

---

## Softwaretechnologie-Projekt (Prototyp)

### Aufgabenblatt 2

---

#### Vorbereitungsaufgaben

##### 1. Design-Patterns

Welche Design-Pattern kennt ihr? Wie können diese beispielhaft in C++ umgesetzt werden? Informiert euch im speziellen zu folgenden Design-Pattern:

- Singleton:
  - Welche Möglichkeiten der Implementierung dieses Design-Patterns gibt es in C++?
  - Welche Vor- und Nachteile haben diese im Besonderen auf die Lebensdauer des Objektes?
- Publish/Subscribe:
  - Welche Möglichkeiten der Implementierung im Hinblick auf die benötigten Klassen gibt es in C++?
- Factory

##### 2. Anti-Patterns

Informiert euch über *Anti-Patterns*. Was ist ihre Bedeutung und wie können sie verhindert werden? Was besagt das *Golden Hammer*-Pattern?

## Übungsaufgaben

In dieser Übung wollen wir die besprochenen Design-Pattern zur Strukturierung unseres Codes anwenden.

### 1. Das Singleton Design-Pattern

Um auf globale Parameter eines Programms an verschiedenen Stellen im Quellcode zugreifen zu können, bietet sich das Singleton-Pattern an. Implementiert eine Klasse *Settings* als Singleton. Standardwerte der Parameter sollen im Konstruktor der Klasse zugewiesen werden und später durch *set* und *get* Funktionen manipuliert und ausgelesen werden können.

### 2. Das Publish/Subscribe Design-Pattern

Implementiert das *Publish/Subscribe* Design-Pattern. Definiert dazu die Klassen des *Topic*, *Publishers* und *Subscribers*. Nutzt eine weitere Klasse *Broker*, die als Singleton implementiert werden soll, als Vermittler zwischen *Publisher* und *Subscriber*.

### 3. Das Factory Design-Pattern

Implementiert Factory-Methoden zur Generierung von *Publisher* und *Subscriber* Objekten.