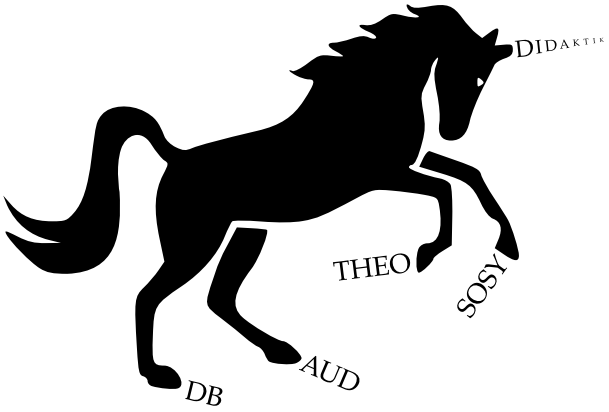
46116
Frühjahr
2010

Softwaretechnologie / Datenbanksysteme (nicht vertieft)

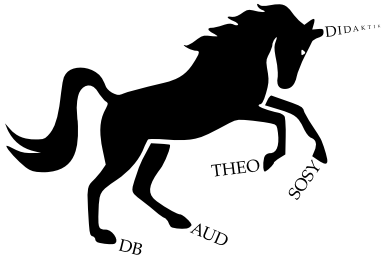
Aufgabenstellungen mit Lösungsvorschlägen



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.

Diese Materialsammlung unterliegt den Bestimmungen der Creative Commons Namensnennung-Nicht kommerziell-Share Alike 4.0 International-Lizenz.

Thema Nr. 1

Aufgabe 1 [Reiseunternehmen]

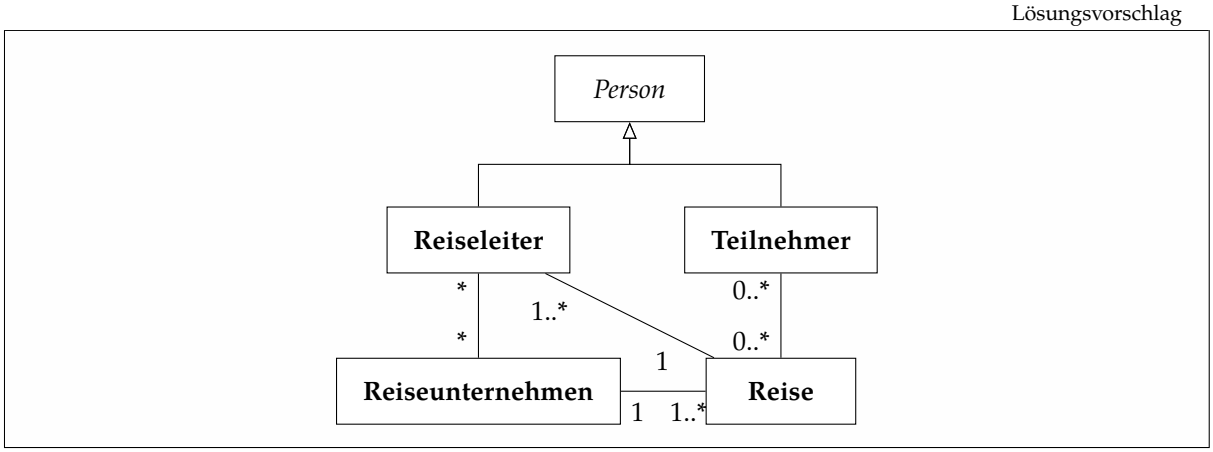
Es sei folgender Sachverhalt gegeben:

Ein Reiseunternehmen bietet verschiedene Reisen an. Dazu beschäftigt es eine Reihe von Reiseleitern, wobei eine Reise von mindestens einem Reiseleiter geleitet wird. Da Reiseleiter freiberuflich arbeiten, können sie bei mehreren Reiseunternehmen Reisen leiten.

An einer Reise können mehrere Teilnehmer teilnehmen, ein Teilnehmer kann auch an verschiedenen Reisen teilnehmen.

(a) Modellieren Sie diesen Sachverhalt in einem UML-Klassendiagramm. Für Teilnehmer und Reiseleiter sollen Sie dabei eine abstrakte Oberklasse definieren. Achten Sie dabei auf die Multiplizitäten der Assoziationen. Sie müssen keine Attribute bzw. Methoden angeben.

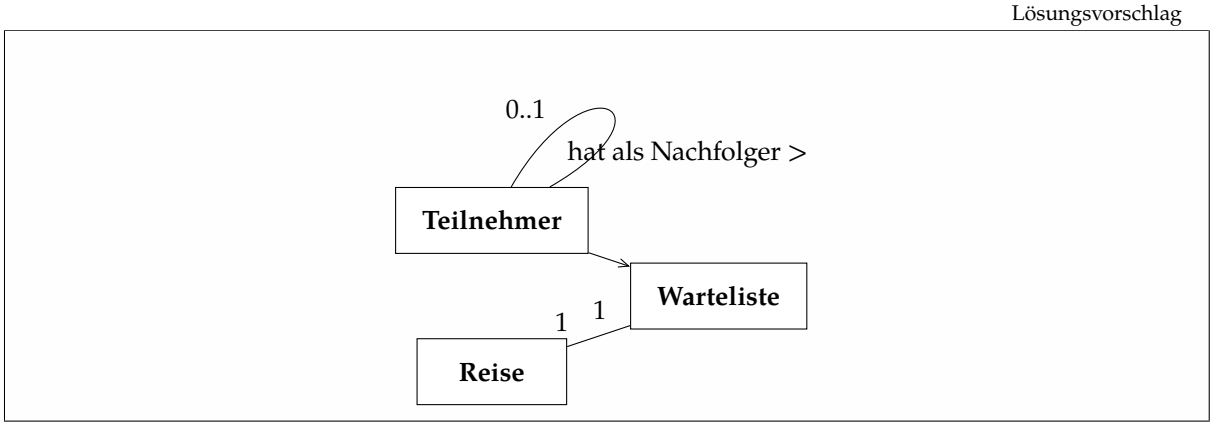
Lösungsvorschlag



(b) Eine Reise kann jedoch nur mit einer begrenzten Kapazität angeboten werden, das heißt, zu einer bestimmten Reise kann nur eine begrenzte Anzahl von Teilnehmern assoziiert werden. Als Ausgleich soll pro Reise eine Warteliste verwaltet werden.

Modellieren Sie diesen erweiterten Sachverhalt in einem neuen Diagramm. Nicht veränderte Klassen brauchen nicht noch einmal angegeben werden. Beachten Sie dabei, dass die Reihenfolge bei einer Warteliste eine Rolle spielt.

Lösungsvorschlag



(c) Implementieren Sie die in Aufgabenteil b) modellierten Klassen in Java. Fügen Sie eine Methode hinzu, die einen Teilnehmer von einer Reise entfernt. Dabei soll automatisch der erste Platz der Warteliste zu einem Reiseteilnehmer werden, wenn die Warteliste nicht leer ist. Achten Sie auf die Navigierbarkeit Ihrer Assoziationen. Sie können davon ausgehen, dass die Methode nur mit Teilnehmern aufgerufen wird, die in der Tat Teilnehmer der Reise sind.

```java
public class Teilnehmer {
  Teilnehmer nächster;

}
```

Code-Beispiel auf Github ansehen: `src/main/java/org/bschlangaul/examen/examen_46116/jahr_2010/fruehjahr/reiseunternehmen/Teilnehmer.java`

```java
/**
 * Diese Klasse ist eine Implementation einer einfach verketteten Liste. Sie
 * wird einerseit in der Klasse {@link Reise} genutzt, um die Reiseteilnehmer zu
 * speichern, andererseits um eine Warteliste darauf aufbauen zu könnten.
 */
public class TeilnehmerListe {

}
```

Code-Beispiel auf Github ansehen: `src/main/java/org/bschlangaul/examen/examen_46116/jahr_2010/fruehjahr/reiseunternehmen/TeilnehmerListe.java`

```java
/**
 * Eine Reise kann jedoch nur mit einer begrenzten Kapazität angeboten
 * werden, das heißt, zu einer bestimmten Reise kann nur eine begrenzte
 * Anzahl von Teilnehmern assoziiert werden. Als Ausgleich soll pro
 * Reise eine Warteliste verwaltet werden.
 *
 * Modellieren Sie diesen erweiterten Sachverhalt in einem neuen
 * Diagramm. Nicht veränderte Klassen brauchen nicht noch einmal
 * angegeben werden. Beachten Sie dabei, dass die Reihenfolge bei einer
 * Warteliste eine Rolle spielt.
 *
 * Implementieren Sie die in Aufgabenteil b) modellierten Klassen in
 * Java. Fügen Sie eine Methode hinzu, die einen Teilnehmer von einer
 * Reise entfernt. Dabei soll automatisch der erste Platz der Warteliste
 * zu einem Reiseteilnehmer werden, wenn die Warteliste nicht leer ist.
 * Achten Sie auf die Navigierbarkeit Ihrer Assoziationen. Sie können
 * davon ausgehen, dass die Methode nur mit Teilnehmern aufgerufen wird,
 * die in der Tat Teilnehmer der Reise sind.
 */
public class Warteliste extends TeilnehmerListe {

}
```

Code-Beispiel auf Github ansehen: `src/main/java/org/bschlangaul/examen/examen_46116/jahr_2010/fruehjahr/reiseunternehmen/Warteliste.java`

```java
public class Reise {


  Teilnehmer teilnehmer;
```

```
  Warteliste warteliste;

  void entferneTeilnehmer(Teilnehmer teilnehmer) {

  }

}
```

Code-Beispiel auf Github ansehen: `src/main/java/org/bschlangaul/examen/examen_46116/jahr_2010/fruehjahr/reiseunternehmen/Reise.java`