3. Geben wir die Variable `s.var` direkt aus, so erhalten wir wie erwartet erneut die 2 als Ergebnis.

4. Führen wir jedoch eine Umwandlung der Referenz in den Typ `Vater` durch, so erhalten wir völlig überraschend als Ergebnis die 1. Das liegt nun daran, dass für Variablen keine dynamische Bindung ins Spiel kommt, sondern der Compiler bereits zur Übersetzungszeit abhängig vom Typ der Referenz entscheidet, auf welche der Variablen zugegriffen wird.

Was wollte uns dieser Abschnitt also sagen? Das Prinzip des Polymorphismus bzw. das dynamische Binden greift lediglich bei Methoden, *nicht* bei Variablen.

9.3.2 Überschreiben von Methoden verhindern durch `final`

Wollen wir dafür sorgen, dass eine Methode in allen Unterklassen in der gleichen Version vorliegt, die Polymorphie also ausschalten, können wir wiederum das Schlüsselwort **`final`** einsetzen. Ähnlich der Vorgehensweise bei der Markierung von ganzen Klassen, die nicht mehr erweitert werden können, verhindert der Modifizierer **`final`** vor einer Methoden-Deklaration das Überschreiben dieser Methode in einer Unterklasse. Definieren wir also eine Klasse

```
1 public class Papa {
2   public final void singe() {
3     System.out.println("La la la la la ...");
4   }
5 }
```

und versuchen anschließend in der Klasse

```
1 public class Kind extends Papa {
2   public void singe() {
3     System.out.println("Do Re Mi Fa So ...");
4   }
5 }
```

die Methode `singe` zu überschreiben, erhalten wir beim Compilieren

Konsole

```
Kind.java:2: singe() in Kind cannot override singe() in Papa;
  overridden method is final
    public void singe() {
                ^
```

als Fehlermeldung vom Compiler. Auf einige Beispiele solcher finalen Methoden kommen wir in Kapitel 18 noch zu sprechen. Sie finden sich in der Klasse `Object`, auf die wir im folgenden Abschnitt eingehen. Durch die **`final`**-Deklaration dieser Methoden ist sichergestellt, dass bei diesen Methoden alle Java-Objekte das gleiche Verhalten aufweisen. Außerdem ist der entsprechende compilierte Programmcode effizienter, weil kein dynamisches Binden mehr durchgeführt werden muss.