

SOLID / GRASP / **SINGLETON PATTERN**

WEITER WISSEN →



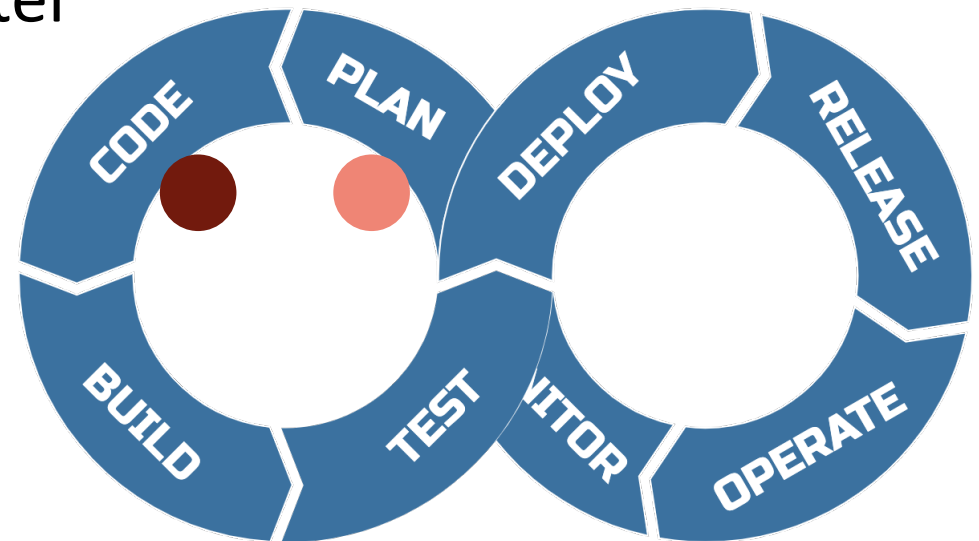
Ziel

Nach der Lektion können die Studierenden die SOLID und GRASP Prinzipien gegenüberstellen und diese kombinieren.

Nach der Lektionen haben die Studierenden analysiert und begründet, welche Entwurfsmuster sie für die Modularbeit benötigen und einsetzen werden.

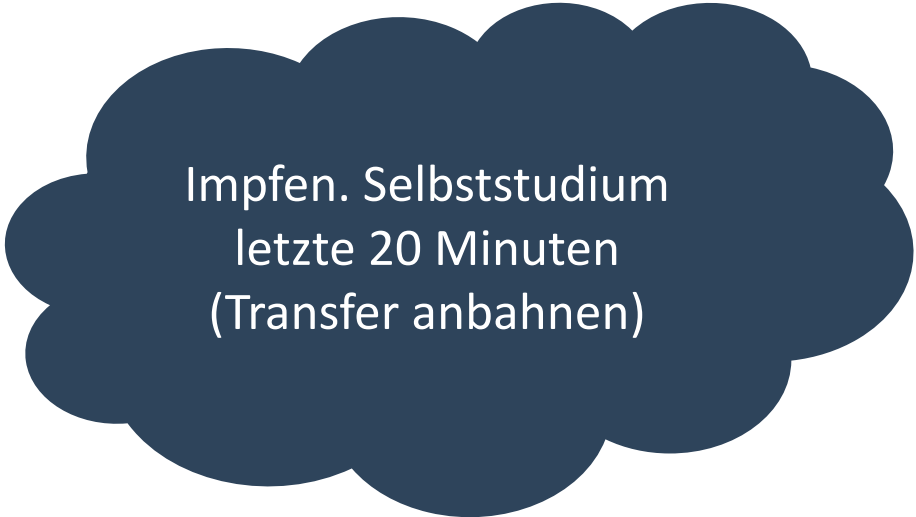
Singleton Pattern

- Sinn und Zweck von Mustern
 - Strategy Pattern (sehr SOLID)
 - Automatisierte Tests
-
- GRASP interpretieren
 - Lernen, wo und wie Entwurfsmuster nachgeschlagen werden können



Agenda

- Ziel
- Singleton Pattern
 - Relation zu GRASP/SOLID
- Zielkontrolle





Impfen. Selbststudium
letzte 20 Minuten
(Transfer anbahnen)

Das Pattern

<https://refactoring.guru/design-patterns/singleton>

– Wir wollen von einer Klasse genau **EINS**

 Use the Singleton pattern when a class in your program should have just a single instance available to all clients; for example, a single database object shared by different parts of the program.

 Use the Singleton pattern when you need stricter control over global variables.

Wie SOLIDE ist das Singleton Pattern?

- S (Single Responsibility)
 - Ich denke der Name ist Antwort genug 🤔
- O (Open-Closed)
 - 🩸 Verletzt, das Singleton Pattern ist in sich geschlossen und kaum erweiterbar
- L (Liskov Substitution)
 - 🩸 Verletzt, direkt vom Singleton Abhängig
- I (Interface Segregation)
 - 🩸 Verletzt, direkt vom Singleton Abhängig
- D (Dependency Inversion)
 - 🩸 Verletzt, direkt vom Singleton Abhängig
- Zusammenfassung: Nützlich, aber nicht SOLID (dafür GRASP).
 - Creator and Expert
 - Könnte mit Factory SOLID-er ersetzt werden
 - Es ist trotzdem enorm wichtig, weil wir den Fall oft antreffen, genau EINE Instanz zu wollen

GRASP (General Responsibility Assignment Software Patterns)

- Responsibility can be:
 - accomplished by a single object.
 - or a group of object collaboratively accomplish a responsibility.
- GRASP helps us in deciding which responsibility should be assigned to which object/class.
- Identify the objects and responsibilities from the problem domain, and also identify how objects interact with each other.
- Define blue print for those objects – i.e. class with methods implementing those responsibilities.



Ein Werkzeug dafür
sind CRC Karten

Design Patterns analysieren

Schauen Sie folgendes Video

<https://www.youtube.com/watch?v=FLmBql3IKMA>

Analysieren Sie min. 3 weitere Pattern:

<https://refactoring.guru/design-patterns/catalog>

Suchen Sie Ihre:m MA Partner:in auf und teilen Sie sich auf! Je 3 Patterns.

Notieren Sie sich die Pattern, welche für Ihre Modularbeit interessant oder hilfreich sein werden/könnten.

Für andere Sprachen, googeln Sie: <Sprache> <Pattern Name> 😊

In der Lernkontrolle ist ein Feld in dem Sie die Patterns nennen können!

Feedback

AUFTRAG

 5min

<https://forms.office.com/r/5Bzv0rY9Yq>



ABB Login





WEITER WISSEN.

Wir begleiten Sie!