

AUFGABENBESCHREIBUNG LIVE-AUFGABE

OBJEKTTECHNOLOGIEN - WINTERSEMESTER 2017/18

1 Grundlegendes

Im Rahmen dieser Live-Aufgabe soll mit Hilfe des Decorator-Patterns ein (sehr einfacher) Video-Downloader mit Cache-Funktion erstellt werden.

2 Erstellen Sie folgende Klassen/Interfaces

2.1 DefaultVideoDownloader

Diese Klasse erlaubt es Videos herunter zu laden. In unserem Fall werden die Videos durch einfache Strings repräsentiert.

Dazu erhält sie folgende Methode:

2.1.1 *String downloadVideo(int videoid)*

Diese Methode liefert ein „Video“ in Form eines kurzen Strings zurück. Für jede unterschiedliche Video-Id wird ein entsprechend unterschiedlicher String zurück gegeben.

So wird z.B. für die *videoid* 7 folgender String zurück gegeben werden: „Video-Content für ID 7“.

3 Decorator-Pattern

3.1 Interface VideoDownloader

Erstellen Sie für die erzeugte Klasse und deren Methode ein Interface mit dem Namen *VideoDownloader*

3.2 VideoCache

Erstellen Sie eine Klasse *VideoCache* die als Decorator für *VideoDownloader* dient.

Dieser Decorator speichert für die jeweils letzte Anfrage die Video-ID sowie den Video-String. Wenn in direkter Folge eine Anfrage zur selben Video-ID erfolgt, soll der gespeicherte Video-String zurück geliefert werden.

Nochmals zur Verdeutlichung:

3.2.1 Fall 1: Kein oder falsches Video gespeichert

Dann wird das dekorierte Objekt nach dem Video gefragt. Dieses Video sowie die zugehörige Video-ID wird im Dekorator als Field gespeichert.

3.2.2 Fall 2: Selbe Video-ID wird ein zweites mal angefragt

Erfolgt eine Anfrage mit einer Video-ID, die der aktuell gespeicherten Video-ID entspricht, wird das gespeicherte Video direkt zurück geliefert. Das dekorierte Objekt wird dabei nicht angefragt.

3.3 Test / Main-Klasse

Erzeugen Sie ein Demo-Programm oder einen JUnit-Test, welcher die Funktionsfähigkeit und Korrektheit Ihrer Lösung demonstriert.