JUST DO IT
HÖHERE ABTEILUNG FÜR
INFORMATIONSTECHNOLOGIE

tgm
Die Schule der Technik

A06: Design Patterns

Softwareentwicklung
5BHIT 2017/18

Maximilian Müller

Version 1.0
Note:
Betreuer:
Begonnen am 16. November 2017
Beendet am 28. November 2017

Inhaltsverzeichnis

1 Aufgabenstellung

Implementiere die folgende Aufgabenstellung in einer Programmiersprache deiner Wahl!

1.1 Funktionalität (20 P.)

Verwende das Decorator Pattern, um die Socket-Kommunikation zwischen einem einfachen Server und einem simplen Client (Konsole genügt) zu dekorieren! Die Plaintext-Kommunikation kann z.B. mit einer RSA- oder AES-Verschlüsselung, einer BASE64-Codierung, einem Hashwert / Fingerprint dekoriert werden. Eine implementierte Verschlüsselungsart reicht.

1.2 Dokumentation (20 P.)

Dokumentiere den Code ausführlich (Sphinx / JavaDoc ist nicht erforderlich).

1.3 Protokoll (30 P.)

Schreibe ein sauberes (Kopf- und Fußzeile, ...) Protokoll, welches Folgendes beinhaltet:

- UML-Klassendiagramm der verwendeten Architektur inkl. Beschreibung
  - Kurze allgemeine Ausarbeitung zu Design Patterns
  - Wie können Design Patterns unterteilt werden
  - Wozu Design Patterns
  - Übersicht existierender Design Patterns
- Ausarbeitung zum Decorator Pattern
  - Allgemeines Klassendiagramm
  - Grundzüge des Design Patterns (wichtige Operationen etc.) am Beispiel des implementierten Programms inkl. spezielles Klassendiagramm
  - Vor- und Nachteile
  - (Weitere) Anwendungsfälle
- Ausarbeitung zu einem der folgenden Design Patterns: Observer, Abstract Factory, Strategy
  - Allgemeines Klassendiagramm
  - Grundzüge des Design Patterns (wichtige Operationen etc.) mit einem kurzen eigenen Beispiel inkl. spezielles Klassendiagramm
  - Vor- und Nachteile
  - (Weitere) Anwendungsfälle

1.4 Erweiterung (30 P.)

Implementiere eine zweite Verschlüsselungsart und zeige im Code, wie sie miteinander kombiniert werden können, um Nachrichten zu ver- und entschlüsseln!

2 Protokoll

2.1 UML