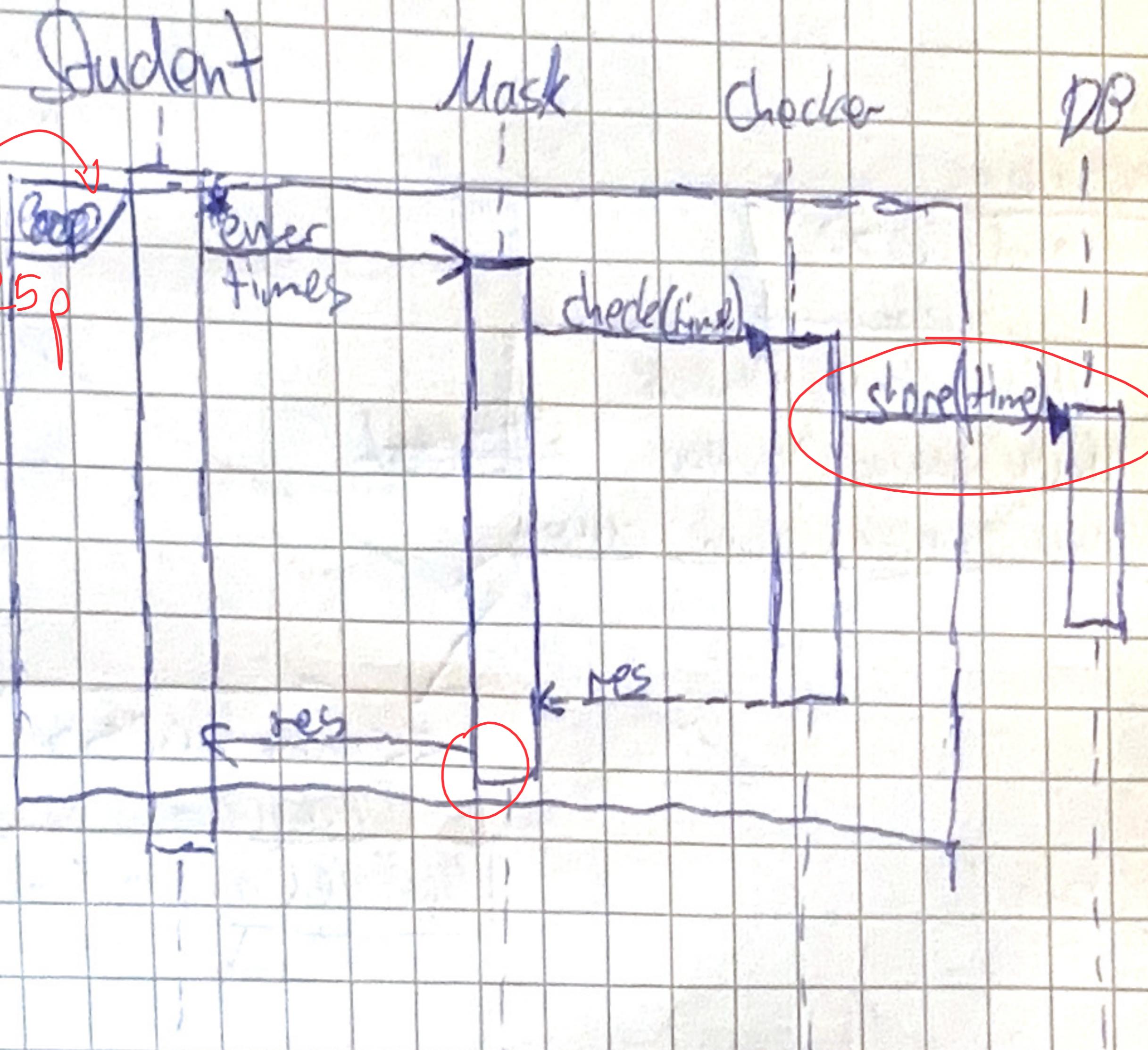


Mr. 1



* condition: conflict with times

↳ Bitte in ordentlicher Notation aufzuschreiben. 2.B.: [res != ok]
-0,25p

Mr. 2

a) context CourseManager :: addCourse(courseName: String)

pre: courseName != null and courses → includes(courseName)

context CourseManager :: addCourse(courseName: String)

post: courses → includes(courseName) and courses → size = courses @ pre → size + 1

context CourseManager

inv: courses → forAll(c | c != null) and courses → isUnique(c | c)

- 6) Aber die Methoden sind doch nur Add und Remove, was sich ja direkt auf die
 Srp: - Problem: Speicherung und Methoden in einer Klasse Speicherung
 bezieht.
 - Lösung: Speicherung des Arraylike im Separaten Ort → 25p

Dass die Methode "final" ist
 das Hauptproblem
 hinsichtlich der
 Erweiterbarkeit.

- OCP: - Problem: Sollten neue Anforderungen wie für z.B. Formate kommen,
 so müsste man bestehende code modifizieren
 - Lösung: Strategy pattern nutzen um Anforderungen hinzufügen ohne die
 Änderung der Klasse

- LSD: - Es entstehen keine Probleme da nicht mit Subtypen gearbeitet wird

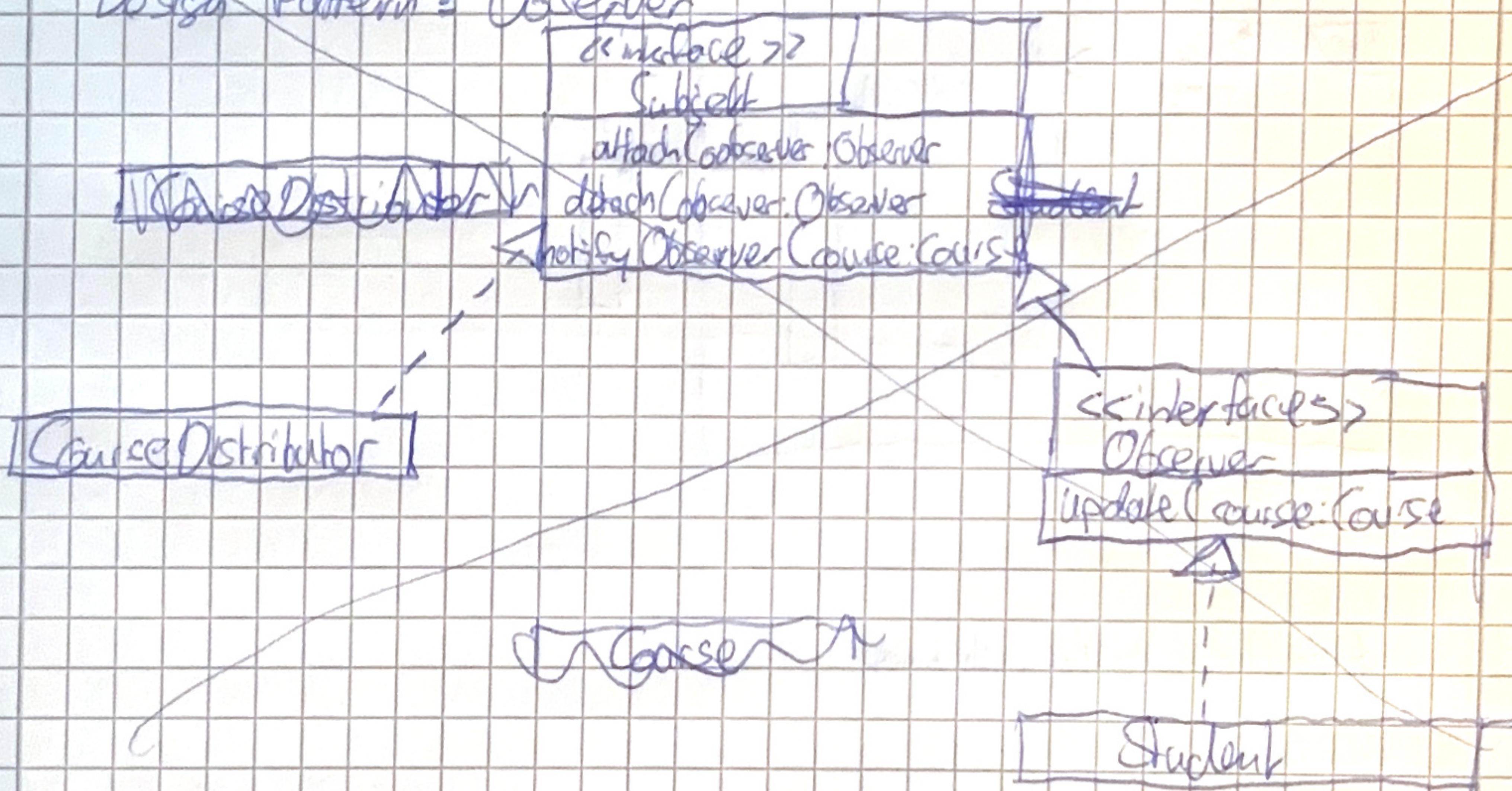
- ISP: - Es wird auch nicht mit Interfaces gearbeitet somit liegt kein
 Problem vor

- DIP: - Problem: CourseManager ist direkt abhängig von der konkreten Implementierung
 der Arraylike. Sollte man diese Speicherung ändern muss man
 Ort der

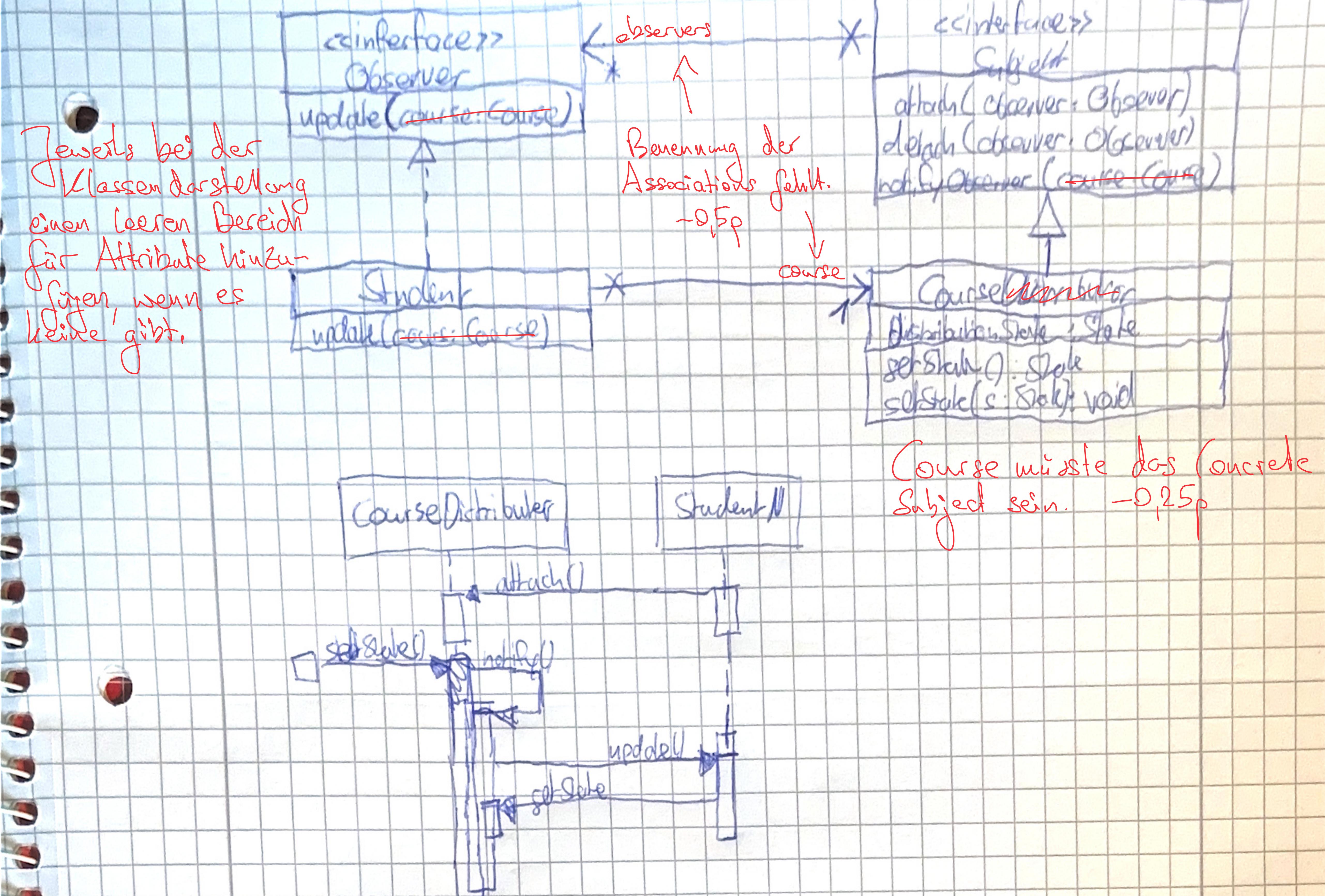
- Lösung: Lieber Interface benutzen ✓

Nr. 3

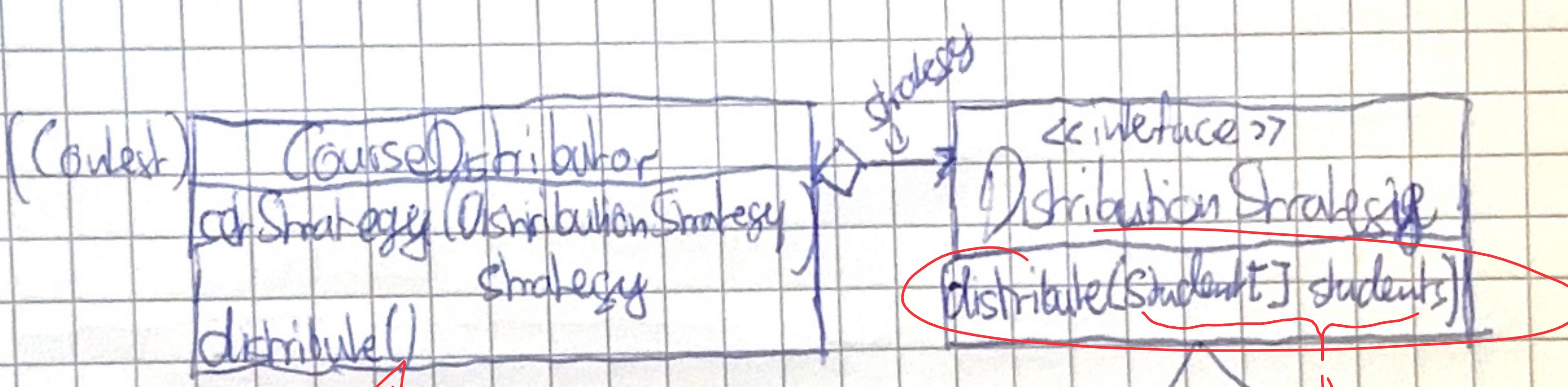
Design Pattern: Observer



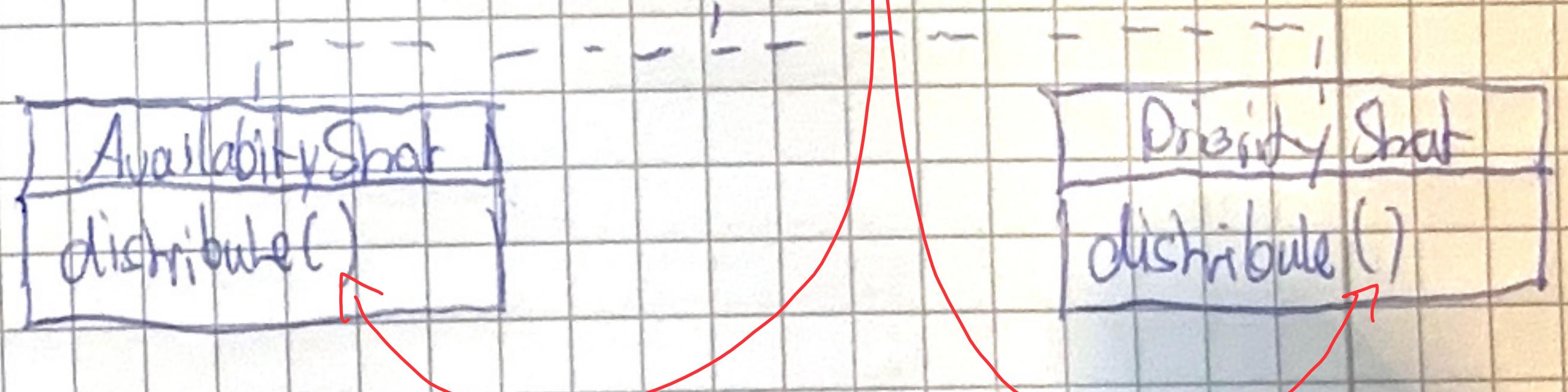
Design Pattern: Observer ✓



Nr. 4



Hier fehlen noch Inputparameter.



Wenn das Interface eine solche Signatur mit Eingabeparameter hat, müssen die os realisierenden Klassen ebenso diese Signatur ihrer Methode haben.
-0,25p