

# Software Engineering I



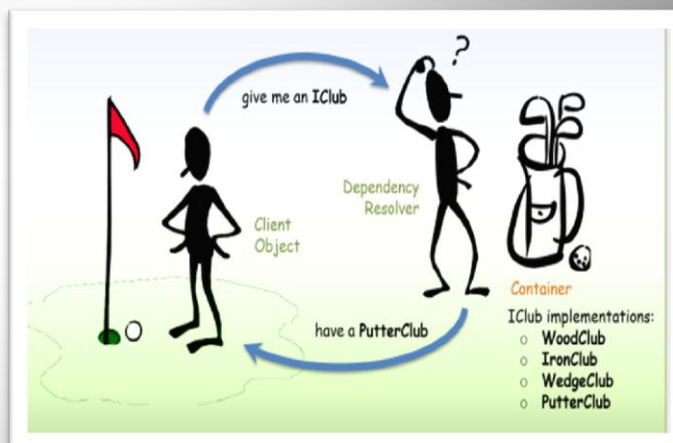
Patrick Steinhauer  
Jan Dennis Bartels

Professor  
Stefan Sarstedt

Assistent  
Norbert Kasperczyk-  
Borgmann

Aufgabenblatt Nummer 2

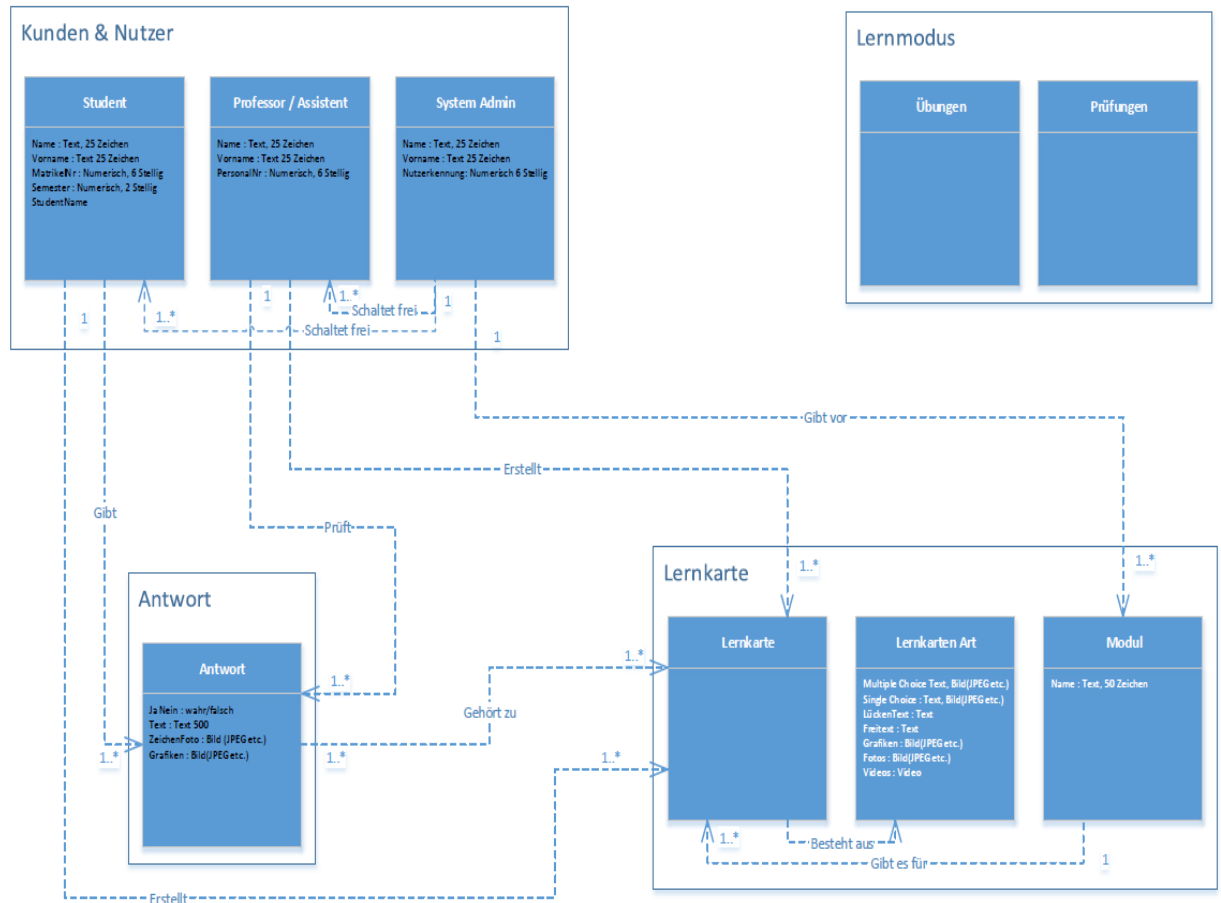
11.11.2014



## Inhaltsverzeichnis

UML-Diagramm .....	1
UML-Komponentendiagramm .....	2
Java Code Entität Lernkarte.....	3
Java Code Datentyp EmailDatentyp .....	4
Java Code Dependency Injection.....	6
NutzerverwaltungsKomponente .....	6
LernkartenVerwaltungsKomponente .....	7
AntwortAbgabeKomponente .....	8
Main Methode.....	9

# UML-Diagramm



Bei unserem UML-Diagramm haben wir den Schnitt so gewählt, dass

Es einmal eine LernkartenVerwaltungsKomponente gibt, eine AntwortAbgabeKomponente und eine NutzerVerwaltungsKomponente.

Hierbei haben wir uns gedacht, dass wir die Personen, die das System nutzen eine Komponente darstellen sollten, da die Kunden, User oder System Admins beispielsweise vieles gemeinsam haben. Alle sind Personen und haben bestimmte Attribute, die gleich sind.

Eine weitere Komponente ist die LernkartenVerwaltungsKomponente, die wiederum für die Lernkarten, Lernkartenart und das Modul zuständig ist.

Das Modul haben wir mit in die Komponente Lernkarte genommen, da es zu verschiedenen Modulen mehrere Lernkarten gibt.

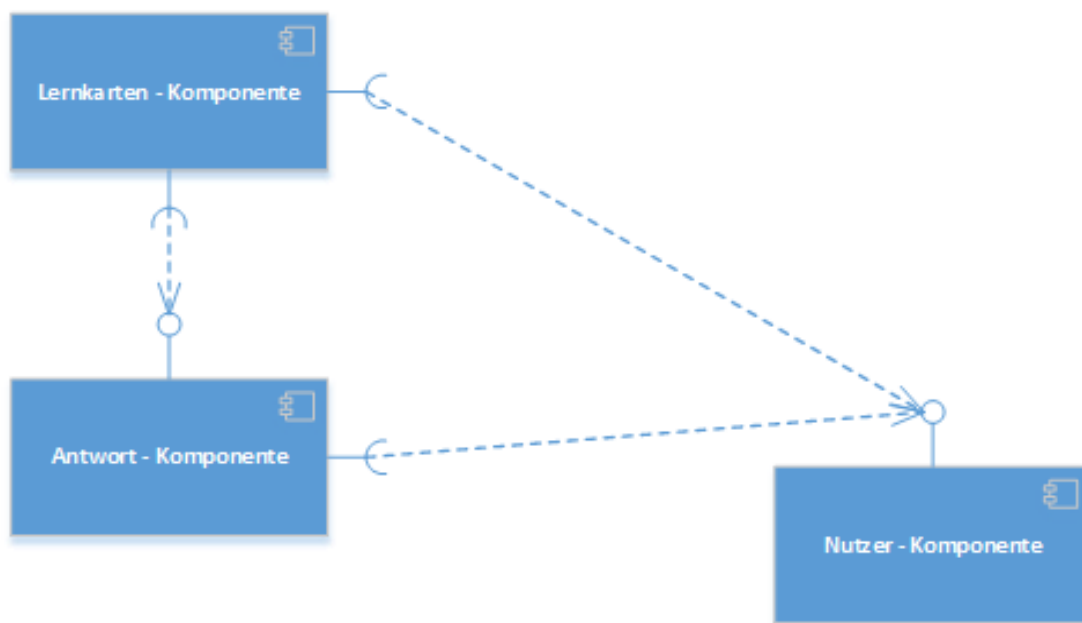
Die letzte Komponente, die wir modelliert haben ist die AntwortAbgabeKomponente, die für jegliche Antworten des Systems zuständig sein soll.

### Weitere Möglichkeiten:

In unserem Diagramm ist ebenfalls eine Komponente, die jedoch erst einmal noch alleine steht, da wir uns noch nicht einig waren, ob diese mit in unser Modell soll.

Weitere Überlegungen wären aus unserer Sicht, dass man die Antwort möglicherweise mit der Lernkartenverwaltungskomponente verbinden kann, da hier auch einige Attribute gleich sind (die Arten der Lernkarten). In unserem Modell ist es aus unserer Sicht jedoch sinnvoller dies extra zu machen, da Nutzer und Lernkarte beide jeweils mit der Antwort arbeiten werden/müssen.

## UML-Komponentendiagramm



## Java Code Entität Lernkarte

```
package lernkartenVerwaltungsKomponente;

import lernkartenArt.Lernkartenart;

/**
 * Created by patrick_steinhauer on 10.11.2014.
 */

public class Lernkarte {

    public Lernkarte() {
        this.lernkartenart = new Lernkartenart();
    }

    public Lernkarte(Lernkartenart art) {
        this.lernkartenart=art;
    }

    private Modul modul;

    private Lernkartenart lernkartenart;

    public Modul getModul() {
        return modul;
    }

    public void setModul(Modul modul) {
        this.modul = modul;
    }

    public Lernkartenart getLernkartenart() {
        return lernkartenart;
    }

    public void setLernkartenart(Lernkartenart lernkartenart) {
        this.lernkartenart = lernkartenart;
    }

}
```



Patrick Steinhauer  
Jan Dennis Bartels

```

public EmailDatentyp(String email) {
    Email = email;
}

public String getEmail() {
    return Email;
}

public static boolean isValidEmail(String inputMail) {
    Pattern ptr = Pattern.compile(EMAILREGEX);
    return ptr.matcher(inputMail).matches();
}

@Override
public boolean equals(Object o) {
    return Email.equals(o);
}
}

```

## Java Code Dependency Injection

### NutzerverwaltungsKomponente

```

package nutzerVerwaltungsKomponente;

/**
 * Created by patrick_steinbauer on 10.11.2014.
 */
public class NutzerVerwaltungsKomponente implements INutzerVerwaltung{

    public NutzerVerwaltungsKomponente() {

    }

}

```

Patrick Steinhauer  
Jan Dennis Bartels



## LernkartenVerwaltungsKomponente

```
package lernkartenVerwaltungsKomponente;

import antwortAbgabeKomponente.IAntwortVerwaltung;
import lernkartenArt.Lernkartenart;
import persistenz.IPersistenzServices;
import nutzerVerwaltungsKomponente.INutzerVerwaltung;
import java.util.Collection;

/**
 * Created by patrick_steinhauer on 10.11.2014.
 */

public class LernkartenVerwaltungsKomponente implements ILernkartenVerwaltung {

    private IAantwortVerwaltung antwortVerwaltung;

    private INutzerVerwaltung nutzerVerwaltung;

    private Collection<Lernkarte> lernkarte;

    public LernkartenVerwaltungsKomponente(IPersistenzServices persistenzServices,
    IAantwortVerwaltung antwortVerwaltung, INutzerVerwaltung nutzerVerwaltung) {

        this.antwortVerwaltung = antwortVerwaltung;

        this.nutzerVerwaltung = nutzerVerwaltung;

    }

    @Override

    public Lernkarte erstelleLernkarte() {

        return new Lernkarte();

    }

    @Override

    public Lernkarte aendereLernkarte(Lernkartenart neueArt) {

        return new Lernkarte(neueArt);

    }

}
```

Patrick Steinhauer  
Jan Dennis Bartels

```
@Override  
  
public Modul zeigeModuleVonLernkarteAn() {  
    return null;  
}  
}
```

### **AntwortAbgabeKomponente**

```
package antwortAbgabeKomponente;  
  
import nutzerVerwaltungsKomponente.INutzerVerwaltung;  
  
/**  
 * Created by patrick_steinhauer on 10.11.2014.  
 */  
  
public class AntwortAbgabeKomponente implements IAntwortVerwaltung{  
    private INutzerVerwaltung nutzerVerwaltung;  
    public AntwortAbgabeKomponente(INutzerVerwaltung nutzerVerwaltung) {  
        this.nutzerVerwaltung = nutzerVerwaltung;  
    }  
}
```

## Main Methode

```
package main;

import antwortAbgabeKomponente.AntwortAbgabeKomponente;

import antwortAbgabeKomponente.IAntwortVerwaltung;

import lernkartenVerwaltungsKomponente.ILernkartenVerwaltung;

import lernkartenVerwaltungsKomponente.LernkartenVerwaltungsKomponente;

import persistenz.IPersistenzServices;

import nutzerVerwaltungsKomponente.INutzerVerwaltung;

import nutzerVerwaltungsKomponente.NutzerVerwaltungsKomponente;

/**
 * Created by patrick_steinhauer on 11.11.2014.
 */

public class Main {

    public static void main(String[] args) {

        IPersistenzServices persistenzServices = null;

        INutzerVerwaltung nutzerVerwaltung = new NutzerVerwaltungsKomponente();

        IAantwortVerwaltung antwortVerwaltung = new AntwortAbgabeKomponente(nutzerVerwaltung);

        ILernkartenVerwaltung lernkartenVerwaltung = new
LernkartenVerwaltungsKomponente(persistenzServices, antwortVerwaltung, nutzerVerwaltung);

    }

}
```