

Aufgabe 1: Sequenzdiagramm aus Klassendiagramm

Gegeben sei das auf der folgenden Seite dargestellte Klassendiagramm zur Verwaltung der Spiele der WM 2010.

Klasse Spiel:

- Spiel.ID ... die Identifikationsnummer des Spiels
- addTor ... fügt ein Tor zu einem Spiel hinzu
- addTeam ... fügt ein Team zu einem Spiel hinzu

Klasse Team:

- Teambez ... Name des Teams
- addGoal ... fügt ein Tor zu einem Team hinzu
- addSpiel ... fügt ein Spiel zu einem Team hinzu

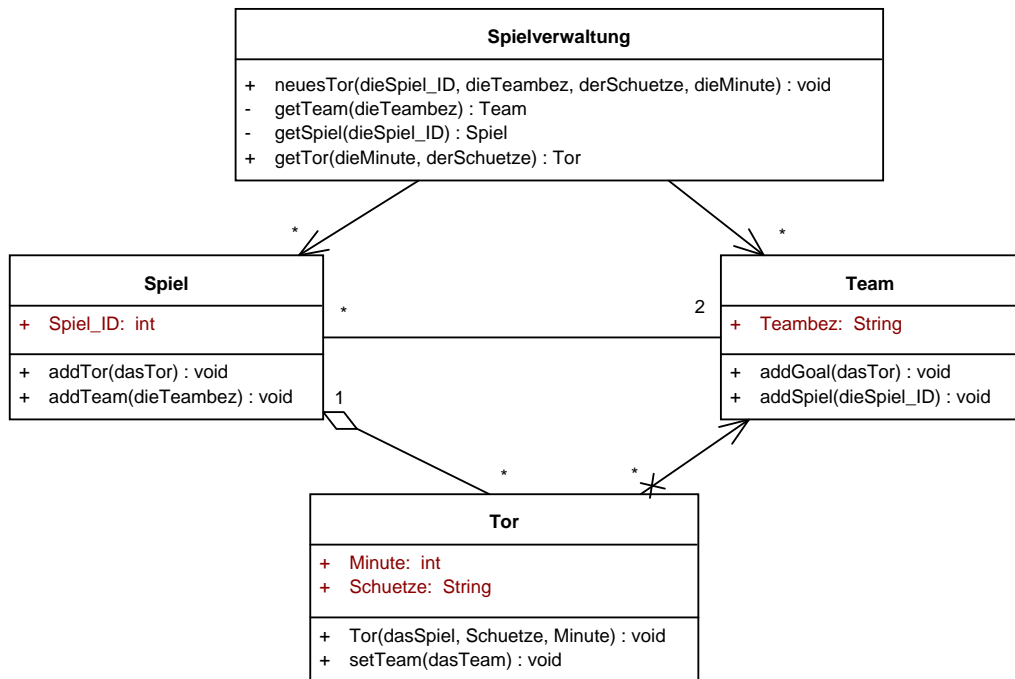
Klasse Tor:

- Minute ... die Minute, in der das Tor gefallen ist
- Schuetze ... der Torschütze (derjenige, der das Tor geschossen hat)
- Tor ... „erzeugt“ ein neues Tor und setzt Minute, Schuetze, aber nicht das Team das es geschossen hat
- setTeam ... fügt das Team zu einem Tor hinzu

Klasse Spielverwaltung:

- neuesTor ... ein neues Tor wird geschossen
- getSpiel ... ein Spiel wird abgefragt
- getTeam ... ein Team wird abgefragt
- getTor ... ein Tor wird abgefragt

Aufgabe: Erstellen Sie ein Sequenzdiagramm, um die Sequenz der Aufrufe von Operationen für die erfolgreiche Aufnahme eines neuen Tors (für ein bereits bestehendes Spiel) ins System zu zeigen. Wird ein Interaktionspartner erst im Laufe der Interaktion erzeugt, so geben Sie explizit den Konstruktor an, der das Objekt erzeugt.



Aufgabe 2: Darstellung von Programmabläufen mittels Sequenzdiagramm

Stellen Sie die Abläufe von folgendem Programm mittels Sequenzdiagramm dar. Modellieren Sie auch allfällige Antwortnachrichten.

Sie können davon ausgehen, dass alle nicht explizit deklarierten Variablen bereits deklariert und initialisiert sind.

```
class Main {
    ....
    Worker w = s1.getConnection(user , pw);

    if(w==null) {
        print("Error");
        exit;          // Programm wird beendet
    }

    do {
        m = w.getMail();
        print(m);
    } while (m != null);

    status = w.sendMail("abc", "test");
    ...
    private void print(String m) {
        ...
    }
}
```

```
class Server {

    public Worker getConnection(
        String user , String pw) {
        Worker w = new Worker();
        w.start();
        return w;
    }
}

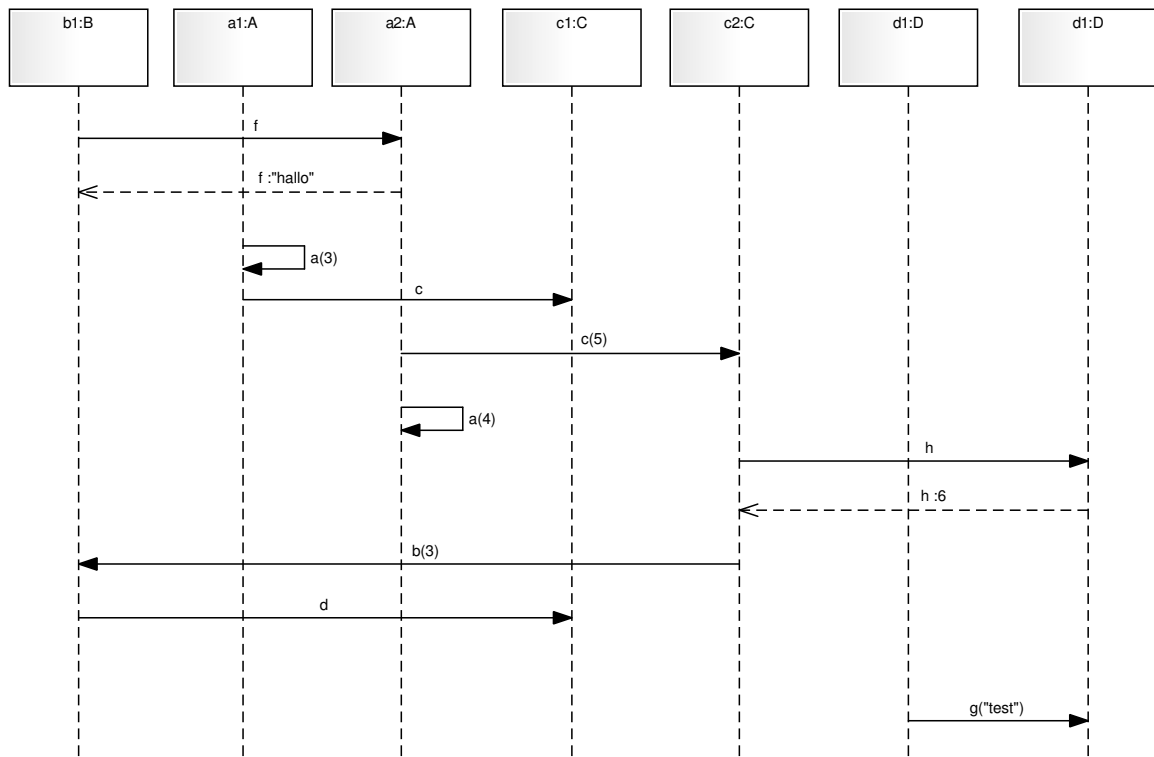
class Worker extends Thread {
    public void start() { }

    public boolean sendMail
        (String msg, String receiver) {...}

    public String getMail() {...}
}
```

Aufgabe 3: Klassendiagramm aus Sequenzdiagramm

Gegeben ist folgendes Sequenzdiagramm:



Vervollständigen Sie nachfolgendes Klassendiagramm

- Operationsdefinitionen mit Typangaben, soweit ersichtlich
- Beziehungen zwischen Klassen in Form von navigierbaren Assoziationen: Zeichnen Sie nur Navigationsrichtungen ein, die aus dem gegebenen Sequenzdiagramm ersichtlich sind

