Marlene Dorfinger, Christian Maran

Design-Patterns in Python

Inhalt

[1. Aufgabenstellung 2](#_Toc406412285)

[2. Zeitaufzeichnung 2](#_Toc406412286)

[3. Patterns 2](#_Toc406412287)

[Singleton Pattern 2](#_Toc406412288)

[Observer pattern 2](#_Toc406412289)

[Decorator Pattern 2](#_Toc406412290)

[Factory Pattern 2](#_Toc406412291)

[Adapter Pattern 2](#_Toc406412292)

[4. Fehleranalyse 2](#_Toc406412293)

[Quellen 3](#_Toc406412294)

# Aufgabenstellung

In einem Team (2) soll die These verifiziert oder falsifiziert werden.

* Wählen Sie zumindest 5 verschiedenen Design-Pattern (zumindest 1 pro Kategorie) aus.
* Ausgehend von einem UML-Klassendiagramm wird jedes Design-Pattern auf Umsetzung in Python untersucht.
* Erstellen Sie für jedes Pattern Beispiel-Code (inkl. Sphinx-Dokumentation)
* Erstellen Sie ein genaues Protokoll allen Aufzeichnungen
* Geben Sie das Protokoll und den Beispielcode für alle gewählten Design-Pattern ab

Viel Erfolg!

Ressourcen:

http://www.aleax.it/gdd\_pydp.pdf  
http://python-3-patterns-idioms-test.readthedocs.org/en/latest/index.html  
http://legacy.python.org/workshops/1997-10/proceedings/savikko.html  
http://www.vincehuston.org/dp/

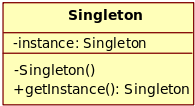
head first: design pattern

# Zeitaufzeichnung

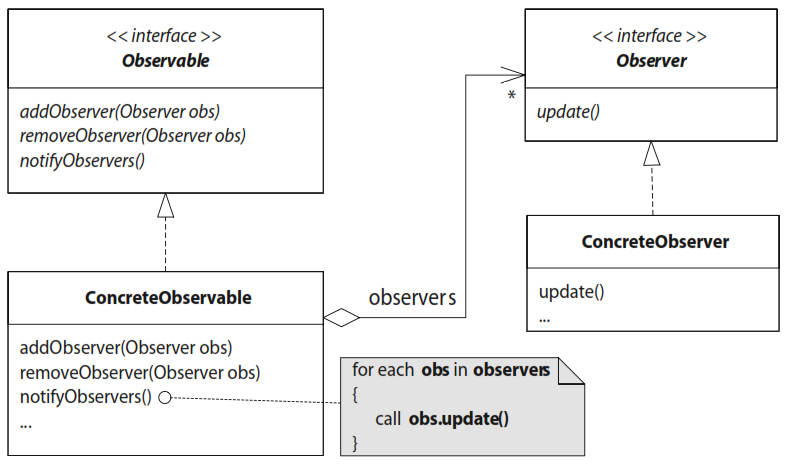
|  |  |  |  |
| --- | --- | --- | --- |
| Aufgabe | Erwartete Zeit | Tatsächliche Zeit | Zuständigkeit |
| Singleton Pattern | 30 min | 20 min | Dorfinger |
| Observer Pattern | 30 min |  | Dorfinger |
| Decorator Pattern | 20 min | 15 min | Maran |
| Factory Pattern | 20 min |  | Maran |
| Adapter Pattern | 20 min |  | Maran |
| Protokoll | 40 min |  | Dorfinger |
| Gesamt | 2 h 40 min |  |  |

# Patterns

## Singleton Pattern

UML:  


## Observer pattern

UML:  


## Decorator Pattern

## Factory Pattern

## Adapter Pattern

# Fehleranalyse

# Quellen