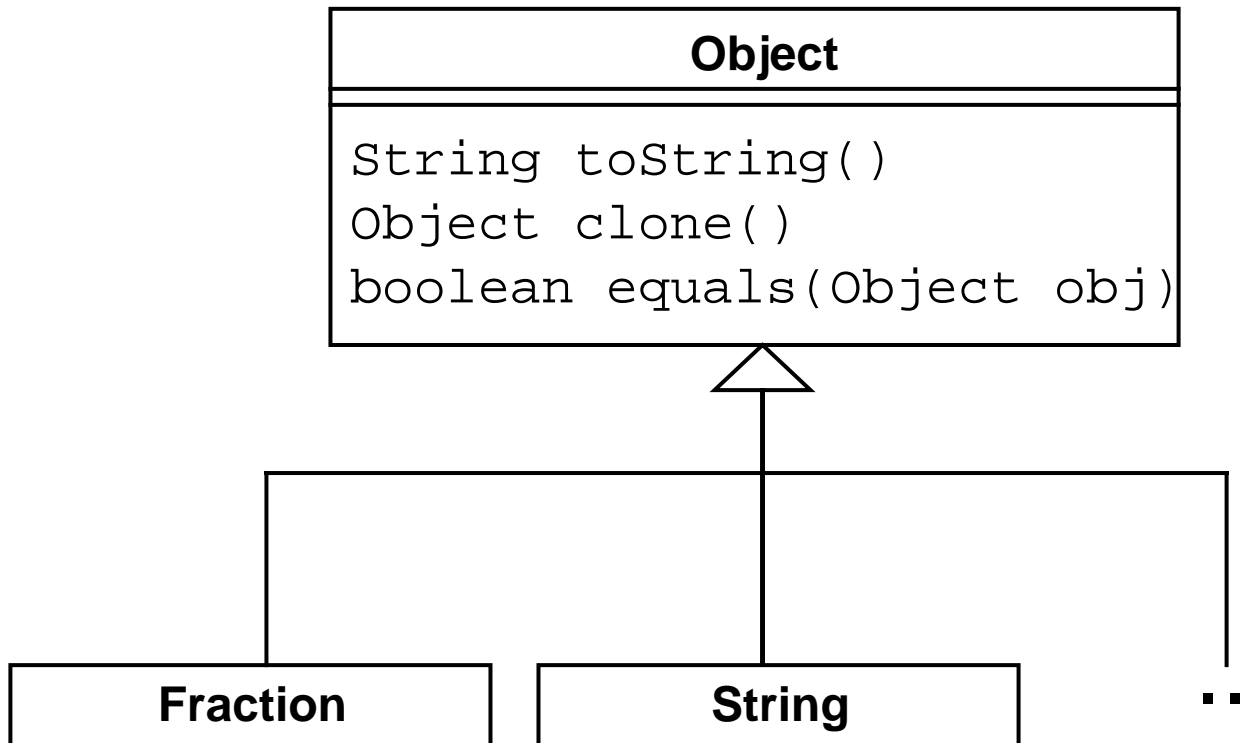
Bsp: Statische Komponenten II

```
class MyProg {  
    static void print (String s){  
        Out.println(s);  
    }  
  
    public static void main (String[] arg) {  
        for (int i=0; i<arg.length; i++) {  
            print(arg[i]);  
        }  
    }  
}
```

Die Methode `main` wird beim Programmstart mit dem Array aus Strings aufgerufen. Z.B. `java MyProg Hallo, wie geht es?`

Die Klasse Object in Java

- In Java wird jede Klasse von Object abgeleitet und erbt daher einige Methoden.



siehe `java.lang.Object` in der API Spezifikation

toString()

Automatische Konversion zu String

```
class Fraction {  
    int n, d;  
    ...  
    public String  
    toString() {  
        return("fraction(" +  
        d + "/" + n + ")")  
    }  
}
```

```
class TestFraction {  
    public static void  
    main(String[] args) {  
        f1 = new  
        Fraction(1,2);  
        System.out.println(f1);  
    }  
}
```