II. Imperative und objektorientierte Programmierung

- 1. Grundelemente der Programmierung
- 2. Objekte, Klassen und Methoden
- 3. Rekursion und dynamische Datenstrukturen
- 4. Erweiterung von Klassen und fortgeschrittene Konzepte

II.2. Objekte, Klassen und Methoden

- 1. Grundzüge der Objektorientierung
- 2. Methoden, Unterprogramme und Parameter
- 3. Datenabstraktion
- 4. Konstruktoren
- 5. Vordefinierte Klassen

Rechteck-Programm mit Arrays

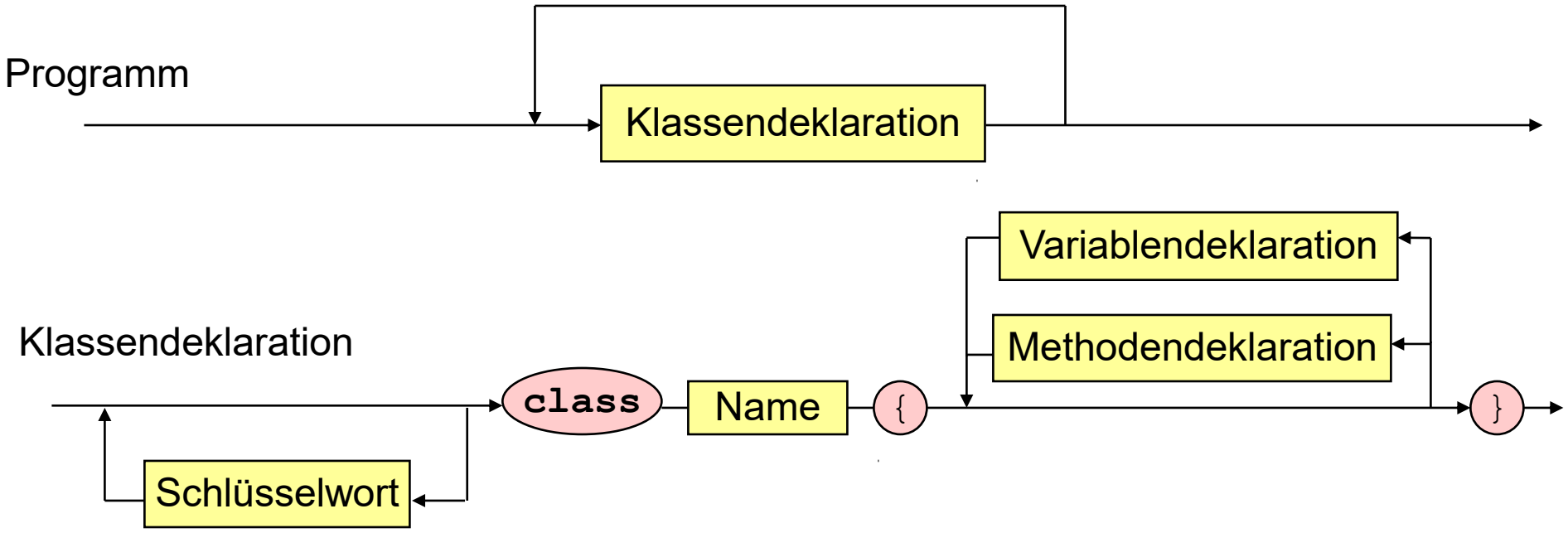
```
void main () {

     double [] laenge = new double [3];
     double [] breite = new double [3];
     int [] strichstaerke = new int [3];
     ...
     laenge [0] = laenge [1];
     breite [0] = breite [1];
     strichstaerke [0] = strichstaerke [1];

     double flaeche = laenge [0] * breite [0];
}
```

Objektorientierte Rechteck-Darstellung

```
public class Rechteck {

  double laenge, breite;

  int strichstaerke;

  double flaeche () {
     return laenge * breite;
  }}
```

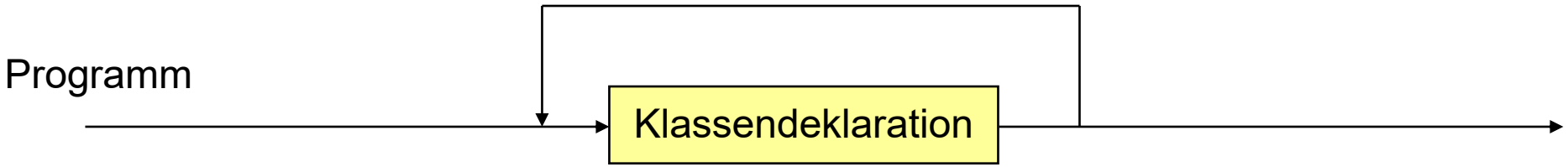
```
public class Rechteck_Programm {

  public static void main () {

     Rechteck r = new Rechteck (),
              s = new Rechteck (),
              t = new Rechteck ();
     ...
     r = s;

     double flaeche = r.flaeche ();
  }}
```

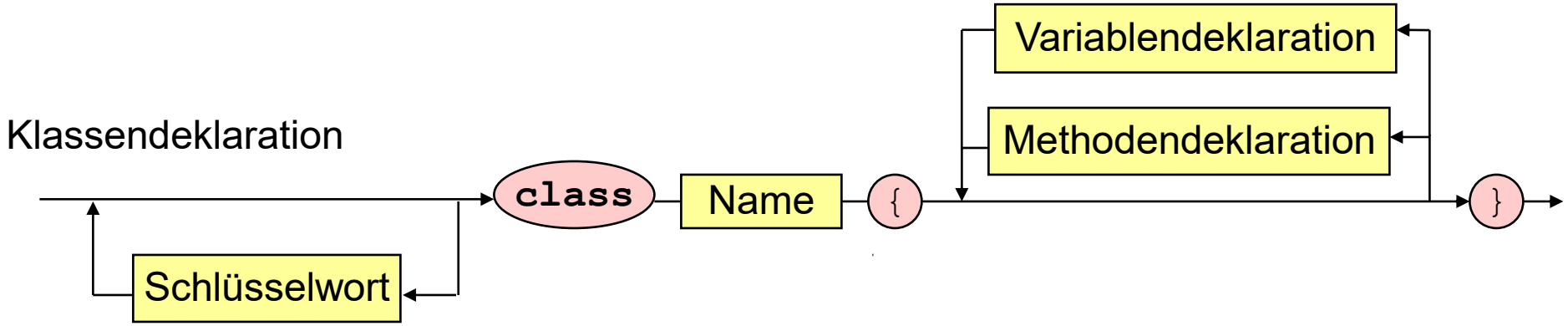
Klassen- und Methodendeklaration

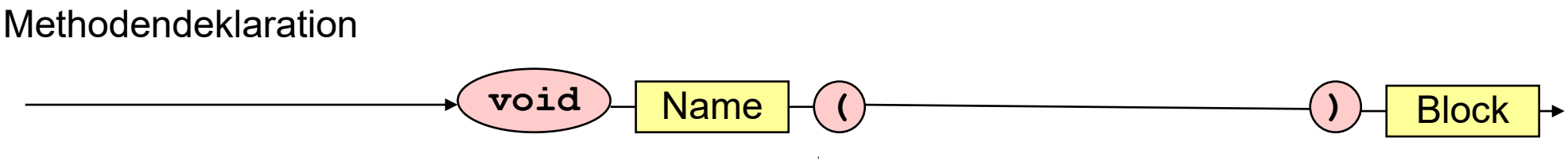


Schlüsselwort: **`public`**, **`static`**, **`final`**, ...

Klassen- und Methodendeklaration

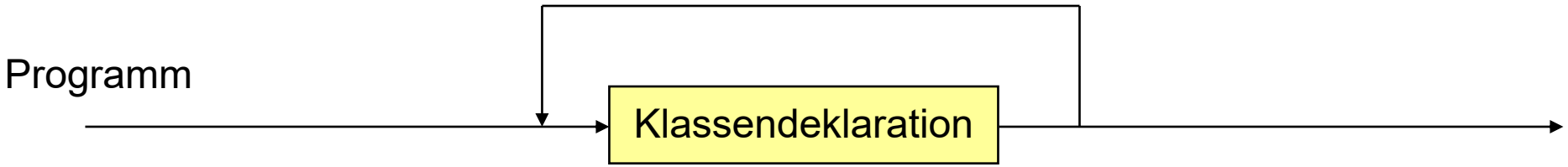


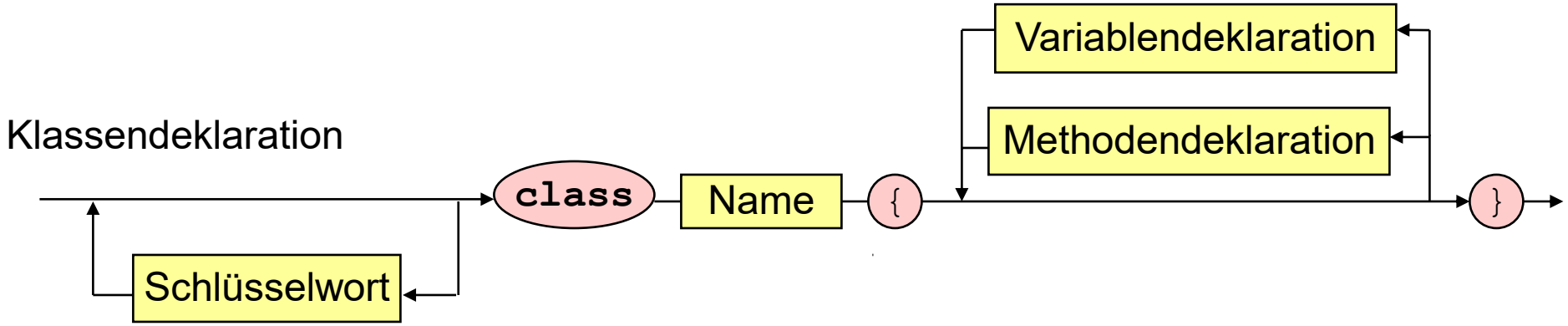


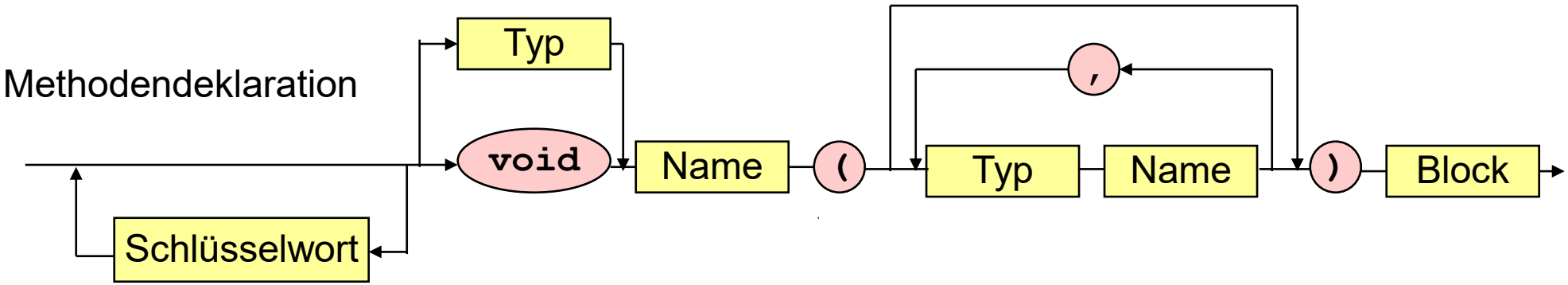


Schlüsselwort: **`public`**, **`static`**, **`final`**, ...

Klassen- und Methodendeklaration







Schlüsselwort: **`public`**, **`static`**, **`final`**, ...

Compact Source Files

Compact Source File = Datei ohne Klassendeklaration

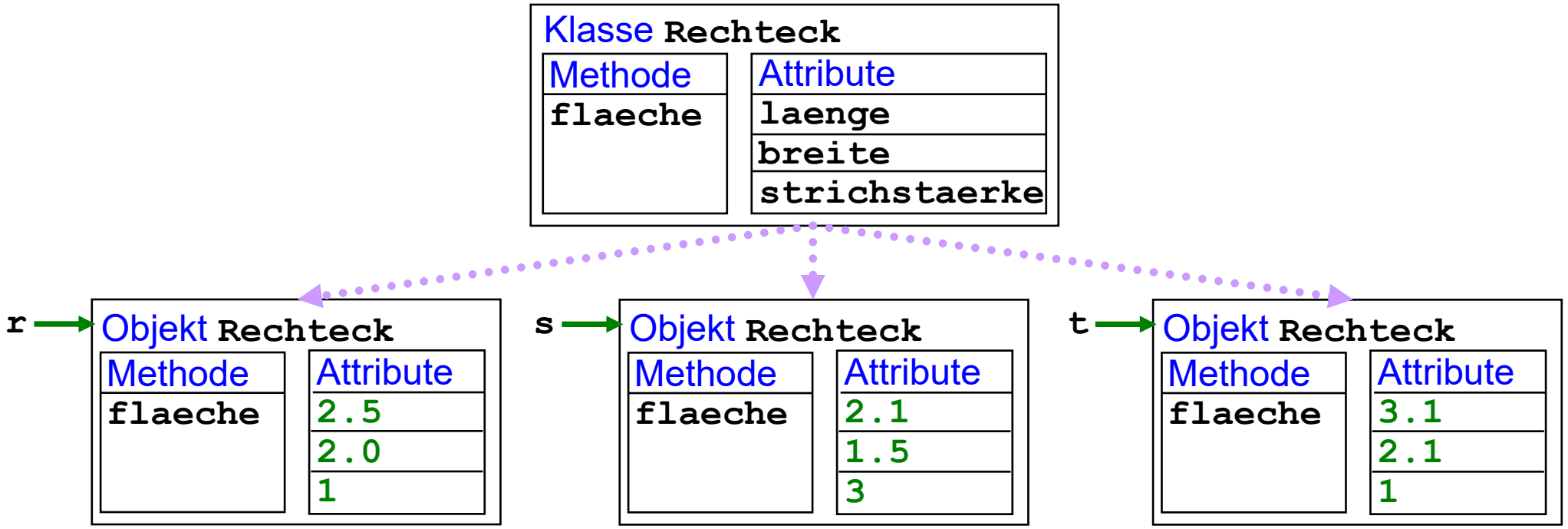
- Dadurch wird implizit eine Klasse deklariert.
- Name der Klasse wird automatisch generiert (ist nicht der Dateiname).
- Variablen und Methoden in der Datei gehören zu dieser Klasse.
- Man kann keine Objekte dieser Klasse selbst erzeugen (mit `new`).
- Datei muss `main` Methode enthalten.

Generell:

- Falls `main` von Klasse `K` nicht statisch ist, wird ein Objekt der Klasse mit `new K()` erzeugt, und `main` für dieses Objekt aufgerufen.
- Es darf `main()` und `main(String[] args)` geben. Letzteres hat Präferenz.

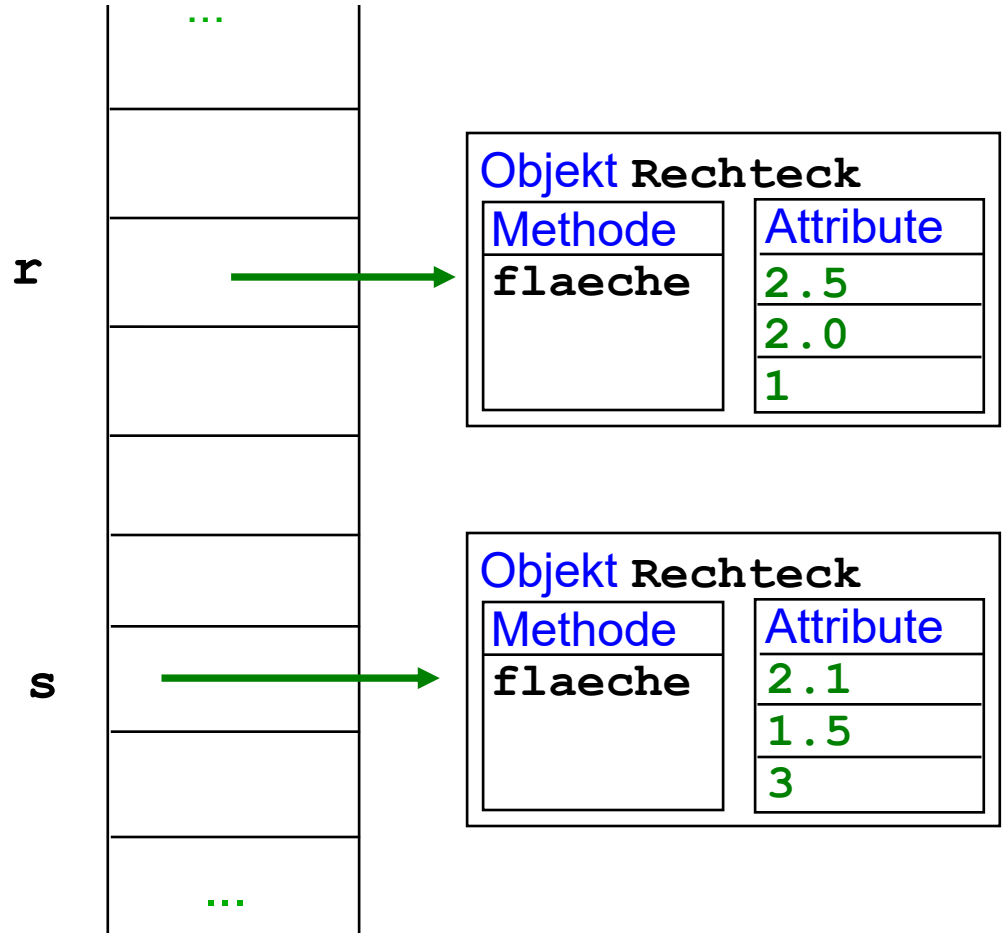
Objektorientierte Rechteck-Darstellung

```
public class Rechteck {

  double laenge, breite;

  int strichstaerke;

  double flaeche () {
     return laenge * breite;
  }}
```



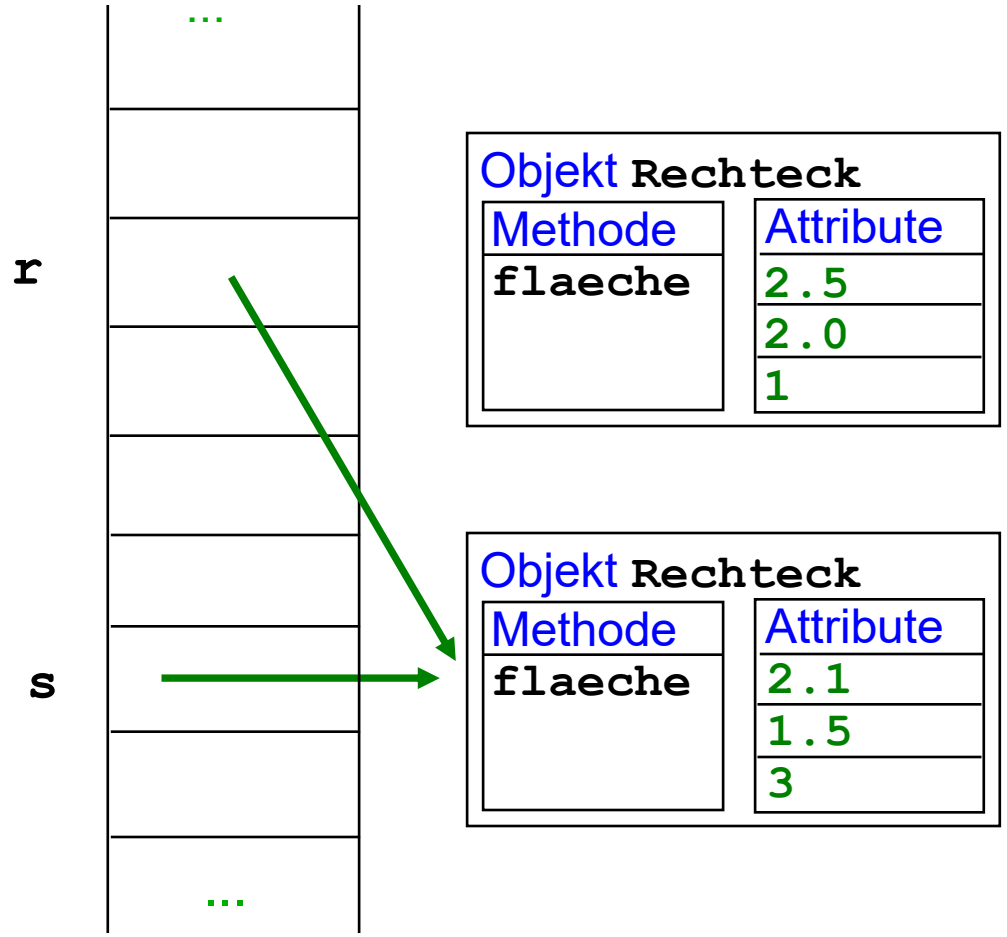
Referenzvariablen bei Klassentypen

```
Rechteck r = new Rechteck();

r.laenge = 2.5;
r.breite = 2.0;
r.strichstaerke = 1;

Rechteck s = new Rechteck();

s.laenge = 2.1;
s.breite = 1.5;
s.strichstaerke = 3;

r = s;

r.laenge = 4.6;
```



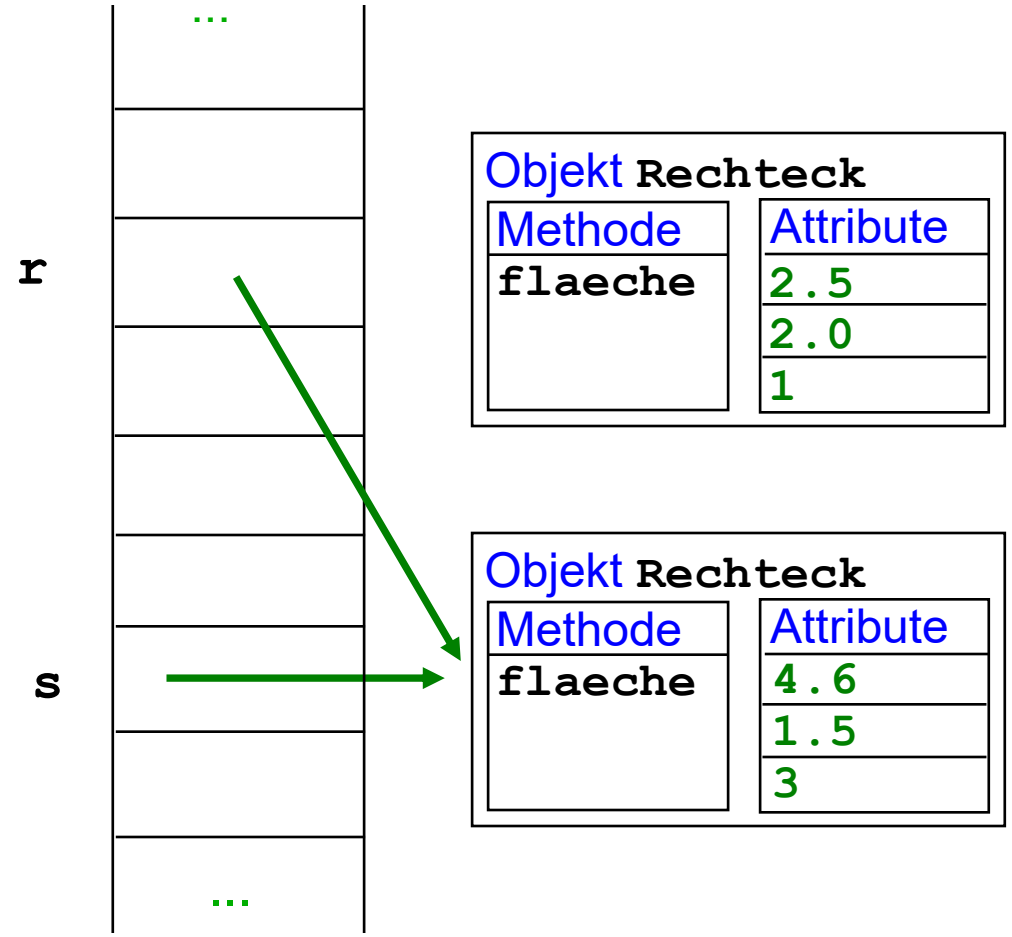
Referenzvariablen bei Klassentypen

```
Rechteck r = new Rechteck();

r.laenge = 2.5;
r.breite = 2.0;
r.strichstaerke = 1;

Rechteck s = new Rechteck();

s.laenge = 2.1;
s.breite = 1.5;
s.strichstaerke = 3;

r = s;

r.laenge = 4.6;
```



Referenzvariablen bei Klassentypen

```
Rechteck r = new Rechteck();

r.laenge = 2.5;
r.breite = 2.0;
r.strichstaerke = 1;

Rechteck s = new Rechteck();

s.laenge = 2.1;
s.breite = 1.5;
s.strichstaerke = 3;

r = s;

r.laenge = 4.6;
```



Zum Schluss: `s.laenge == 4.6`