

# Pflichtenheft / Arbeitsauftrag

Projekt: **SmartHomeSim – Gerätesteuerung und Automationen (Java OOP)**

## 1. Zielsetzung

Es soll eine Konsolenanwendung in Java entwickelt werden, die verschiedene Smart-Home-Geräte verwaltet, schaltet und deren gerätespezifische Funktionen ausführt. Der Fokus liegt auf objektorientierter Programmierung unter Einsatz von abstrakten Klassen, Interfaces und Polymorphie.

## 2. Projektumfang

Die Anwendung umfasst ein Gerätemodell mit Basisklassen und Spezialisierungen, eine zentrale Steuerungseinheit, einfache Automationen (Szenen), eine textbasierte Benutzerführung sowie grundlegende Fehlerbehandlung.

## 3. Fachliche Anforderungen

### 3.1 Geräteverwaltung

- Anlegen verschiedener Gerätetypen
- Verwaltung in einer gemeinsamen Liste
- Auswahl über eindeutige IDs
- Anzeige von Status- und Geräteinformationen

#### Mindestgerätetypen

- Light (Lampe)
- Thermostat
- SecurityCamera

### 3.2 Schaltfunktionen

Alle Geräte müssen über ein gemeinsames Interface schaltbar sein. Dies umfasst das Ein- und Ausschalten sowie das Abfragen des aktuellen Status.

### 3.3 Gerätespezifische Funktionen

**Light:** Helligkeit (0–100), optional Farbe

**Thermostat:** Zieltemperatur (10–30 °C), Temperatursimulation

**SecurityCamera:** Aufnahme starten/stoppen, optionale Bewegungserkennung

### 3.4 Szenen / Automationen

Es sind mindestens zwei Szenen umzusetzen, die mehrere Geräte gleichzeitig steuern, z. B. Abendmodus oder Abwesenheitsmodus.

## 4. Technische Anforderungen

#### Abstrakte Klasse Device

- Attribute: id, name, isOn
- Methoden: turnOn, turnOff, isOn, getInfo

- Abstrakte Methode: performFunction()

### **Interfaces**

- Switchable (Pflicht)
- Mindestens ein weiteres Interface (z. B. Configurable, Monitorable)

## **5. Bedienkonzept**

Die Anwendung wird über ein Konsolenmenü bedient. Ein Hauptmenü erlaubt Geräteverwaltung und Szenenauswahl. Ein Untermenü ermöglicht gerätespezifische Aktionen.

## **6. Validierung und Fehlerbehandlung**

Ungültige Eingaben dürfen nicht zum Programmabbruch führen. Wertebereiche sind zu prüfen und sinnvoll zu melden.

## **7. Abnahmekriterien**

Das Projekt gilt als abgeschlossen, wenn mindestens drei Gerätetypen implementiert sind, Interfaces und abstrakte Klassen korrekt eingesetzt werden, Szenen funktionieren und die Anwendung stabil bedienbar ist.

## **8. Optionale Erweiterungen**

Persistenz (Datei), Timer, Ereignissteuerung, Logging oder Unit-Tests.