

# Übungsblatt 1

## **Aufgabe 1** Ein einfaches Klassendiagramm

Modellieren Sie ein UML Klassendiagramm, das folgenden Sachverhalt beschreibt:

- Jede Person hat einen Namen, eine Telefonnummer und eine E-Mail Adresse.
- Jede Wohnadresse wird von maximal einer Person genutzt.
- Wohnadressen bestehen aus einer Straße, einer Stadt, einer Postleitzahl und einem Land.
- Wohnadressen können bestätigt werden und als Beschriftung gedruckt werden.
- Es gibt zwei Arten von Personen: Student und Professor.
- Studenten können sich in Module einschreiben, haben eine Matrikelnummer und eine Durchschnittsnote.
- Professoren haben ein Einkommen und sind für Module verantwortlich.

## **Aufgabe 2** Use-Case Diagramm

Erstellen Sie ein UML Anwendungsfalldiagramm um die Aufgaben eines Anwendungssystems zur Unterstützung der Geschäftsprozesse eines Krankenhauses darzustellen. Das System soll folgende Aufgaben beinhalten:

- Adam und Eve sind in der Verwaltung angestellt.
- Adam soll Mitarbeiter einstellen und entlassen können.
- Beide können Patienten aufnehmen und entlassen.
- Sowohl beim Einstellen von Mitarbeitern, als auch beim Aufnahmen von Patienten müssen Daten (Name und Adresse ) erfasst werden. (Diese Tätigkeit sollte ausgelagert werden um eine redundante Beschreibung zu vermeiden)
- Beim Erfassen der Daten sollte im Falle einer lokalen Adresse überprüft werden, ob diese Adresse wirklich existiert.

## **Aufgabe 3** Ein komplexeres Klassendiagramm

In dem zu erstellenden UML Klassendiagramm soll ein objektorientierter Entwurf für ein Schachspiel dargestellt werden.

Ein Schachspiel besteht aus einem Schachbrett und 2 Teams. Das Schachbrett besteht aus 64 Felder, die jeweils durch eine x- und y-Koordinate ausgezeichnet sind. Die beiden Teams unterscheiden sich durch das Attribut Farbe (schwarz und weiß). Jede Mannschaft besteht aus insgesamt 16 Figuren. Das sind 8 Bauern, 2 Türme, 2 Läufer, 2 Springer, eine Dame und ein König.

Folgende Regeln sollten durch das Modell berücksichtigt werden:

- Jede Figur steht entweder auf einem Feld oder wurde bereits geschlagen.
- Jede Figur weiß, zu welcher Mannschaft sie gehört.
- Jedes Team kennt alle ihm zugehörigen Figuren.
- Eine Figur kann nicht gleichzeitig sowohl zwei Figuren darstellen. Eine Figur kann somit nicht gleichzeitig Bauer und Läufer sein.
- Jedes Feld kennt seinen aktuellen Besetzungszustand (Besetzt? Wenn ja: Von wem?).
- Jede Figur sollte über eine Operation *move* verfügen, die Spielzüge ermöglicht.

#### **Aufgabe 4** Ein UML Aktivitätsdiagramm

Zeichnen Sie ein UML Aktivitätsdiagramm, welches die Implementierung folgender Methode repräsentiert:

```
void Maximum(int[] a)
{
    if (a == null || a.Length == 0)
    {
        log("Fehler");
    }
    else
    {
        int b = a[0];

        for (int i = 1; i < a.Length; i++)
        {
            if (a[i] > b)
                b = a[i];
        }

        log(b);
    }
}
```

**! Wichtig**

Alle Diagramme können über Anwendungen (z.B. PlantUML, yUML, Visual Studio, ...) oder auch per Hand gezeichnet werden.