

# Arbeitsprobe für SW-Entwickler

## - Getränkeautomat -

---

### Arbeitsauftrag

Entwickeln Sie die zentrale Geschäftslogik eines Getränkeautomaten mit Wechselgeldfunktion. Der Automat soll unterschiedliche Fächer mit jeweils eigenem Preis für unterschiedliche Getränke besitzen. Stellen Sie die Logik über ein sprechend typisiertes Interface zur Verfügung, auf dessen Basis Sie die Funktionalität exemplarisch über einen automatisierten Test nachweisen. Die Hauptfunktion könnte z.B. wie folgt aussehen:

```
GetraenkUndWechselgeld kaufen(Getraenkewunsch auswahl, Muenze... einzahlung);
```

Die Funktion liefert entweder das gewünschte Getränk und ggf. das Wechselgeld in Form von Münzen unterschiedlichen Werts oder einen Fehler, wenn z.B. das gewünschte Getränk ausverkauft ist, der eingezahlte Betrag zu gering ist oder der Automat kein passendes Wechselgeld herausgeben kann. Der Automat soll folgende Geldstücke unterscheiden:

- 10 Cent, 20 Cent, 50 Cent
- 1 Euro (= 100 Cent), 2 Euro (= 200 Cent)

Beim Kauf eines Getränks zum Preis von 1,20 Euro und einer Einzahlung einer 2-Euro-Münze, gibt der Automat z.B. 80 Cent Wechselgeld in Form einer 50-, einer 20- und einer 10-Cent-Münze zurück. Der Waren- und Geldbestand des Automaten ist begrenzt, und das Interface sollte über Funktionen zum Befüllen und Entleeren des Automaten verfügen.

Berücksichtigen Sie in ihrem Entwurf nur so viel Flexibilität, wie diese augenscheinlich sinnvoll ist, z.B. unterschiedliche Preise und Waren für Automaten an unterschiedlichen Standorten, aber z.B. nicht unterschiedliche Münzwerte. Die Übung verlangt weder die Bereitstellung einer Oberfläche noch eine Persistierung des Automatenzustands in einer Datenbank. Eine transiente Lösung ist ausreichend.

### Der Weg ist das Ziel

Ziel ist der Entwurf einer eleganten, leicht verständlichen Lösung, die sich – soweit möglich – natürlicher Begrifflichkeiten bedient. Nutzen Sie dabei die Stilmittel der Objektorientierung im Allgemeinen und die Möglichkeiten von Java und des Standard-APIs im Speziellen aber vermeiden Sie Überfrachtung.

Die Vollständigkeit des Ergebnisses ist zweitrangig. Auch eine Teillösung ist willkommen, sofern diese in sich abgeschlossen und durch einen automatischen Test abgesichert ist. Für die Übung ist eine Dauer von mindestens 2 Stunden, höchstens 4 Stunden angesetzt.

### Arbeitsmittel

- Eclipse in aktueller Version
- JUnit für Unittests
- Apache-Commons Bibliotheken, sofern hilfreich