

State Pattern – Verhaltensmuster

Yassine Bensaleh, Andrea Engel,
Rene Fischer, Sascha Görnert,
Niko Lockenvitz, Julian Rolle

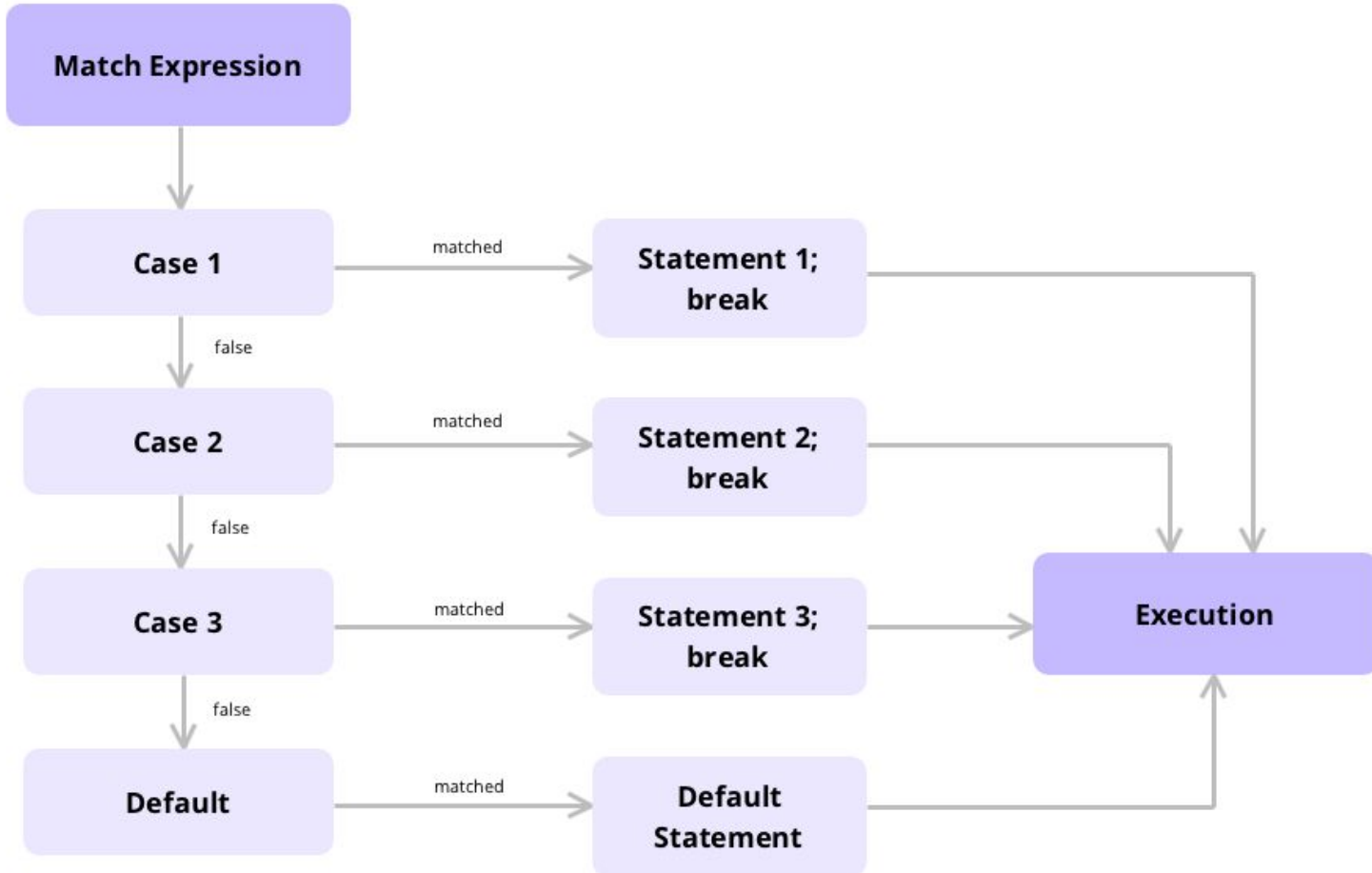
Mannheim, den 23.01.2020



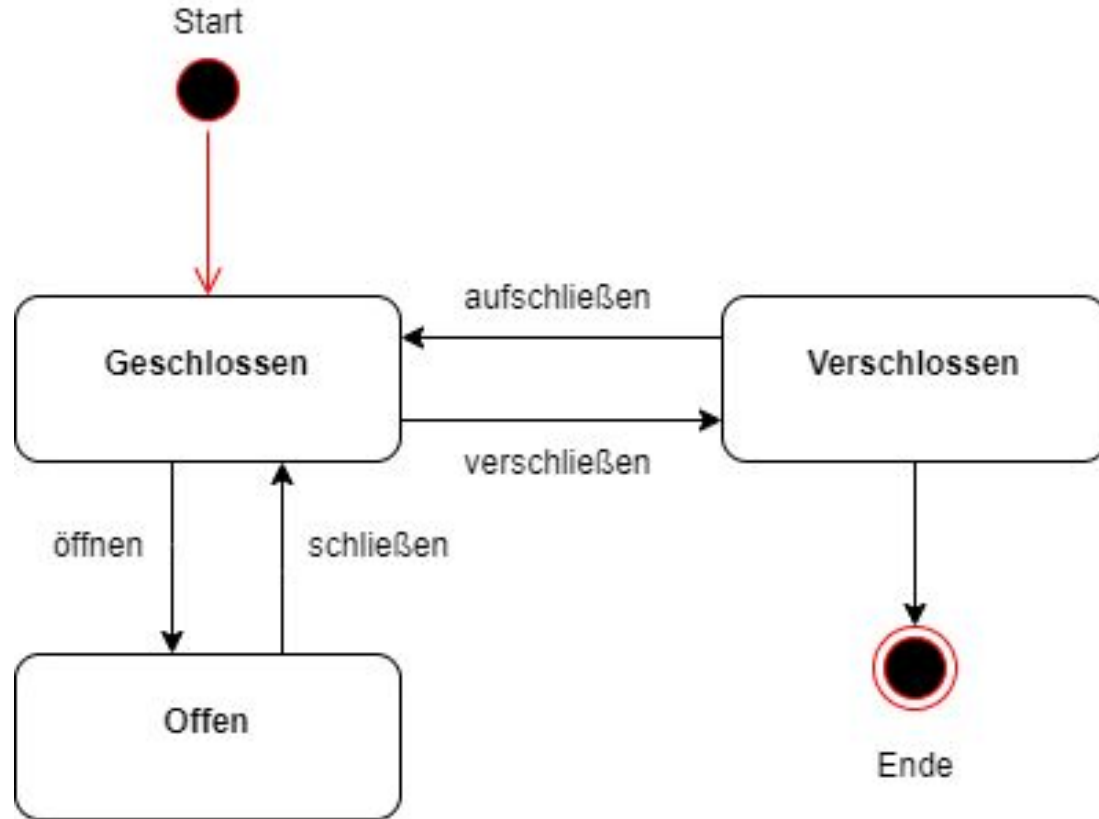
DHBW

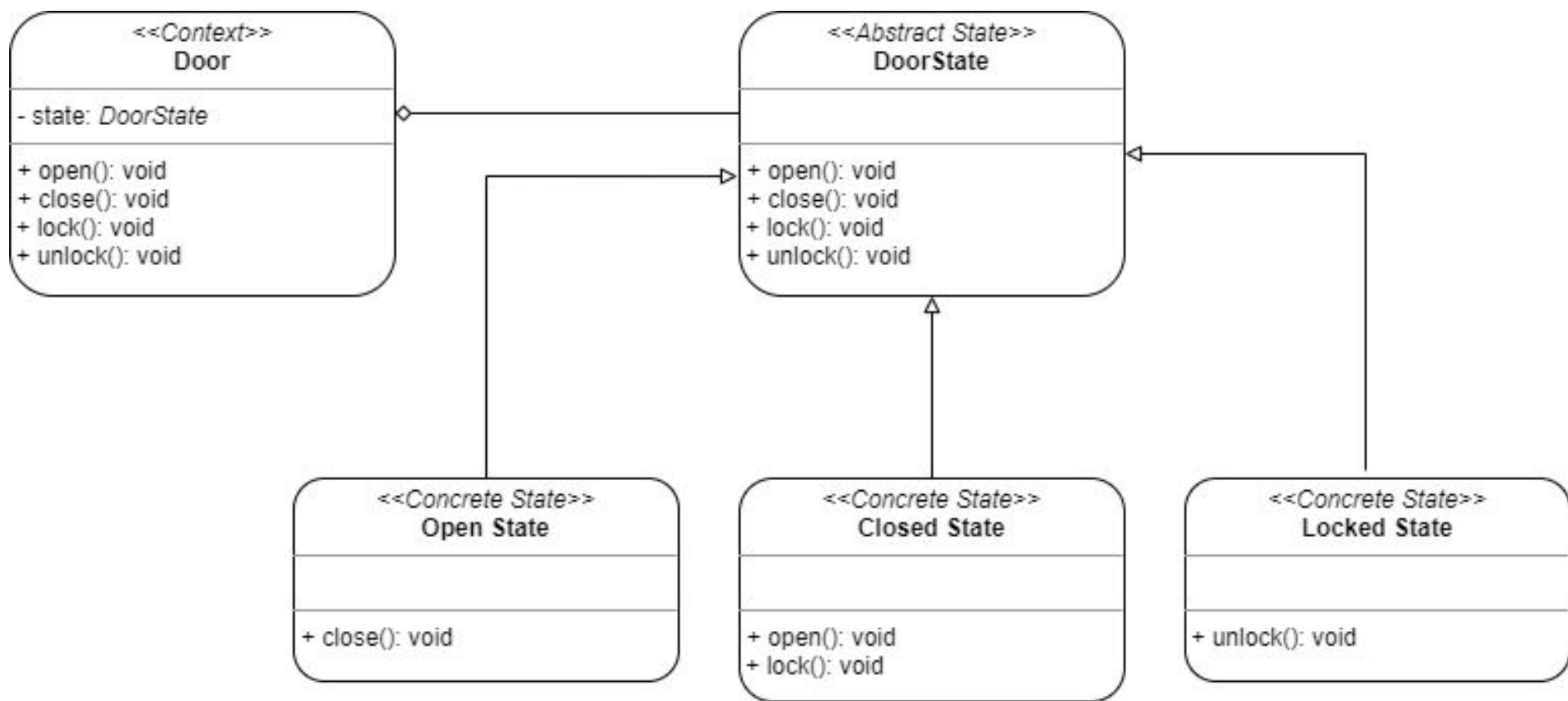
Duale Hochschule
Baden-Württemberg

Verwendung: Objektverhalten



Zustände: Tür (Smart Home)





Codebeispiel

Vorteile

- Erweiterbarkeit
 - neues Objekt für neuen Zustand
- bessere Verständlichkeit und Übersichtlichkeit
 - Verantwortlichkeiten werden delegiert
 - Verhalten einer Klasse wird auf verschiedene Zustandsobjekte verteilt
- geringere Fehleranfälligkeit
 - Fehler in Zustandsobjekten, nicht in einzelner Klasse

Nachteile

- erhöhte Klassenanzahl
- zu hoher Implementierungsaufwand für weniger komplexe Klassen
- Probleme bei zu vielen verschiedenen Zustandsübergängen
 - jeder Zustandsübergang muss in vorangegangener Klasse implementiert werden
 - Ausnahmen stellen Problem dar, für jede Ausnahme Übergang notwendig

Quellen

- [https://de.wikipedia.org/wiki/Zustand_\(Entwurfsmuster\)](https://de.wikipedia.org/wiki/Zustand_(Entwurfsmuster))
- <https://refactoring.guru/design-patterns/state>
- https://www.tutorialspoint.com/python_design_patterns/python_design_patterns_state.htm