

BLueJ-Design-Pattern

Bearbeitet in Gruppen die Aufgaben zu diesen Pattern. Hilfestellungen und Codebeispiele sind angegeben. Bei Fragen fragen!

Gruppenaufteilung

| Gruppe | Thema | Schüler | Projekt |
|--------|---------------------|---------------------------|----------|
| G1 | Singleton, Komposit | Lukas S., Abu, Yigit | Basis-G1 |
| G2 | Iterator | Richards, Lukas Y., Robin | Basis-G2 |
| G3 | Observer | Sahib, Gracjan, Danilo | Basis-G3 |
| G4 | Factory | Cedric, Zoraiz, Lars | Basis-G4 |

Aufgaben

G1

Findet in dem Raumplaner Projekt ein Beispiel für ein Singleton und ein Komposit.

Erklärt diese (nehmt Infos aus Text "Entwurfsmuster") und **Implementiert** je ein weiteres Beispiel. Nehmt als Beispiel für ein Singleton zum Beispiel eine Alarmanlage.

G2

Entwickelt einen Iterator für eine ArrayList.

Erklärt kurz, wie dieser funktioniert.

Nutzt hierzu die Klasse `ArrayListIterator`, die im Projekt zu finden ist. Implementiert die folgenden Methoden:

- `start()` - Springe an den Anfang (index 0)
- `next()` - Nächstes Element
- `hasNext()` - Sind wir beim Letzten Element
- `getCur()` - Aktuelles Element holen

Nutzt die Main-Methode in der `Main` Klasse zum Testen, aber lasst sie unverändert!

G3

Implementiert das Observer-Pattern am Beispiel einer Wetterstation. Als Subjekt soll eine Messstation Werte messen und diese sollen auf verschiedenen Anzeigen (Observern) wie zum Beispiel einer Temperaturanzeige oder einer Windgeschwindigkeitsanzeige angezeigt werden. Die Wetterdaten werde in einem Objekt der Klasse `WeatherData` beschrieben. Bearbeitet hierzu die Klassen

SensorStation (Subjekt) und **Display** (Observer). Die verschiedenen Anzeigen erben von der Klasse **Observer**, müssen aber nicht angepasst werden.

Wenn Ihr schon fertig seid: **Recherchiert** wie *Interfaces* in Java funktionieren und **erklärt** diese.

G4

Entwickle eine Fabrik für Möbel, die verschiedene Arten von Möbeln wie Stühle, Tische und Schränke erzeugen kann.

Erklärt eure Umsetzung sowie die Vor- und Nachteile des Factory-Pattern.

Denkt euch weitere Mögliche Fabrik-Methoden bzw. Anwendungsfälle für das Factory-Pattern **aus** und **stellt** diese **vor**.