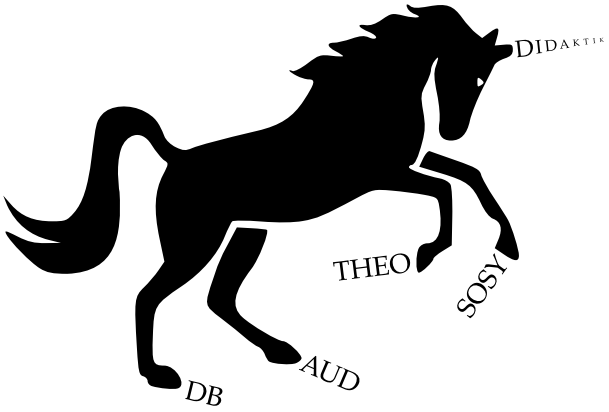
46116
Frühjahr
2011

Softwaretechnologie / Datenbanksysteme (nicht vertieft)

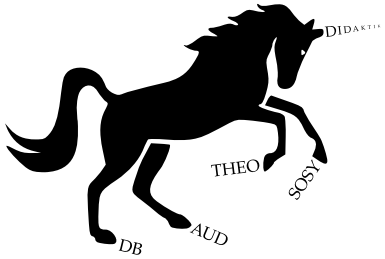
Aufgabenstellungen mit Lösungsvorschlägen



Die Bschlangaul-Sammlung

Hermine Bschlangaul and Friends

Aufgabenübersicht



Die Bschlangaul-Sammlung
Hermine Bschlangaul and Friends

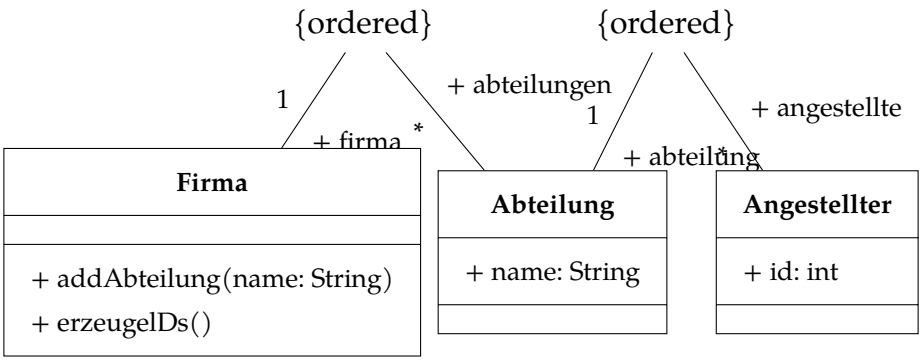
Eine freie Aufgabensammlung mit Lösungen von Studierenden für Studierende zur Vorbereitung auf die 1. Staatsexamensprüfungen des Lehramts Informatik in Bayern.

Thema Nr. 1

Teilaufgabe Nr. 2

1 Firmenstruktur [Firmenstruktur]

Firmenstruktur

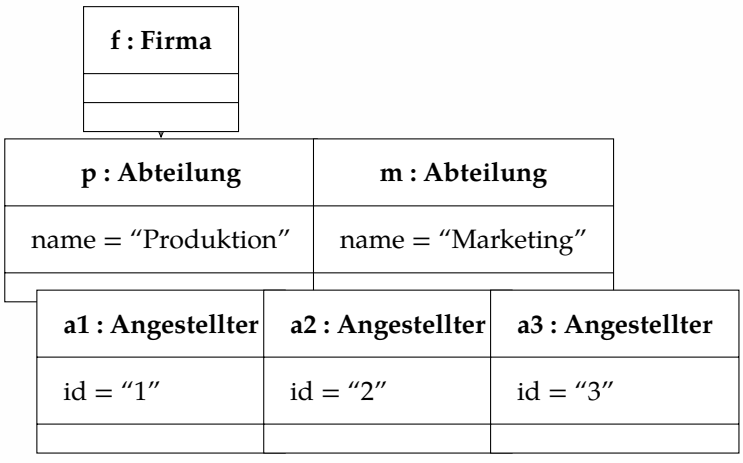


Eine Firma besteht aus `null` oder mehr Abteilungen, von denen jede `null` oder mehr Angestellte hat. Da sowohl die Abteilungen als auch deren Angestellten geordnet sind, sind Angestellte insgesamt geordnet. Sie haben durchgehende, ganzzahlige IDs, die bei 1 beginnen.

(a) Erstellen Sie exemplarisch ein Objektdiagramm: Stellen Sie eine Firma mit dem Instanznamen `f` und den zwei Abteilungen „Produktion“ (Name `p`) und „Marketing“ (Name `m`) dar. Die Produktion hat zwei Angestellte, Marketing hat einen Angestellten. Die Angestellten haben die Namen `a1`, `a2`, und `a3`.

Lösungsvorschlag

Die Instanzbezeichnung müsste noch unterstrichen werden. Das geht aber leider mit TikZ-UML nicht.



(b) Implementieren Sie das Klassendiagramm in Java oder in einer anderen geeigneten objektorientierten Programmiersprache Ihrer Wahl. Beachten Sie, dass die Assoziationen bidirektional und geordnet sind. Die beiden Methoden der Klasse `Firma` sollen dabei folgendes Verhalten haben:

Die Methode `erzeugeIDs` sorgt dafür, dass die IDs wieder korrekt zugewiesen sind. Die alten IDs können beliebig geändert werden, solange das Endergebnis wieder den obenstehenden Kriterien genügt.

```java
import java.util.ArrayList;
import java.util.List;

public class Firma {

  List<Abteilung> abteilungen;

  public Firma() {
    abteilungen = new ArrayList<Abteilung>();
  }

  public void addAbteilung(String name) {
    for (Abteilung abteilung : abteilungen) {
      if (abteilung.name.equals(name)) {

        ↪  System.out.println("Eine Abteilung mit diesem Namen gibt es bereits schon.");
        return;
      }
      abteilungen.add(new Abteilung(name));
    }
  }

  public void erzeugeIDs() {
    int idZähler = 1;
    for (Abteilung abteilung : abteilungen) {
      for (Angestellter angestellter : abteilung.angestellte) {
        angestellter.id = idZähler;
        idZähler++;
      }
    }
  }
}
```

Code-Beispiel auf Github ansehen: `src/main/java/org/bschlangaul/examen/examen_46116/jahr_2011/fruehjahr/Firma.java`

```java
import java.util.ArrayList;
import java.util.List;

public class Abteilung {
  public String name;

  public List<Angestellter> angestellte;

  public Abteilung(String name) {
```

```
    this.name = name;
    angestellte = new ArrayList<Angestellter>();
  }
}
```

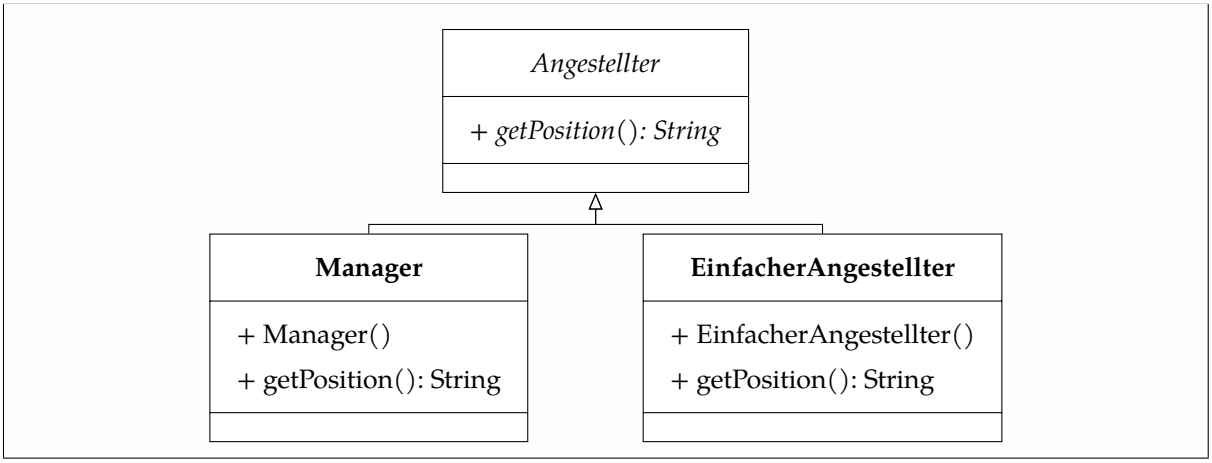
Code-Beispiel auf Github ansehen: `src/main/java/org/bschlangaul/examen/examen_46116/jahr_2011/fruehjahr/Abteilung.java`

```
import java.util.List;

public class Angestellter {
  public int id;

  public Firma firma;

  public List<Angestellter> angestellte;
}
```

Code-Beispiel auf Github ansehen: `src/main/java/org/bschlangaul/examen/examen_46116/jahr_2011/fruehjahr/Angestellter.java`

(c) Angestellte sollen in Manager und einfache Angestellte unterteilt werden. Zeichnen Sie ein Klassendiagramm mit der Oberklasse `Angestellter` und den zwei Unterklassen `Manager` und `EinfacherAngestellter`. Die Klasse `Angestellter` soll nicht instantiierbar sein und erzwingen, dass die Methode `getPosition()` (öffentlich, ohne Argumente, Rückgabewert `String`) von allen konkreten Unterklassen implementiert wird. `Manager` und `EinfacherAngestellter` sollen instantiierbar sein.

Lösungsvorschlag



(d) Wie lautet der Fachbegriff dafür, dass eine Methode in einer Klasse und in deren Unterklassen dieselbe Signatur hat, aber in den Unterklassen unterschiedlich implementiert ist?

Lösungsvorschlag

Abstrakte Methode