

# Einführung in Software Engineering



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

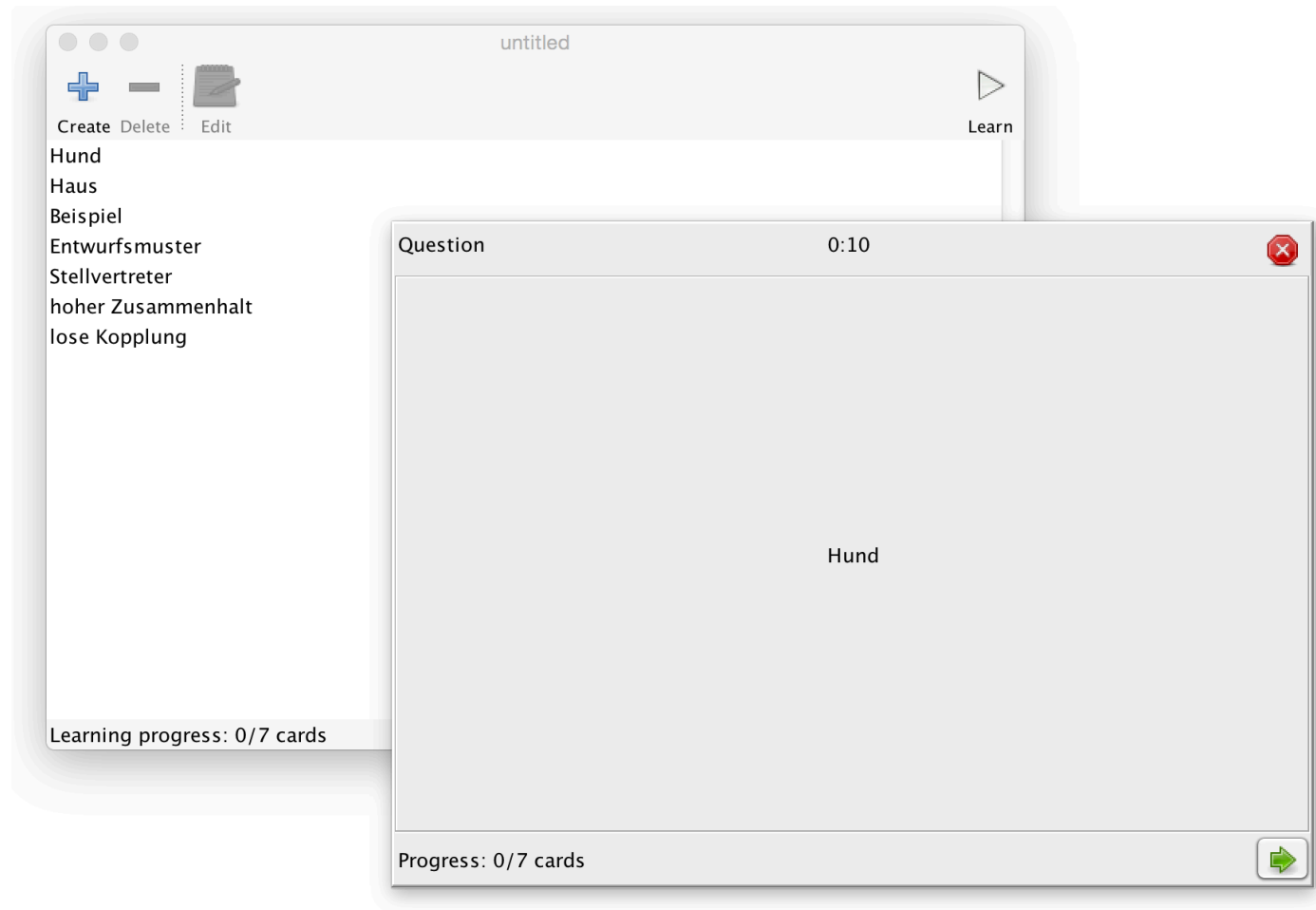
## 12. Übung - Besprechung

Dr.-Ing. Michael Eichberg  
**Sven Amann M.Sc.**

---

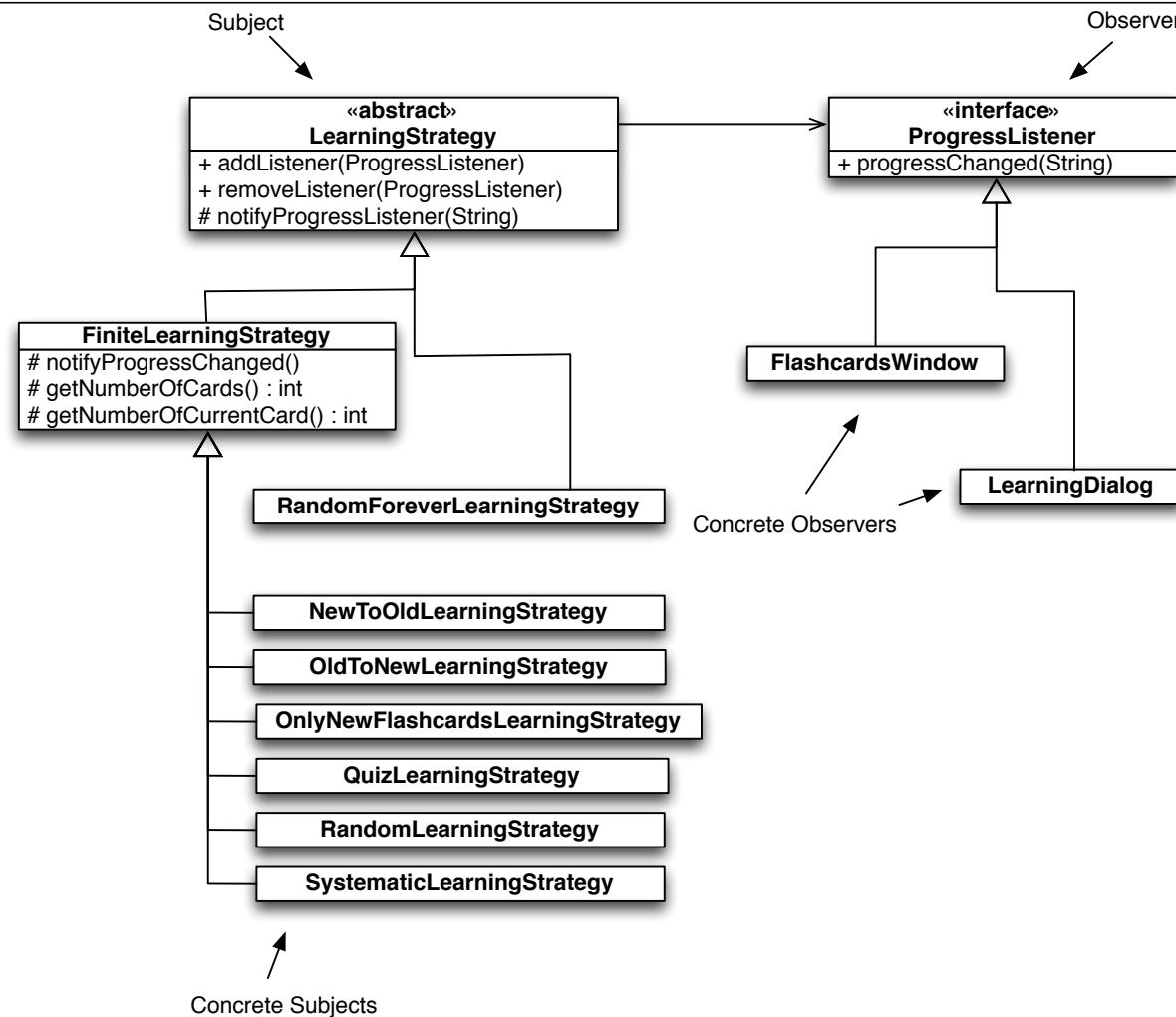
# Aufgabe 1

## Statusanzeige



# Aufgabe 1a)

## Entwurf anhand des Observer Patterns



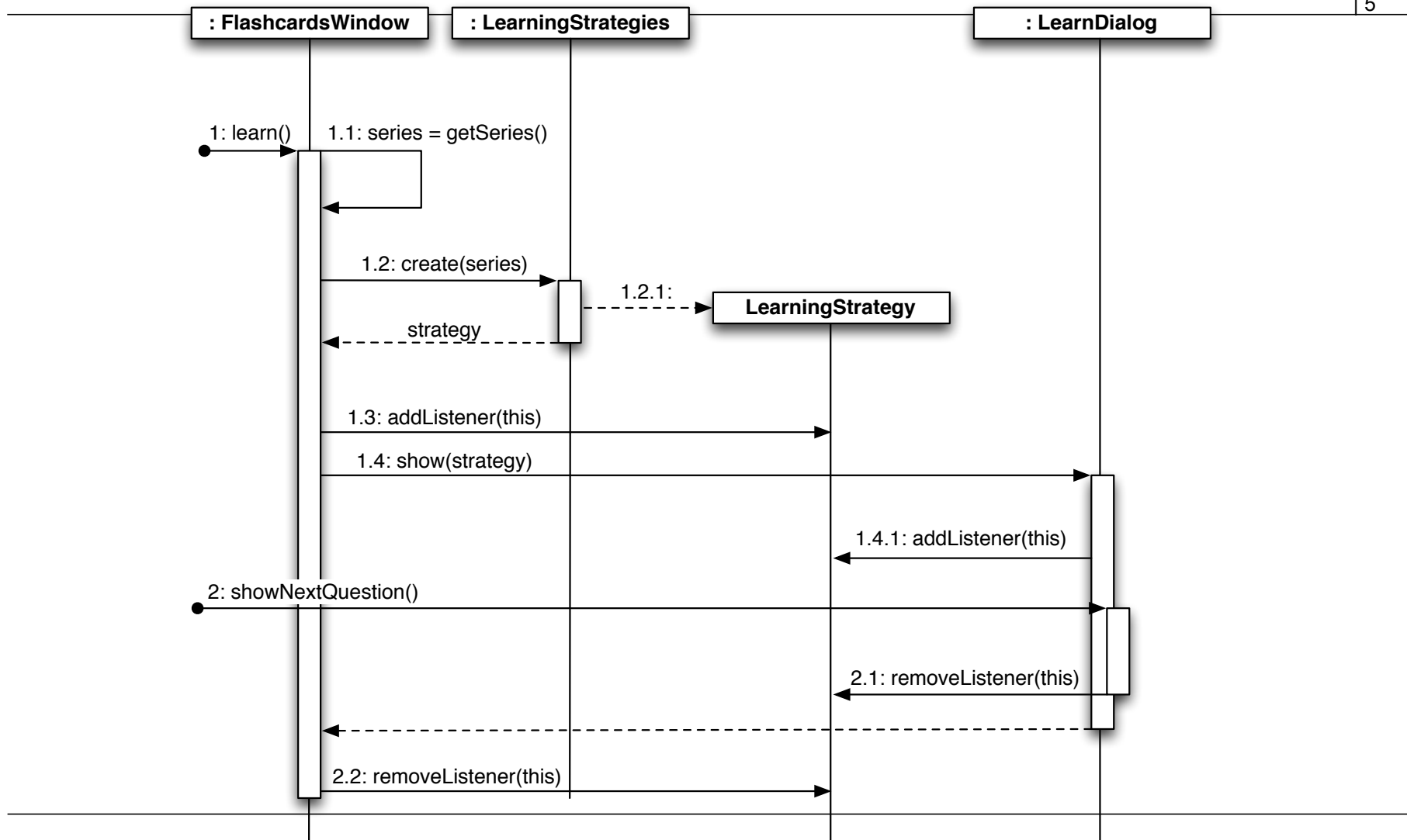
# Aufgabe 1a)

## Entwurf anhand des Observer Patterns

Im Pattern-Schema ist eine “subject”-Beziehung von den “concrete observers” auf die “concrete subjects” eingezeichnet. Dies ist der Fall, da in der Standardvariante der konkrete Observer eine explizite Referenz auf das konkrete Subjekt hält, um bei Änderungen Statusinformationen vom Subjekt abzufragen. In unserer Implementierung ist dies nicht der Fall, daher gibt es auch die entsprechenden Kanten nicht.

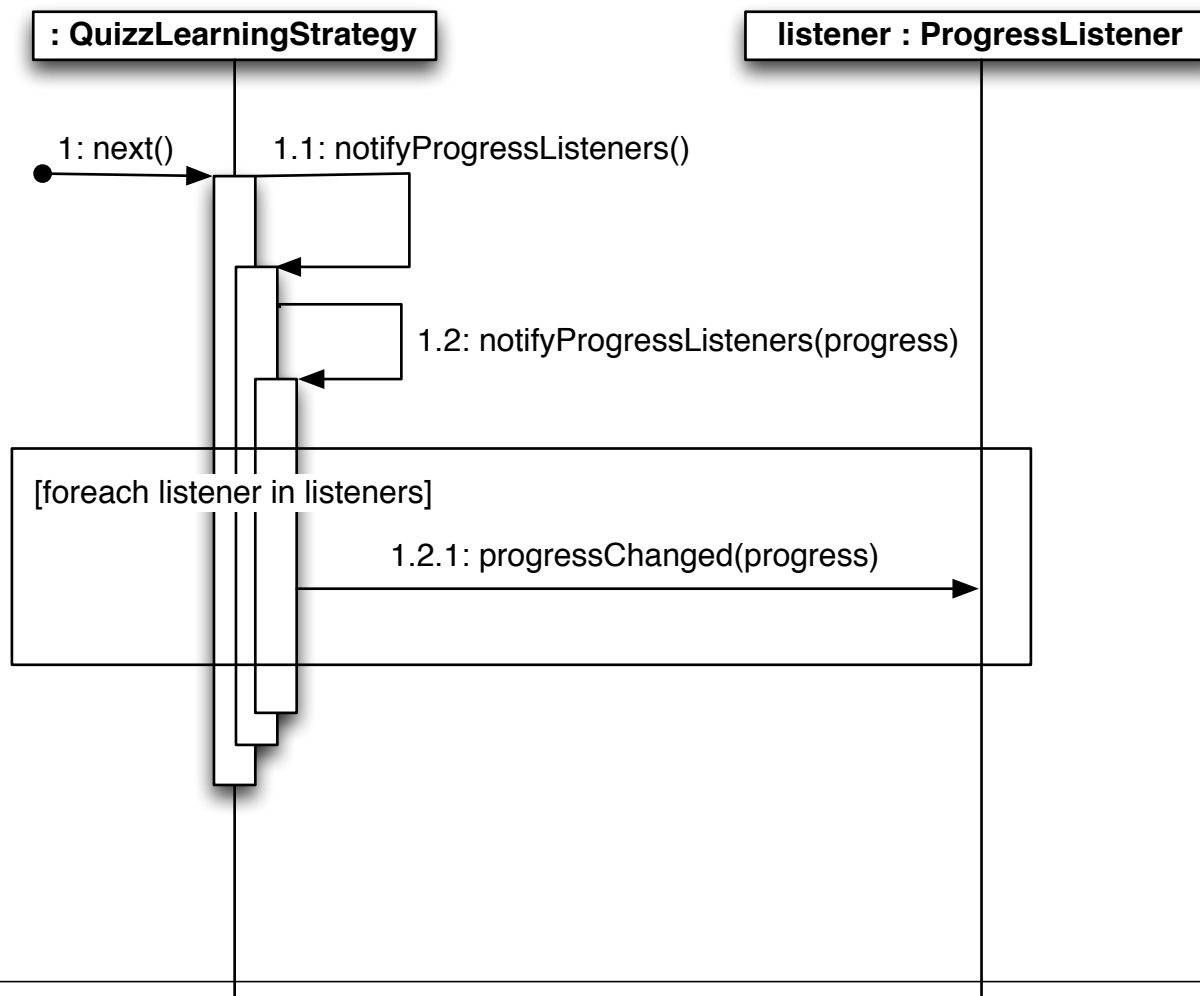
# Aufgabe 1b)

## Dokumentation der Registrierung



# Aufgabe 1c)

## Dokumentation der Kommunikation



# Aufgabe 1d)

## Tests für Subjects (QuizzLearningStrategy)

```
@Test
public void progressNotification() {
    ProgressTestHelper helper = new ProgressTestHelper();
    uut.addListener(helper);
    uut.next();
    helper.assertProgressIs("0/3 cards");
    uut.next();
    helper.assertProgressIs("1/3 cards");
}
```

# Aufgabe 1d)

## Tests für Subjects (Alle Strategien)

```
@Test
public void noProgressNotification() {
    ProgressTestHelper helper = new ProgressTestHelper();
    uut.addListener(helper);
    uut.removeListener(helper);
    uut.next();
    helper.assertProgressIs(null);
}
```