

8.1.1) Kein Fehler

```
(1) A obj = new B();    // es wird ein Objekt mit Namen obj der ex-
                        // tended Klasse B erzeugt.
(2) obj.setA(1, 2);    // Es wird auf die Methode setA der SUPER-
                        // Klasse A zugegriffen und hiermit a der
                        // Wert 1 und b der Wert 2 zugewiesen.
(3) obj.out();         // Es wird auf die Methode out der exten-
                        // ded Class zugegriffen und diese ausge-
                        // führt. Diese startet zuerst die SUPER-
                        // Klassen-Methode gleichen Namens, welche
                        // nach einem Zeilenumbruch "a=1, b=2" aus-
                        // gibt. Anschließend hängt die extended
                        // Klasse die Ausgabe ", c=0" an.
(4) ((B) obj).setB(2, 3, 4); // Es wird Methode setB der extended
                        // Klasse aufgerufen. Diese führt zu-
                        // erst die Methode SetA der SUPER-
                        // Klasse aus, welche den Werten a und
                        // b neue Werte zuweist und anschlies-
                        // send auch c. Die neuen Werte sind:
                        // a=2, b=3, c=4
(5) obj.out();         // Nun wird wieder die Methode out der ex-
                        // tended Klasse aufgerufen, wodurch dank
                        // der zuvor ereigneten neuen Wertzuweisung
                        // für a, b und c sich nur die Ausgabe ver-
                        // ändert hat. Die gesamte Ausgabe lautet
                        // nun:
                        // "a=2, b=3, c=4".
```

8.1.2) Fehler in Zeile (4): 'Exception in thread "main" java.lang. ClassCastException: A cannot be cast to B'

```
(1) A obj = new A();    // Es wird ein neues Objekt mit Namen obj der
                        // Klasse A angelegt.
(2) obj.setA(1, 2);    // Über die Methode SetA der Klasse A wird der
                        // Variablen a der Wert 1 und der Variablen b
                        // der Wert 2 zugewiesen
(3) obj.out();         // Es wird die Methode out nach einem Zeilenum-
                        // bruch die Zeile "a=1, b=2" ausgegeben.
(4) ((B)obj).setB(1, 2, 3); // Es wird versucht das Objekt der Klasse
                        // A auf Klasse B umzucasten um anschlies-
                        // send die Methode setB der extended Klas-
                        // se aufzurufen.
                        // Da jedoch Objekte nicht umgecastet wer-
                        // den können entsteht ein Syntaxfehler.
(5) ((B)obj).out();
```

8.1.3) Fehler in Zeile (3): 'a has private access in A' **Fehler in Zeile (5):** 'c has private access in B'

```
(1) B obj = new B();    // Es wird ein neues Objekt der Klasse B mit Namen
                        // obj angelegt.
(2) obj.setB(4, 5, 6); // Über die Methode setB werden den Variablen a, b
                        // und c neue Werte zugeordnet. a=4, b=5, c=6
(3) obj.a = 3;         // Es wird versucht der privaten Variablen a aus
                        // der SUPER-Klasse der Wert 3 zuzuordnen.
                        // Da jedoch das Variablen-Attribut private ist,
                        // entsteht dadurch ein Syntaxfehler, da auf pri-
                        // vate Variablen nur durch Methoden der Klasse
                        // Schreibzugriff besteht.
```

```

(4) obj.b = 4;          // Über das Objekt wird der geschützten Variablen
                        // b der Wert 4 zugeordnet, da durch die Vererbung
                        // zur extended Klasse B der Schreibzugriff auf öf-
                        // fentlich gesetzt wurde.
(5) obj.c = 5;          // Hier wird versucht der privaten Variablen c aus
                        // der Klasse B der Wert 5 zuzuordnen. Es ergibt
                        // das gleiche Ergebnis wie in Zeile (3).

```

8.1.4) Fehler in Zeile (1) (3x): ';' expected

```
A[] obj = f new A(), new B() g;
```

Fehler in Zeile (1): 'not a statement A[] obj = f new A(), new B() g;'

```

(1) A[] obj = f new A(), new B() g; // Es soll wahrscheinlich versucht
                                     // Array der Klasse A oder B zu er-
                                     // zeugen. Jedoch damit dieses ge-
                                     // lingen kann müsste auf der rech-
                                     // ten Quelltextzeilenseite ein for-
                                     // mal korrekter Ausdruck stehen,
                                     // je nachdem aus welcher Klasse (A
                                     // oder B die Elemente des Arrays
                                     // stammen "{new A(), new B(),
                                     // ...}";".
                                     // Des weiteren bleibt zu beachten,
                                     // das für eine weitere fehlerfreie
                                     // Ausführung der weiteren Zeilen
                                     // mindestens drei Array-Elemente aus
                                     // den Klassen A oder B vorhanden
                                     // sein müssen.
                                     // Durch diese Syntax entstehen meh-
                                     // rere Syntaxfehler, da der Compil-
                                     // er nicht entscheiden kann, was
                                     // er nun wirklich tun soll.
(2) obj[0].setA(1, 2); // Hier sollte eigentlich dem ersten Objekt im
                       // obj-Array über die Methode setA der Klasse
                       // A der Variablen a der Wert 1 und der Varia-
                       // blen b der Wert 2 zugewiesen werden.
(3) obj[1].setA(1, 2); // An dieser Stelle sollte selbiges mit dem
                       // zweiten Array-Objekt zugewiesen werden.
(4) obj[2].setA(2, 3); // Hier sollte dem 3. Array-Objekt für über
                       // die A-Klassen-Methode A der Variablen a
                       // der Wert 2 und der Variablen b der Wert
                       // 3 zugewiesen werden.
(5) for (int i = 0; i<obj.length; i++) obj[i].out();
    // Hier würde eine Schleife ausgeführt, die
    // für alle Array-Felder die Methode out
    // aufrufe, für Objekte der Klasse A die
    // Klasse A-Methode und für Objekte der Klas-
    // se B die Klasse-B-Methode.
    // Bei Klasse-B-Objekten wird an die Ausgabe,
    // die ich hier vermerke am Ende ein ", c=0"
    // angehängt.
    // Somit wird für Objekt 0 und 1 jeweils nach
    // einem Zeilenumbruch "a=1, b=2" ausgegeben,
    // für Objekt 2 dagegen nach dem Zeilenumbruch
    // "a=2, b=3", für alle weiteren Objekte nach
    // dem Zeilenumbruch "a=0, b=0"

```