



Technische Universität Dresden, 01062 Dresden



**Klausur  
Softwaretechnologie WS 2016/17**

Prof. Dr.rer.nat.habil.  
Uwe Aßmann

Name:	
Vorname:	
Immatrikulationsnummer:	

Aufgabe	Maximale Punktzahl	Erreichte Punktzahl
1	26	
2	64	
<b>Gesamt</b>	<b>90</b>	

**Hinweise:**

- In der Klausur ist als Hilfsmittel lediglich ein **A4-Blatt, beidseitig beschrieben**, zugelassen.
- Die Klammerung der Aufgabenblätter darf **nicht** entfernt werden.
- Tragen Sie bitte die Lösungen auf den Aufgabenblättern ein!
- Verwenden Sie keine roten, grünen Stifte oder Bleistifte!
- Es ist kein eigenes Papier zu verwenden! Bei Bedarf ist zusätzliches Papier bei der Aufsicht erhältlich. Bitte jedes zusätzliche Blatt mit Name, Vorname und Immatrikulationsnummer beschriften.
- Es sind alle Aufgabenblätter abzugeben!
- Ergänzen Sie das Deckblatt mit Name, Vorname und Immatrikulationsnummer!
- Halten Sie Ihren Studentenausweis und einen Lichtbildausweis zur Identitätsprüfung bereit.

# Anwendungsdomäne Fahrradverleih

---

## Aufgabe 1: Domänenmodell (26 Punkte)

Ergänzen Sie das folgende unvollständige Analyse-Klassendiagramm (Domänenmodell)!

Berücksichtigen Sie dabei folgende Hinweise:

- Denken Sie an Klassen, Enumerationen, Klassenbeziehungen, geeignete Rollennamen und Attribute! Methoden werden nicht modelliert.
- Berücksichtigen Sie dabei Assoziationsklassen, um die semantischen Beziehungen zwischen den Klassen korrekt zu beschreiben! Die gegebenen Klassen können auch zu Assoziationsklassen werden!

Alle domänenspezifischen Begriffe, die im obigen Text *kursiv* geschrieben sind, müssen im Modell wiederzufinden sein!

Es soll eine Software zum Management eines städtischen Fahrradverleihs geschaffen werden. Der Service dieses Fahrradverleihs besteht für einen registrierten Kunden darin, dass der Kunde an einem *Standort* ein *Fahrrad* ausleihen und an einem anderen *Standort* das *Fahrrad* zurückgeben kann. Jedes *Fahrrad* hat ein Kennzeichen (*kennzeichen*).

Ein *Standort* (beschrieben durch seine *Adresse*) ist entweder eine *Fahrradstation* oder ein *FreierStandort*. Eine *Fahrradstation* hat einen *Namen* und verfügt mindestens über einen *Fahrradstand*. Jeder *Fahrradstand* wird durch einen Code (*code*) identifiziert. Ein *FreierStandort* wird zusätzlich durch seine *GPSKoordinaten* identifiziert.

Für den Verleih verfügbare Fahrräder werden durch ihren Standort beschrieben. Dabei ist ein *Fahrrad* an einem freien Standort (*FreierStandort*) oder in einem *Fahrradstand* geparkt (*istGeparktAm FreierStandort* bzw. *istGeparktIm Fahrradstand*).

Für jeden Kunden wird ein *Kundenkonto* mit seinen Stammdaten (*name*, *adresse*, *email*, *telNummer*) angelegt. Zusätzlich muss sich der Kunde für einen *Tarif* entscheiden.

Der Fahrradverleih arbeitet mit verschiedenen *Partnerunternehmen* zusammen. Falls der Kunde des Fahrradverleihs gleichzeitig Kunde eines der *Partnerunternehmen* ist, kann ihm beim Fahrradverleih ein Rabatt (*rabatt*) gewährt werden. In diesem Fall müssen im *Kundenkonto* ein *Partnerunternehmen* und die Abbonnentennummer (*aboNummer*) bei diesem *Partnerunternehmen* gespeichert werden.

Die *Ausleihen* und die *Rückgaben* von Fahrrädern durch einen Kunden werden in seinem *Kundenkonto* registriert. Zur Bezahlung einer monatlichen *Rechnung* muss der Kunde eine *Kreditkarte* mit den üblichen Informationen (*Nummer*, *Gültigkeit*, *Kreditrahmen*) hinterlegen. Zusätzlich muss der *Kreditkartentyp* angegeben werden. Der Fahrradverleih akzeptiert dabei folgende *Kreditkartentypen*: *Mastercard*, *Visa*, *AmericanExpress*.

Der monatliche Rechnungspreis (*rechnungspreis*) ergibt sich aus der Summe der Preise für die einzelnen Nutzungen eines *Fahrrads* in einem Monat. Zur Berechnung des Preises einer einzelnen Nutzung eines *Fahrrads* wird das Datum (*datum*) und die Uhrzeit (*uhrzeit*) sowohl der *Ausleihe* als auch der *Rueckgabe* des *Fahrrads* gespeichert. Bei der *Ausleihe* des *Fahrrads* wird dem Kunden der Code für das Fahrradschloss (*schlossCode*) bereitgestellt. Die monatlichen *Rechnungen* eines Kunden werden in seinem *Kundenkonto* gespeichert.

Ausleihe
<i>attributes</i> datum uhrzeit schlosscode

Rueckgabe
<i>attributes</i> datum uhrzeit

Kundenkonto
<i>attributes</i> name adresse email telnummer

«enumeration» Tarif
<i>enumeration literals</i> NORMALTARIF CARDTARIF

Partnerunternehmen
<i>attributes</i> name rabatt

**Aufgabe 2: Entwurf und Implementierung (64 Punkte)**

Ausgehend vom Domänenmodell wurde das Entwurfsmodell entwickelt. Das Entwurfsklassendiagramm (auf Extrablatt) zeigt einen Ausschnitt aus dem Entwurfsmodell. Dabei wurden am Domänenmodell Änderungen, Ergänzungen und Verfeinerungen vorgenommen.

Die Klassen *Ausleihe* und *Rueckgabe* wurden in der Klasse *Fahrradausleihe* vereinigt, indem nur noch der Ausleihmonat (*monat* und zugehöriges *jahr*) sowie die Dauer der Ausleihe in Anzahl der Tage (*anzTage*) und Anzahl von Minuten eines nicht vollständigen Tages (*minuten*) erfasst wird.

Die Abrechnung für den Kunden erfolgt monatlich. Das *Kundenkonto* speichert alle Monatsrechnungen (*meineRechnungen*) in einer *Map*. Der *Monat* der *Rechnungen* wird aus der *Fahrradausleihe* (*monat, jahr*) erzeugt.

Eine *Rechnung* listet für jede *Fahrradausleihe* einen *Rechnungsposten* auf. Der Preis (*postenpreis*) für eine *Fahrradausleihe* wird auf Basis des gewählten Tarifes (*tarif*) und eines ggfs. gewährten Rabatts (*rabatt*) berechnet. Der Rabatt ist kleiner als 1,0 (kein Rabatt bedeutet *rabatt*=0, 20% Rabatt bedeutet *rabatt*=0.2). Der Gesamtpreis (*rechnungspreis*) ergibt sich aus der Summe der Postenpreise (*postenpreis*) und des darauf gewährten Rabatts (*rabatt*). Jeder *Rechnungsposten* wird der *Rechnung* hinzugefügt. Die *Rechnung* wird dem *Kundenkonto* hinzugefügt (*addRechnung()*).

Der Kunde kann zwischen zwei Tarifen (*tarifoption*) wählen. Beim *NORMALTARIF* wird jede (angefangene) halbe Stunde (30 min) mit 1 EUR berechnet. Ein ganzer Tag kostet 9 EUR. Kunden, die eine RadCard gekauft haben (*CARDTARIF*), bekommen für jede Ausleihe 1 EUR Rabatt..

Die Erstellung einer Monatsrechnung (*erstelleMonatsrechnung()*) wird durch das Sequenzdiagramm (auf Extrablatt) dargestellt.

**Lösen Sie folgende Teilaufgaben:**

- I. **Erstellen Sie ein Objektdiagramm für das beschriebene Szenario! Beachten Sie, dass das Objektdiagramm zum Entwurfsdiagramm konsistent sein muss! Verwenden Sie im Objektdiagramm ausschließlich die Objekte, die im Sequenzdiagramm dargestellt sind! Zur Beschreibung der Objekte im Objektdiagramm genügen der Objektname und die Klasse.**
  
- II. **Geben Sie ein Entwurfsmuster an, welches im Entwurf enthalten ist!**
  - **Achtung, raten Sie nicht! Für falsch angegebene Entwurfsmuster gibt es Punktabzug.**
  - **Tragen Sie das gefundene Entwurfsmuster als UML-Kollaboration in das Entwurfsklassendiagramm ein!**
  
- III. **Vervollständigen Sie die Implementierung der folgenden Java-Klassen (auf den Seiten 6 bis 8)! Parameter in Methoden (übergebene Objekte) müssen nicht auf null bzw. falsche Werte getestet werden!**

```
import java.util.*; // gilt für alle Klassen  
public class Kundenkonto
```

```
private void druckeRechnung(Rechnung rechnung){  
    // soll nicht implementiert werden  
}
```

```
public class Rechnung
```

**public**

**class Rechnungsposten**

**public class Normaltarif**

**public class CardTarif**