Errata: Einfach Java (978-3-86490-852-1)

Stand: 31. Juli 2022

S. 122 Fehler im Cast und bei Variablennamen in der `equals()`-Methode

```
Car other = (Car) obj;
```

=>

```
Car otherCar = (Car) other;
```

Die Methode muss korrekt wie folgt aussehen:

```
@Override
public boolean equals(final Object other)
{
    if (other == null)                          //   Null-Akzeptanz
        return false;
    if (this == other)                          //   Reflexivität
        return true;
    if (this.getClass() != other.getClass())  // Typgleichheit
        return false;

    Car otherCar = (Car) other;
    return Objects.equals(brand, otherCar.brand) &&
           Objects.equals(color, otherCar.color) &&
           horsePower == otherCar.horsePower;
}
```

S. 220 Tippfehler im Listing: `o` statt `obj` bei `default`

```
static String formatterJava17PatternSwitch(Object obj) {
    return switch (obj) {
        case Integer i -> String.format("int %d", i);
        case Long l    -> String.format("long %d", l);
        case Double d  -> String.format("double %f", d);
        case String s  -> String.format("String %s", s);
        default        -> o.toString();
    };
}
```

=>

```
static String formatterJava17PatternSwitch(Object obj) {
    return switch (obj) {
        case Integer i -> String.format("int %d", i);
        case Long l    -> String.format("long %d", l);
        case Double d  -> String.format("double %f", d);
        case String s  -> String.format("String %s", s);
        default        -> obj.toString();
    };
}
```

S. 232 Tippfehler im Listing unten

Die Methode beisst `isPalindromeRec()`, wird aber fälschkicherweise per `isPalindromeRecV2()` aufgerufen:

```
isPalindromeRecV2

=>

isPalindromeRec
```

S. 255 Fehlerhafter Satz

In den nachfolgenden Auflistungen sind die Namen von Short-circuiting Operations jeweils kursiv dargestellt.

- ⇨ **Streichen, der gesamte Satz ist zu viel**

S. 319 Logikfehler: Programm verwendet statischen Import, der nicht erklärt wurde

Im Kapitel 12.2.2 Formatierung und Parsing (wird u.a. die `'ofPattern'` Methode verwendet

```
DateTimeFormater ddMMMMyyFormat = ofPattern("dd.MM.yyyy")
```

Der direkte Zugriff auf den Namen `ofPattern` könnte verwirren und man könnte annehmen, dass vor dem `ofPattern` noch `'DateTimeFormater'` fehlt:

```
DateTimeFormater ddMMMMyyFormat =
        DateTimeFormater.ofPattern("dd.MM.yyyy")
```

- ⇨ Das ist nicht nötig, da ein statischer Import fehlt, der leider nicht erklärt wurde und mit dem Funktionalität eingebunden kann, so dass man den definierenden Klassennamen weglassen kann

S. 365 Fehlerhafter Satz

Während es im Kontext von Fallunterscheidungen praktisch sein kann, ist das Redefinieren aber ungünstig, wenn die Semantik wechselt:

- ⇨ **Ersten Teilsatz streichen, der ist zu viel**

Das Redefinieren einer Variablen ist vor allem dann ungünstig, wenn die Semantik wechselt: