Globalübung Programmierung

Globalübung Programmierung

- Verifikation von Code mit Arrays
- Arrays und Seiteneffekte
- Objekte und Seiteneffekte
- Spielereien

Verifikation von Arrays

Vorbedingung: a Array a[0], ..., a[n-1], n ≥ 0

```
res = false;
i = 0;
while (i < n) {
    if (a[i] == b) {
        res = true;
    }
    i = i + 1;
}
```

Nachbedingung: res = true gdw.
b in a[0], ..., a[n-1]

Verifikation von Arrays

Hoare-Regel für bedingte Anweisung

$$\frac{\langle A, B\rangle\ P\ \langle C\rangle \qquad\qquad A, \bar{B} \Rightarrow C}{\langle A\rangle\ \text{if (B) } \{P\}\ \langle C\rangle}$$

Programmieren mit Arrays

Vektorprodukt

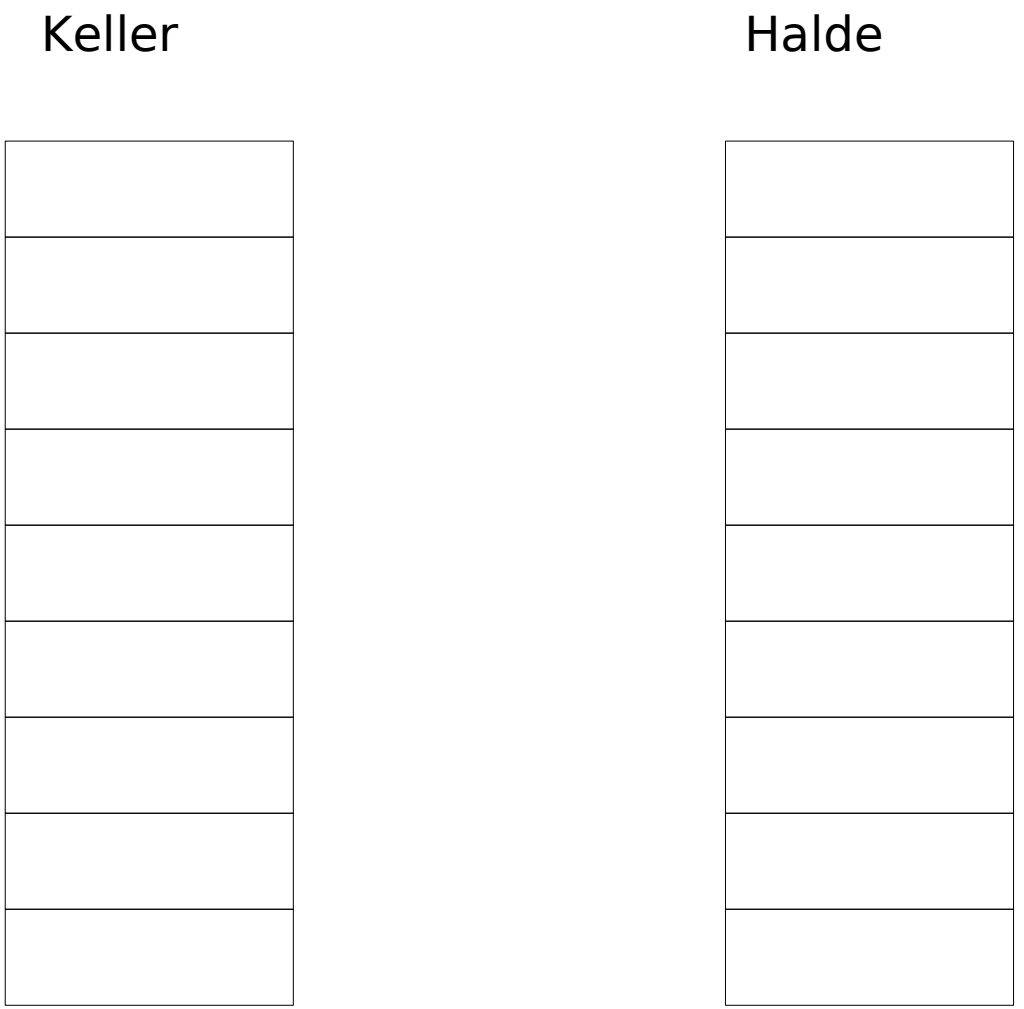
$$Sei \quad \vec{a} = \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix}, \quad \vec{b} = \begin{pmatrix} b_0 \\ b_1 \\ b_2 \end{pmatrix}.$$

$$Dann\,ist \quad \vec{c} = \begin{pmatrix} c_0 \\ c_1 \\ c_2 \end{pmatrix} = \vec{a} \times \vec{b}$$

$$mit \quad c_i = a_{(i+1) mod\ 3} * b_{(i+2) mod\ 3} - a_{(i+2) mod\ 3} * b_{(i+1) mod\ 3}$$

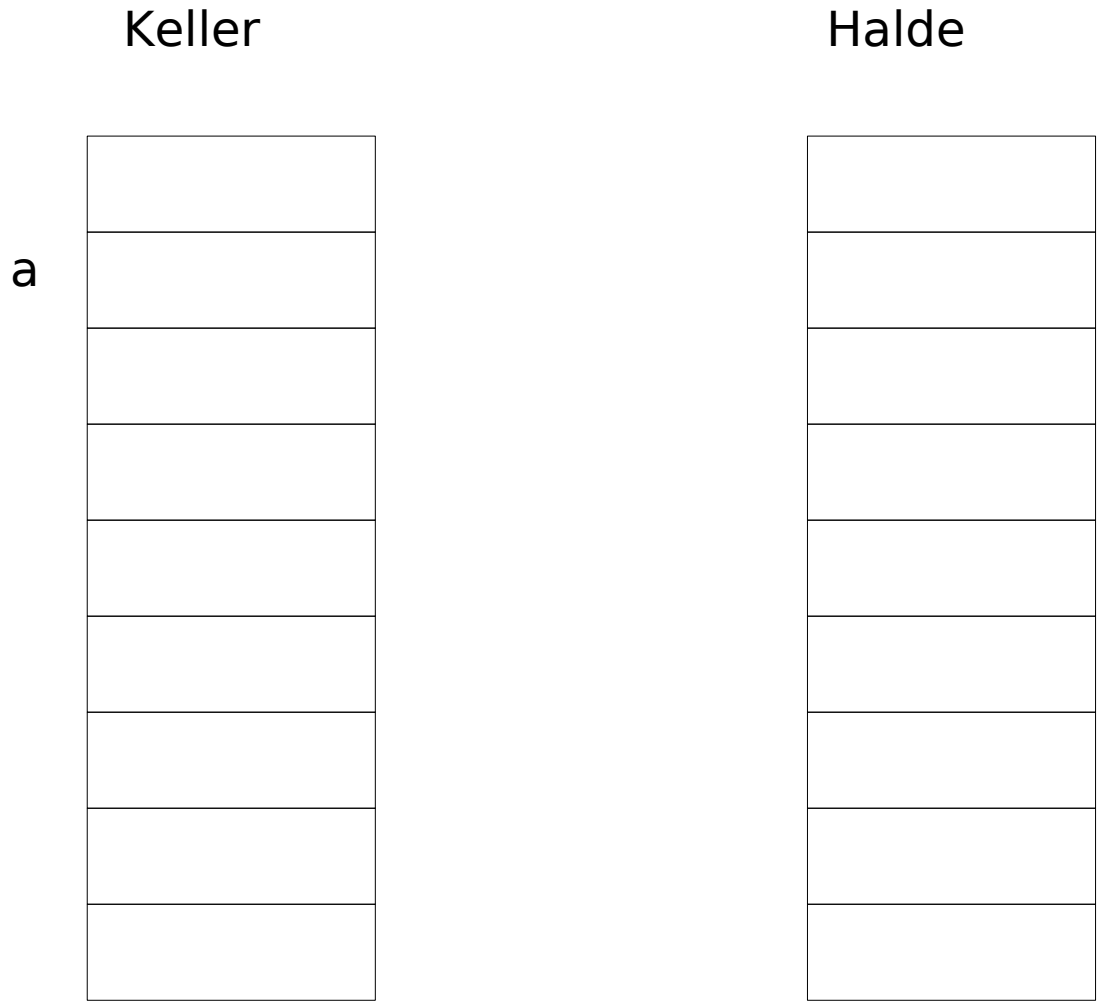
$das\,Vektorprodukt\,von\,\vec{a}\,und\,\vec{b}.$

Arrays im Speicher



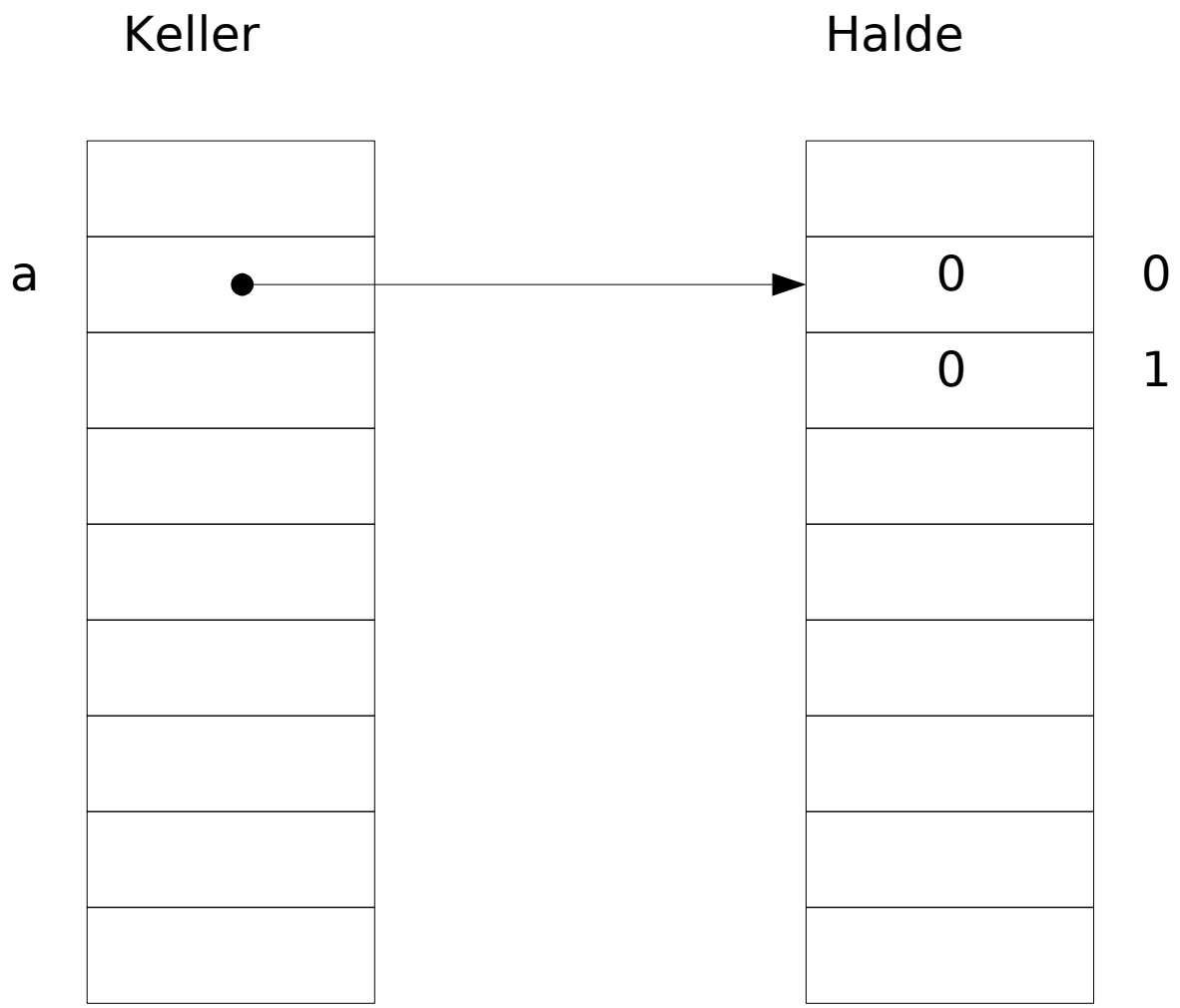
Arrays im Speicher

Int[] a;



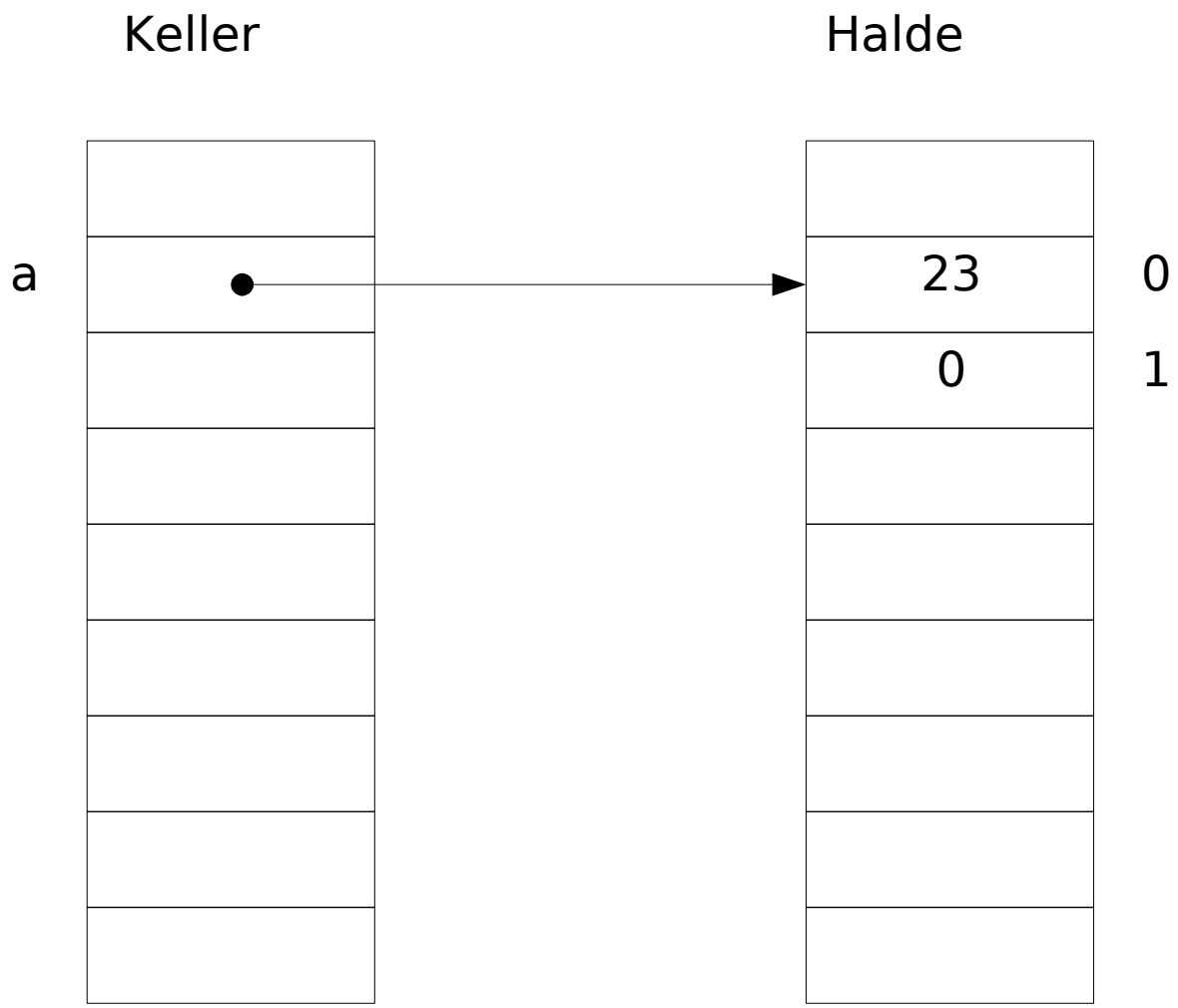
Arrays im Speicher

```
Int[] a;
a = new int[2];
```



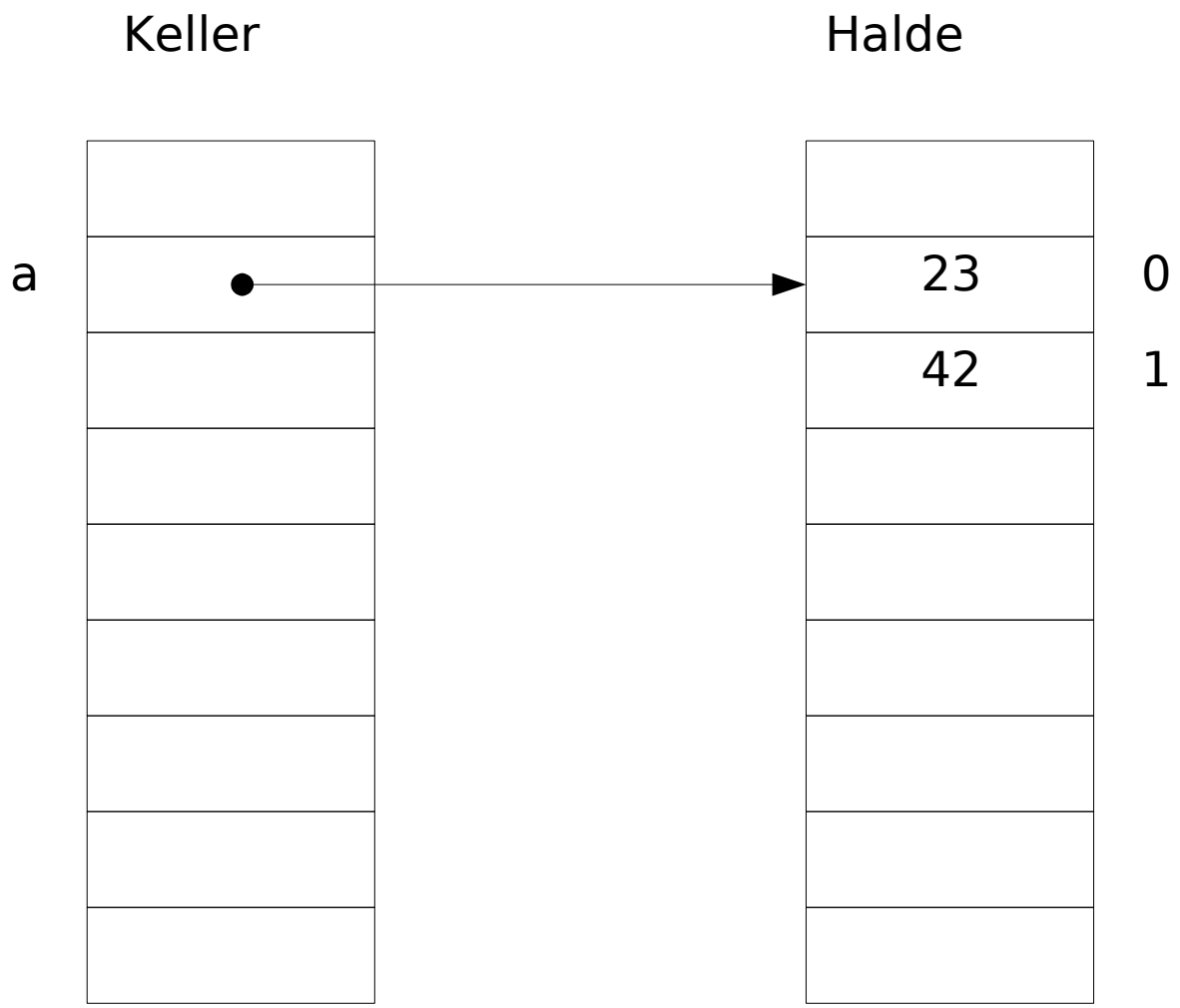
Arrays im Speicher

```
Int[] a;
a = new int[2];
a[0] = 23;
```



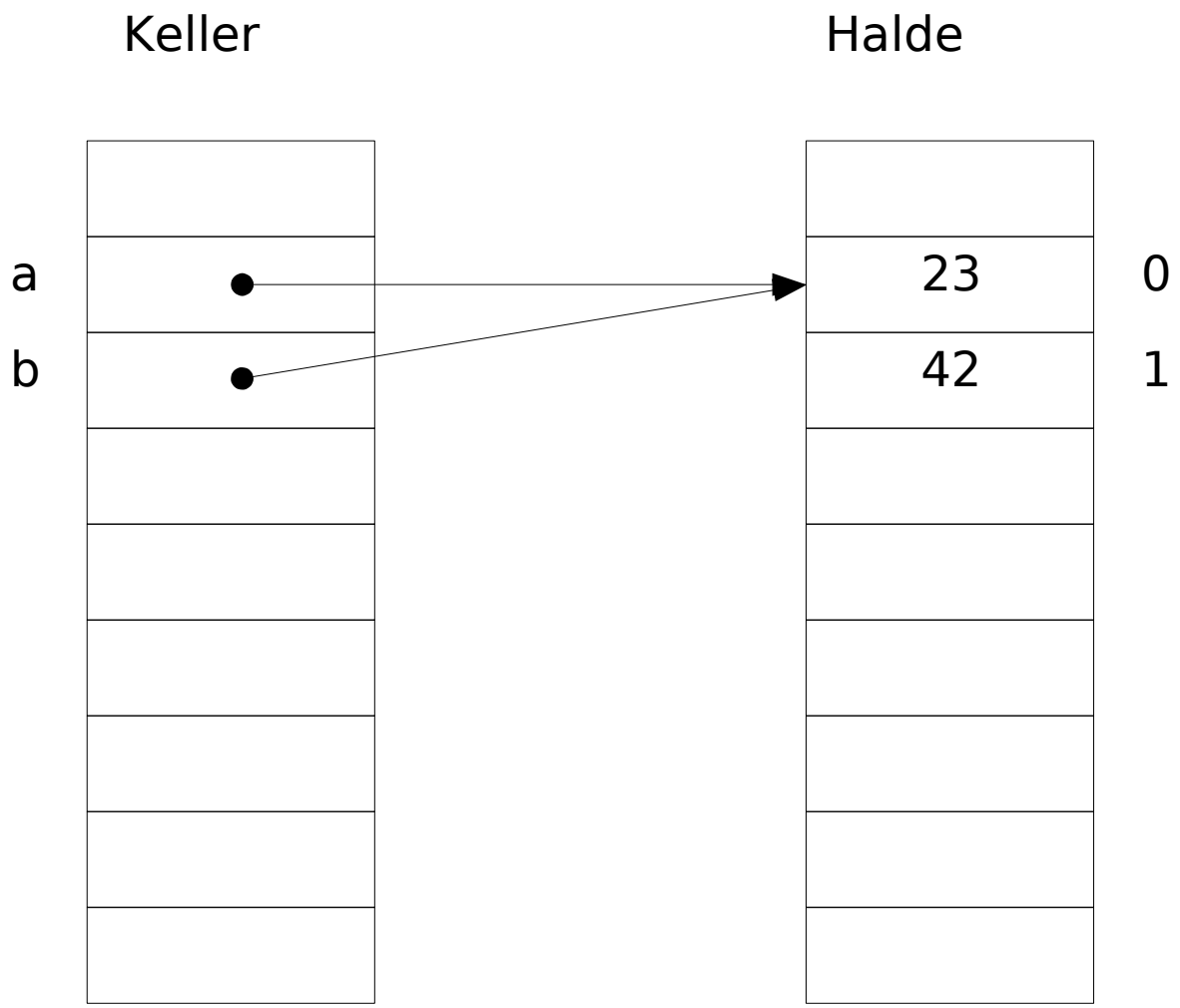
Arrays im Speicher

```
Int[] a;
a = new int[2];
a[0] = 23;
a[1] = 42;
```



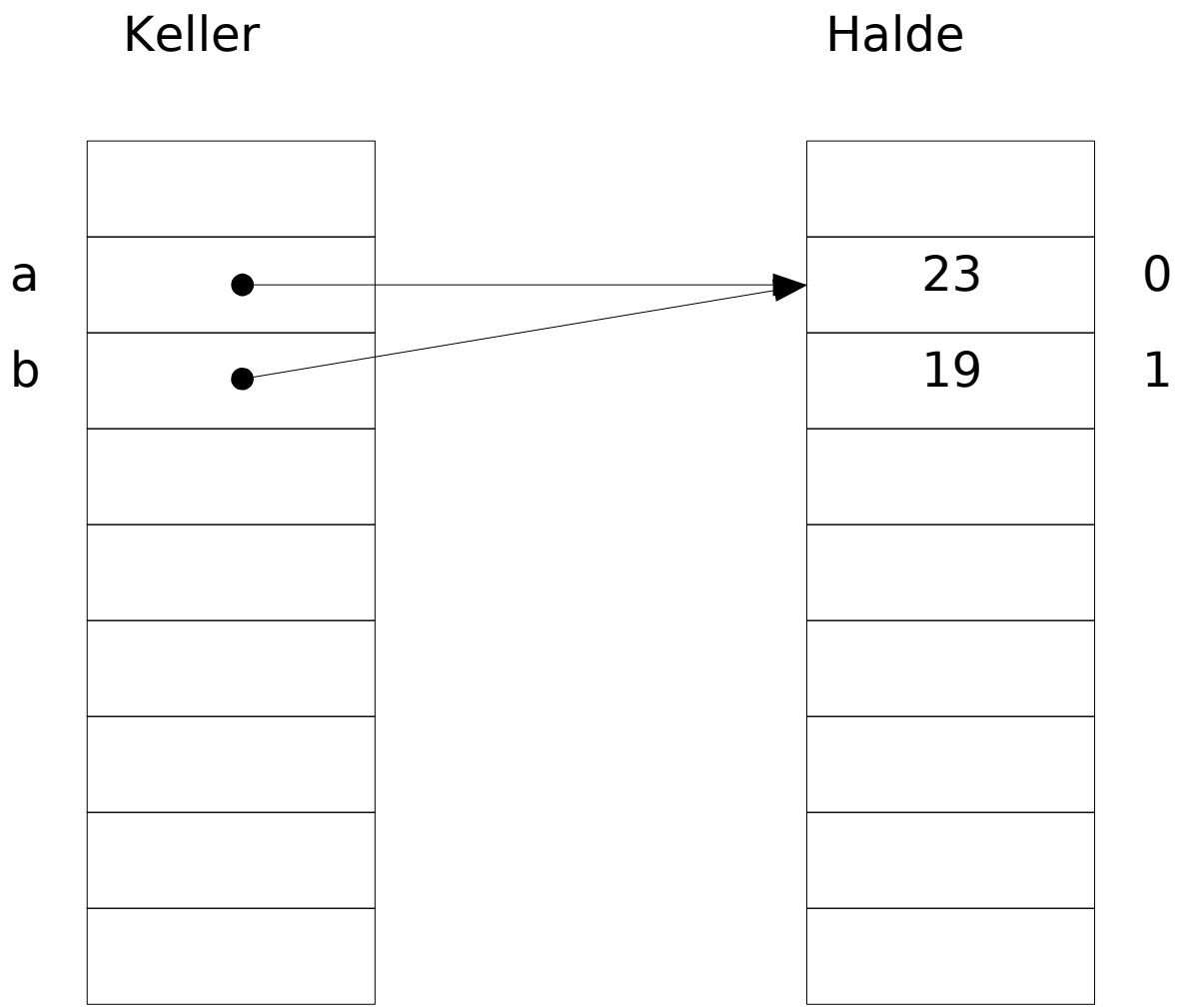
Arrays im Speicher

```
Int[] a;
a = new int[2];
a[0] = 23;
a[1] = 42;
int[] b = a;
```



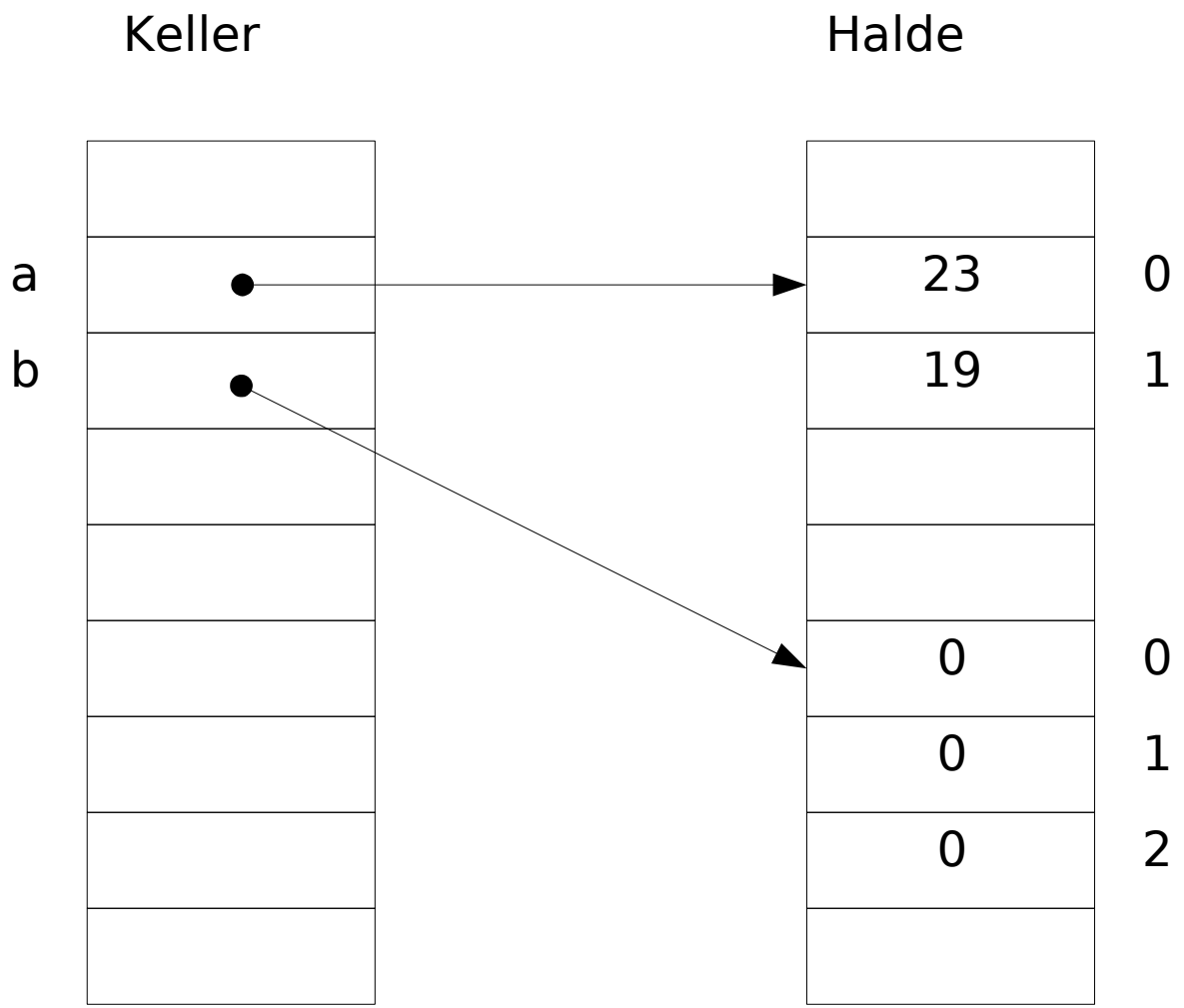
Arrays im Speicher

```
Int[] a;
a = new int[2];
a[0] = 23;
a[1] = 42;
int[] b = a;
b[1] = a[1]-a[0];
```



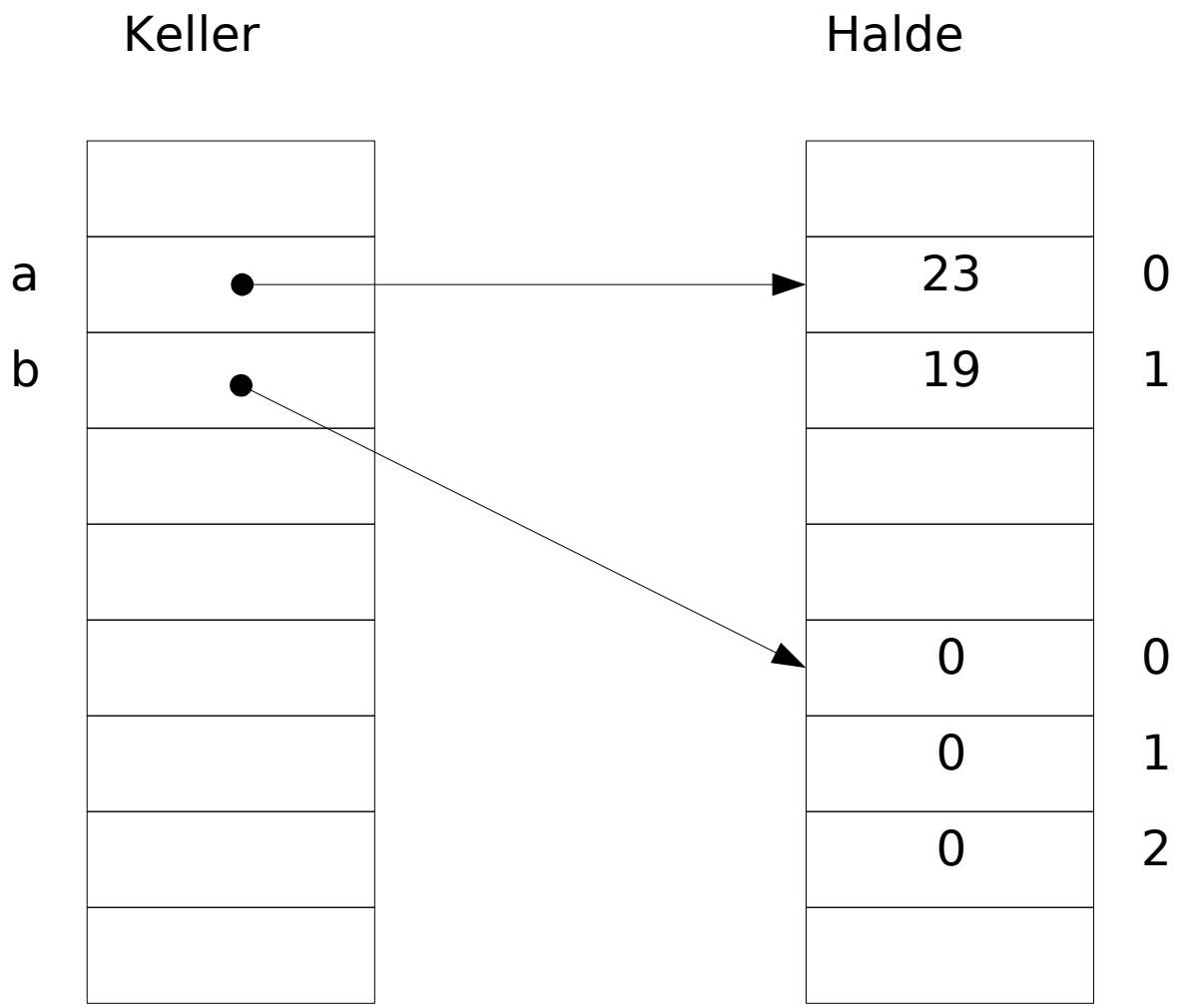
Arrays im Speicher

```
Int[] a;
a = new int[2];
a[0] = 23;
a[1] = 42;
int[] b = a;
b[1] = a[1]-a[0];
b = new int[3];
```



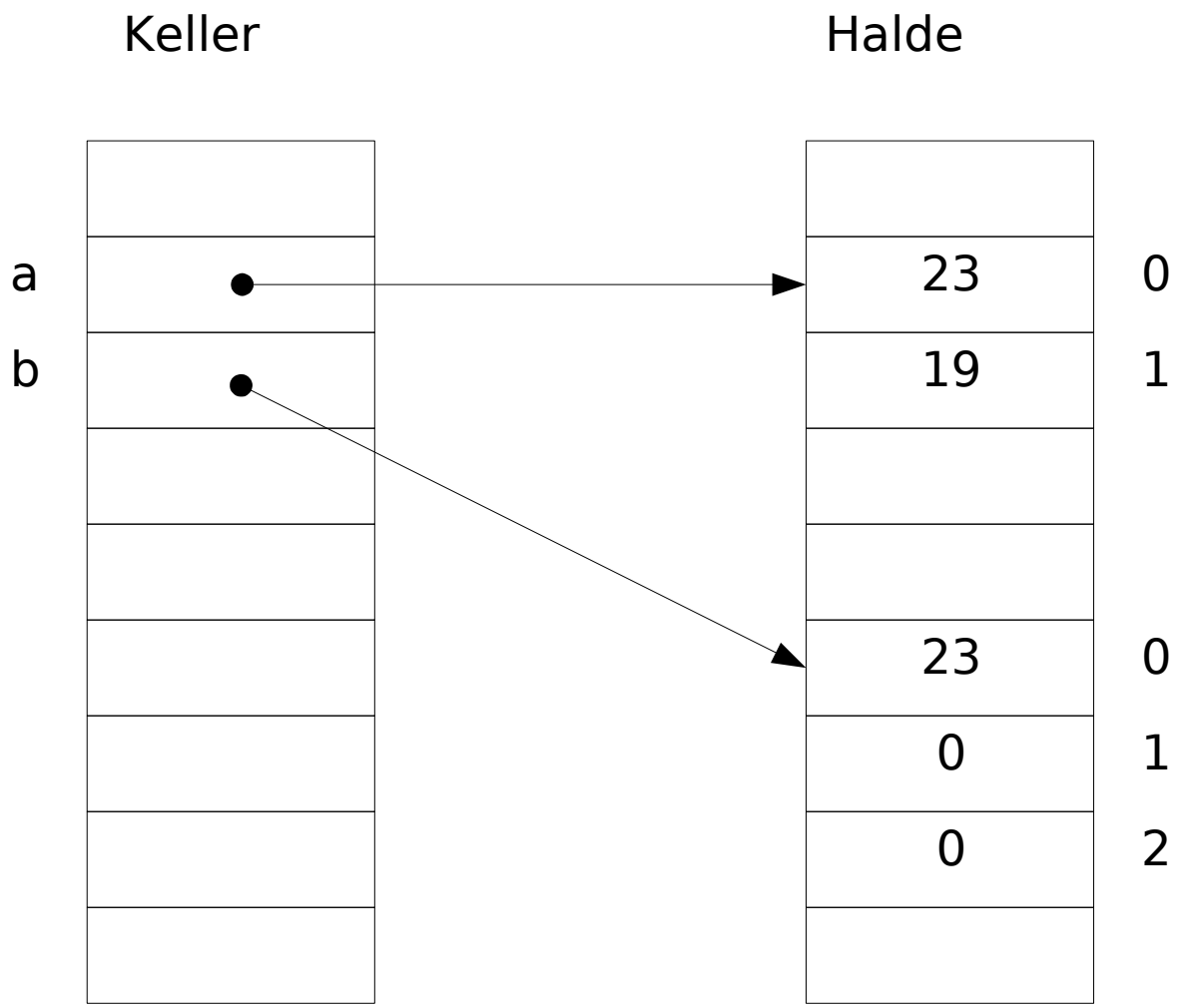
Arrays im Speicher

```
Int[] a;
a = new int[2];
a[0] = 23;
a[1] = 42;
int[] b = a;
b[1] = a[1]-a[0];
b = new int[3];
for (int i = 0;
      i < 2; i++) {
   b[i] = a[i];
}
```



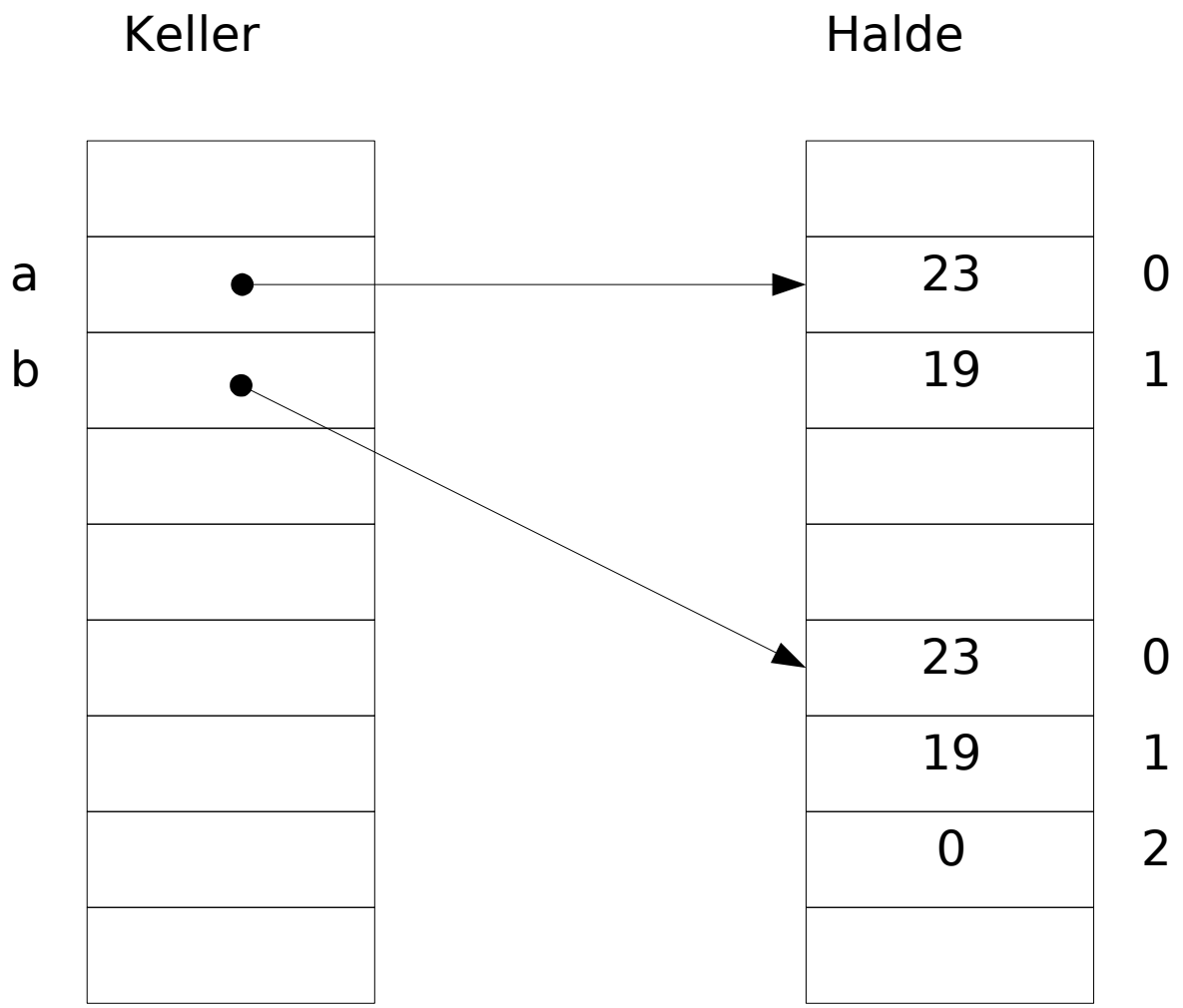
Arrays im Speicher

```
Int[] a;
a = new int[2];
a[0] = 23;
a[1] = 42;
int[] b = a;
b[1] = a[1]-a[0];
b = new int[3];
for (int i = 0;
      i < 2; i++) {
   b[i] = a[i];
}
```



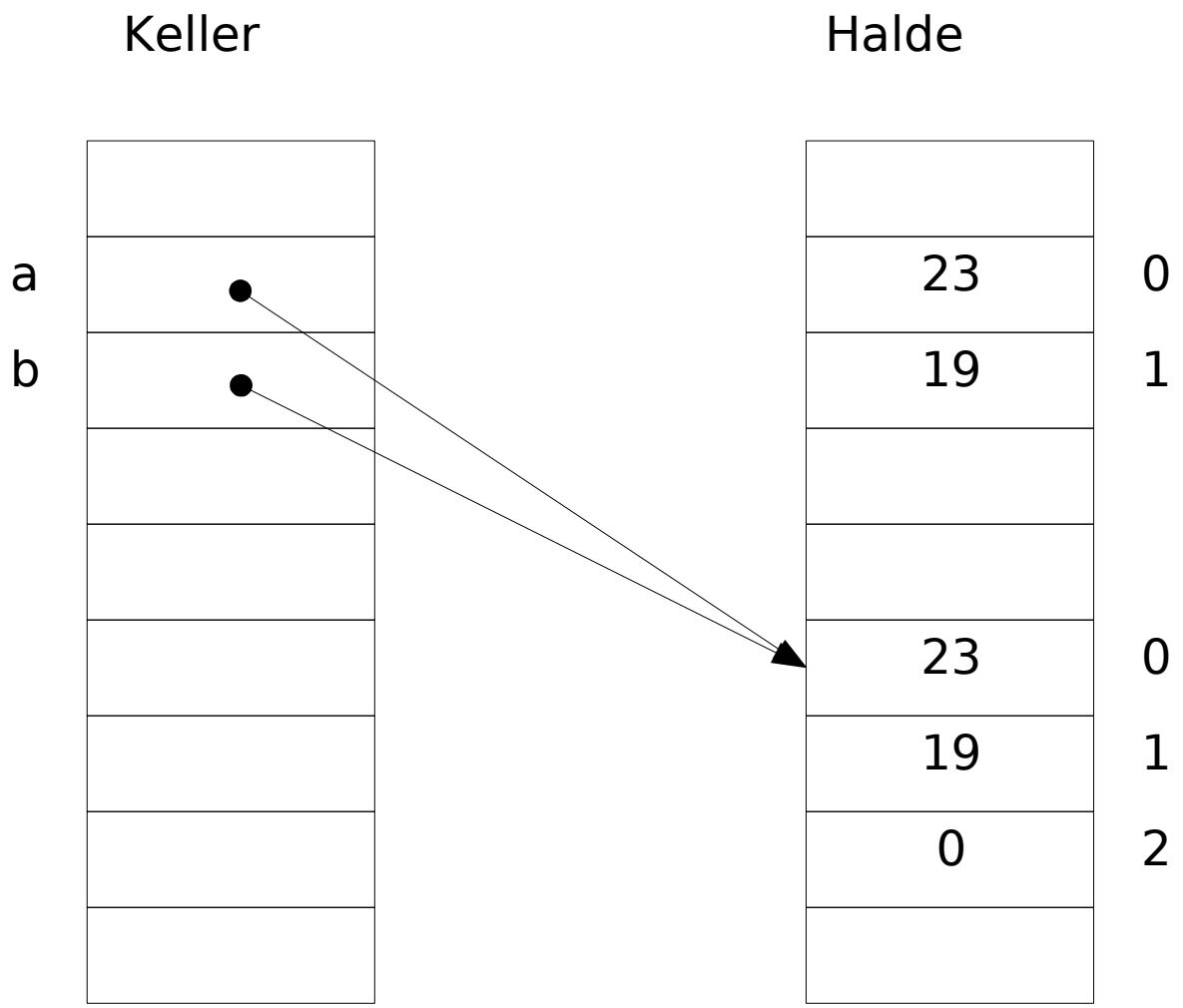
Arrays im Speicher

```
Int[] a;
a = new int[2];
a[0] = 23;
a[1] = 42;
int[] b = a;
b[1] = a[1]-a[0];
b = new int[3];
for (int i = 0;
      i < 2; i++) {
   b[i] = a[i];
}
```



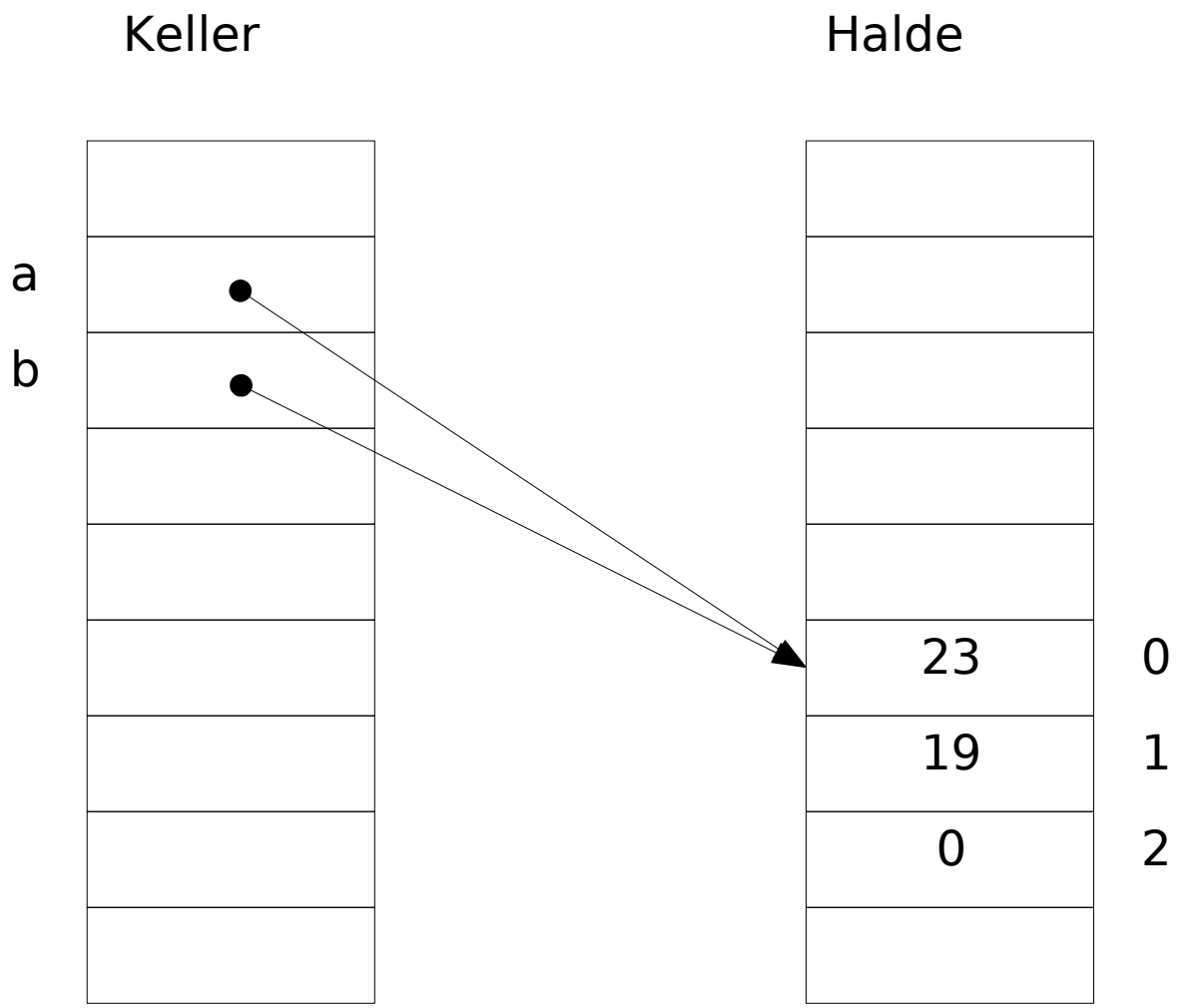
Arrays im Speicher

```
Int[] a;
a = new int[2];
a[0] = 23;
a[1] = 42;
int[] b = a;
b[1] = a[1]-a[0];
b = new int[3];
for (int i = 0;
       i < 2; i++) {
    b[i] = a[i];
}
a = b;
```



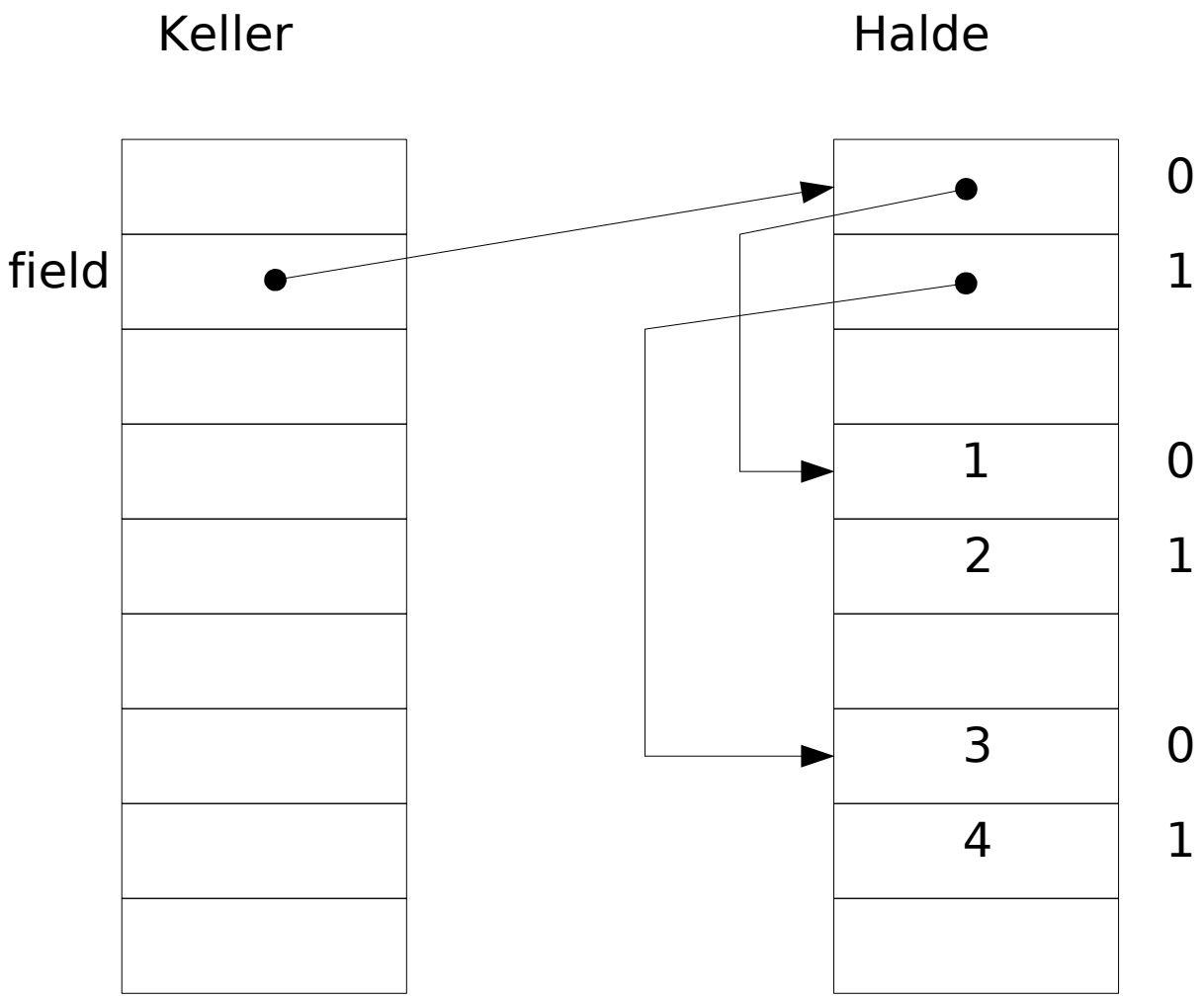
Arrays im Speicher

```
Int[] a;
a = new int[2];
a[0] = 23;
a[1] = 42;
int[] b = a;
b[1] = a[1]-a[0];
b = new int[3];
for (int i = 0;
      i < 2; i++) {
   b[i] = a[i];
}
a = b;
```



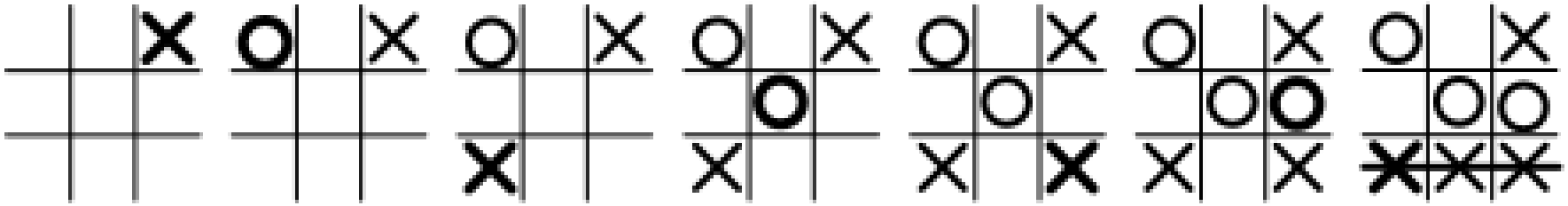
Arrays im Speicher

Int[][] field =
{{1,2},{3,4}}



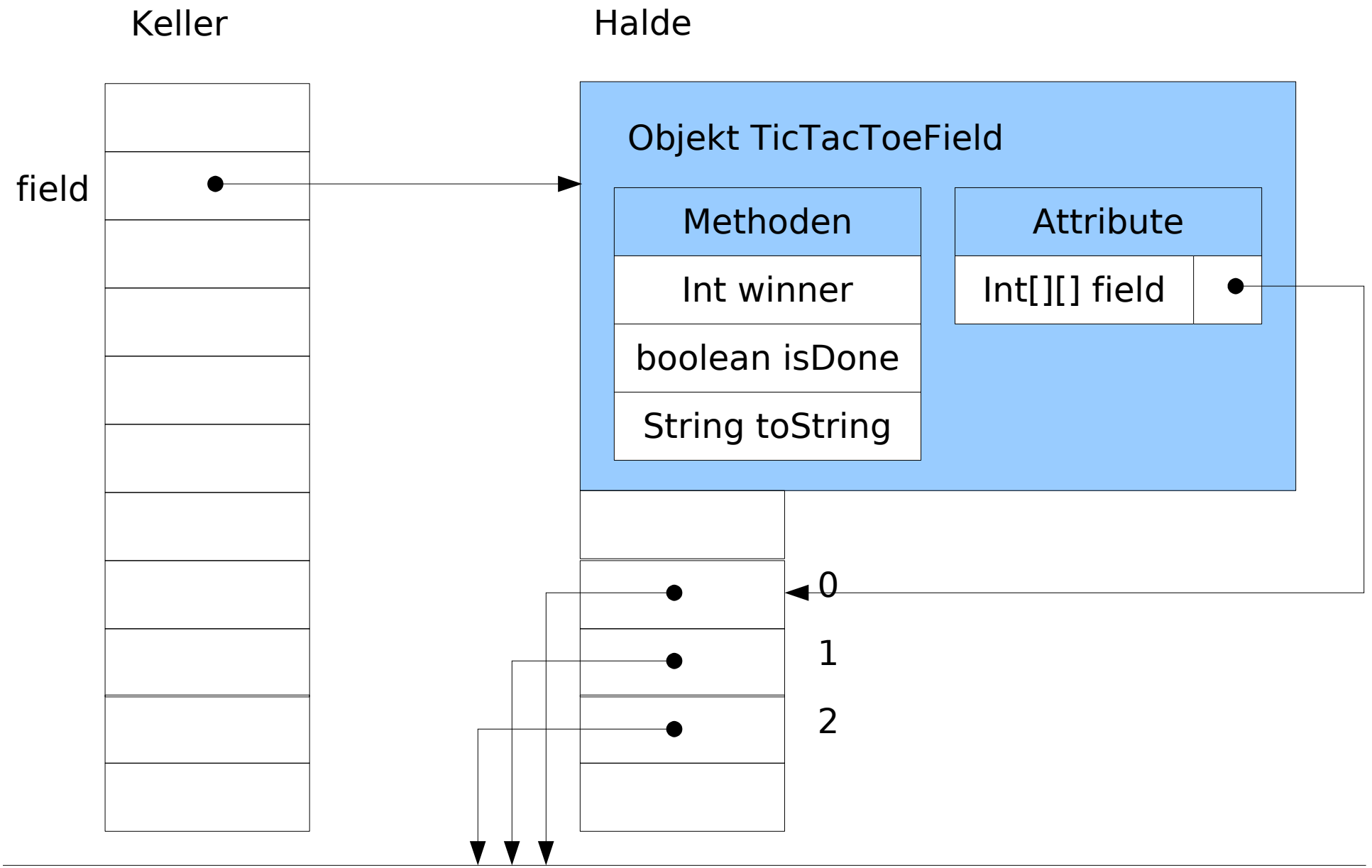
Programmieren mit Objekten

Spielname: Tic-Tac-Toe



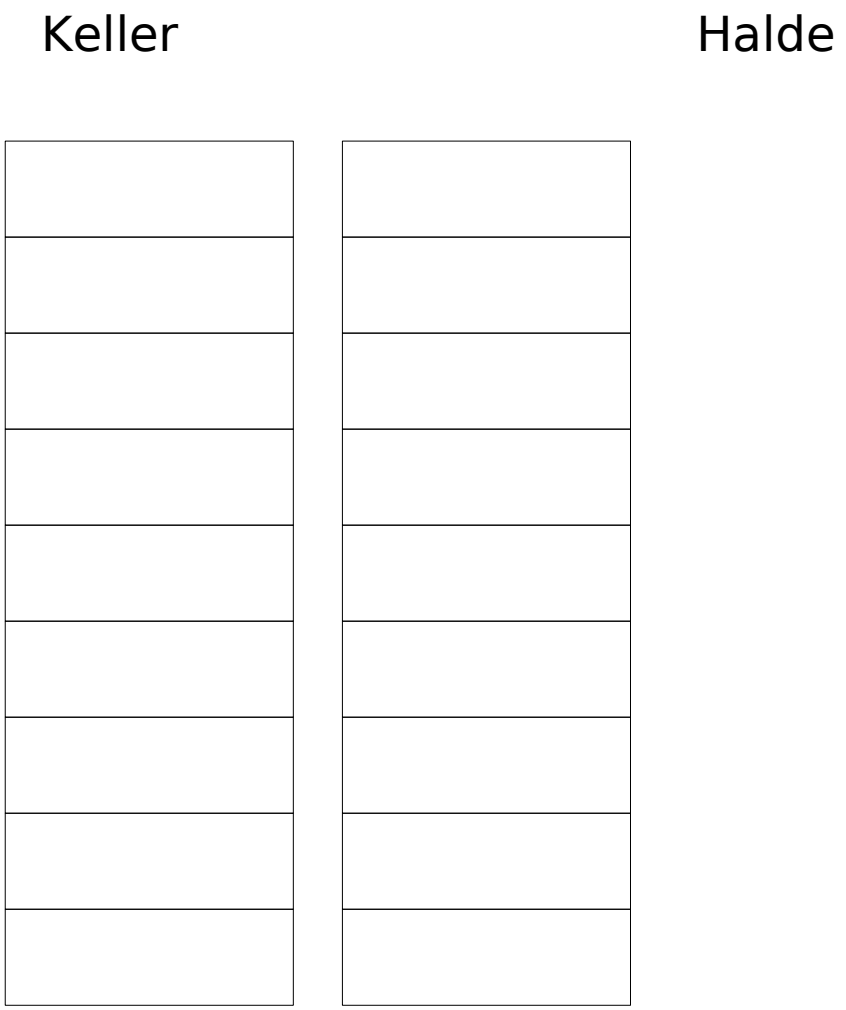
Spielfeld: 3x3 Positionen
Anzahl Spieler: 2 Spieler (X und O)
Zuege: Abwechselndes Setzen von X bzw.O
Ziel: 3 X oder 3 O in einer Zeile

Objekte im Speicher



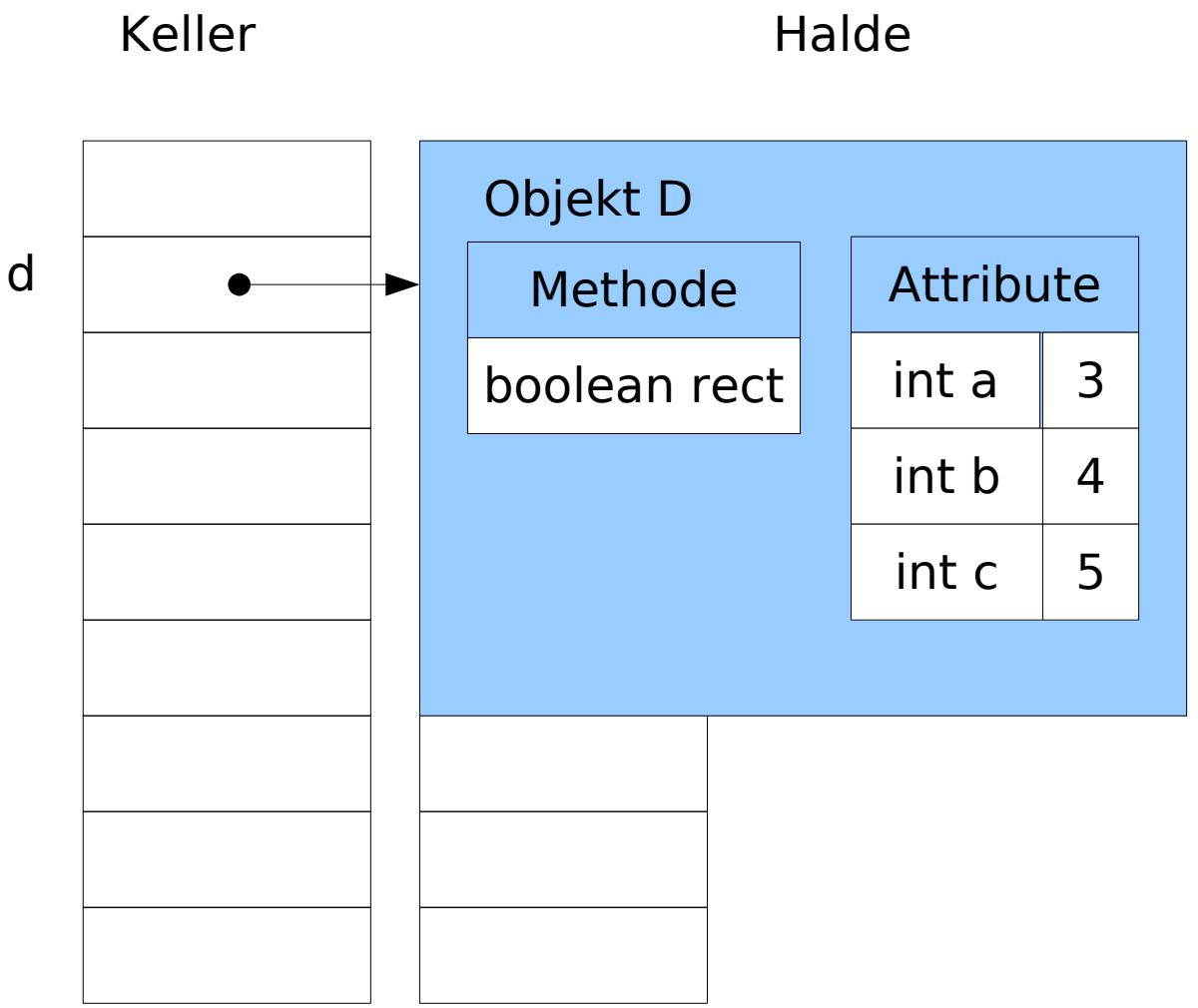
Objekte im Speicher

```
public class D {
  int a = 3, b = 4,
      c = 5;
  boolean rect() {
    return a*a+b*b
      == c*c;
  }
}
...
```



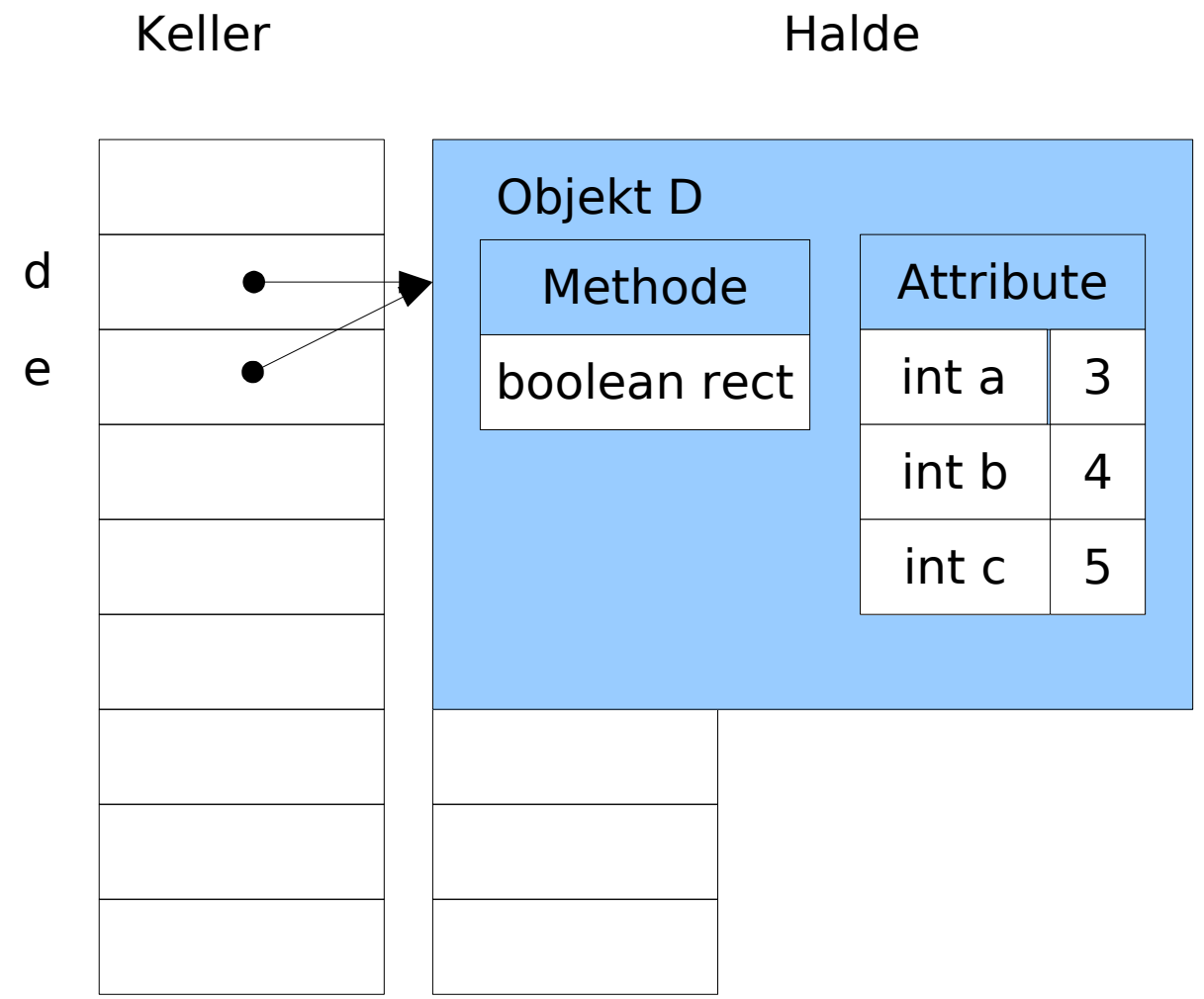
Objekte im Speicher

```
public class D {
  int a = 3, b = 4,
      c = 5;
  boolean rect() {
    return a*a+b*b
      == c*c;}}
...
D d = new D();
```



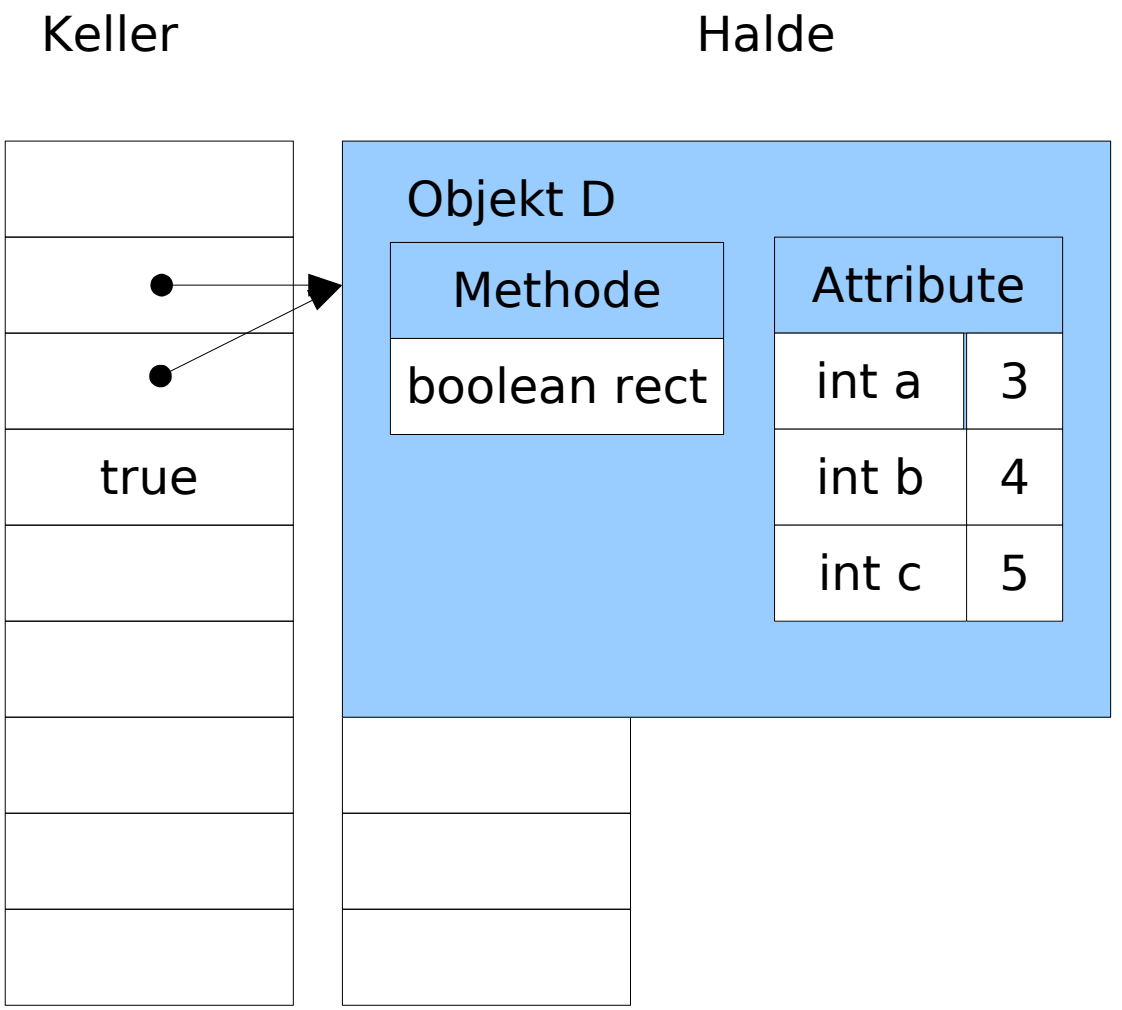
Objekte im Speicher

```
public class D {
  int a = 3, b = 4,
      c = 5;
  boolean rect() {
    return a*a+b*b
      == c*c;}}
...
D d = new D();
D e = d;
```



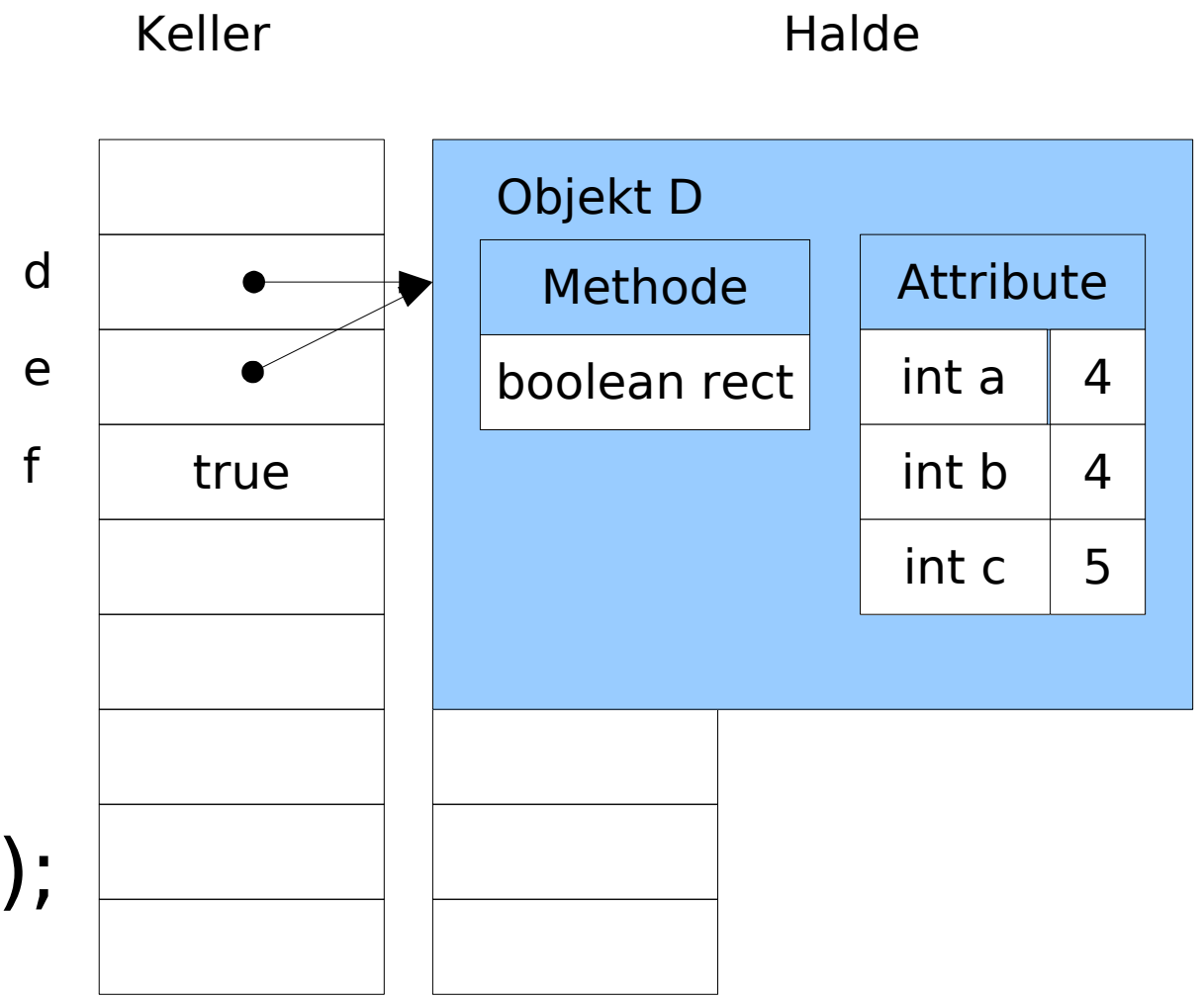
Objekte im Speicher

```
public class D {
  int a = 3, b = 4,
      c = 5;
  boolean rect() {
    return a*a+b*b
      == c*c;}}
...
D d = new D();
D e = d;
boolean f = d.rect();
```



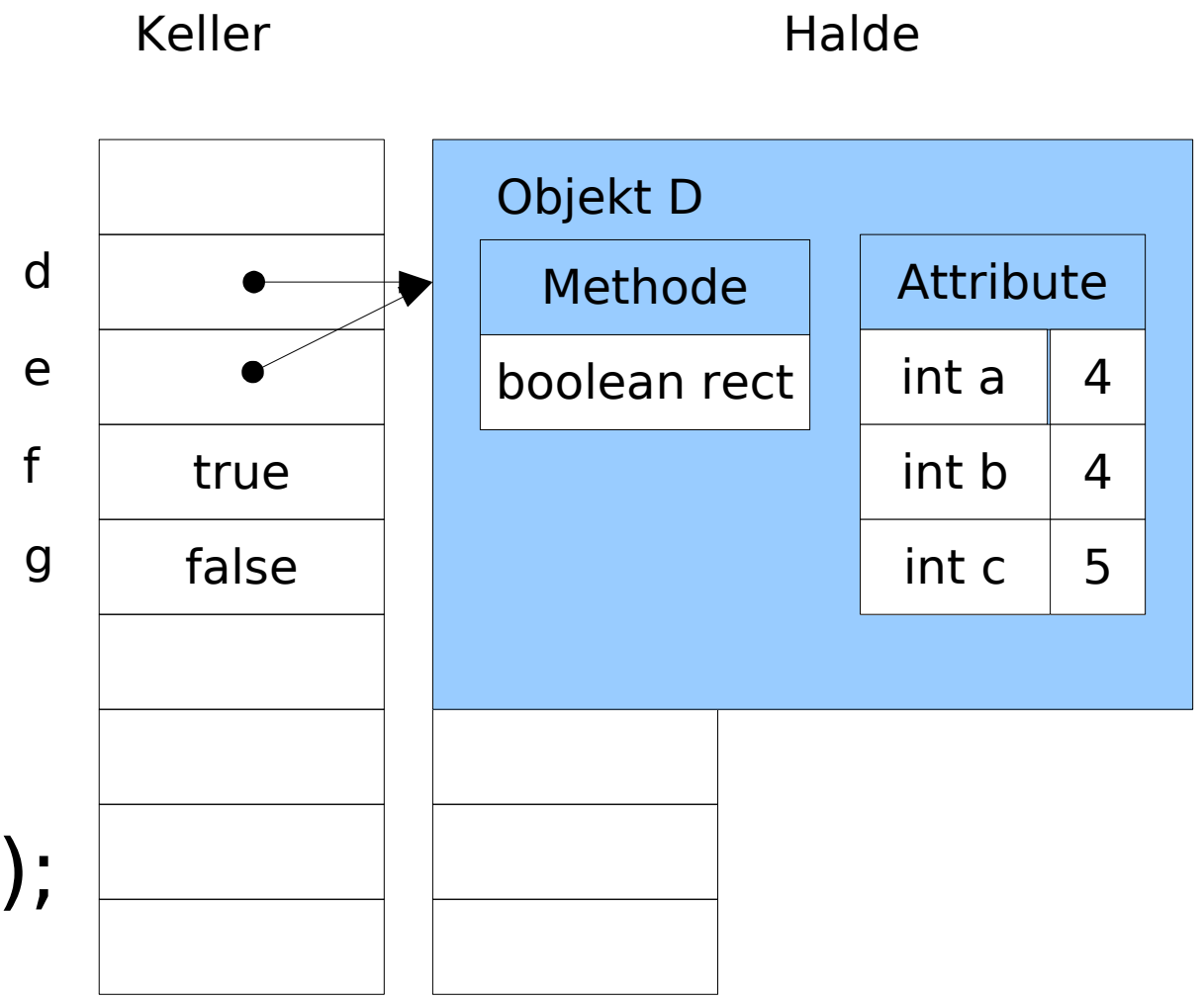
Objekte im Speicher

```
public class D {
  int a = 3, b = 4,
      c = 5;
  boolean rect() {
    return a*a+b*b
      == c*c;}}
...
D d = new D();
D e = d;
boolean f = d.rect();
e.a = 4;
```



Objekte im Speicher

```
public class D {
  int a = 3, b = 4,
      c = 5;
  boolean rect() {
    return a*a+b*b
      == c*c;}}
...
D d = new D();
D e = d;
boolean f = d.rect();
e.a = 4;
boolean g = d.rect();
```



Globalübung Programmierung