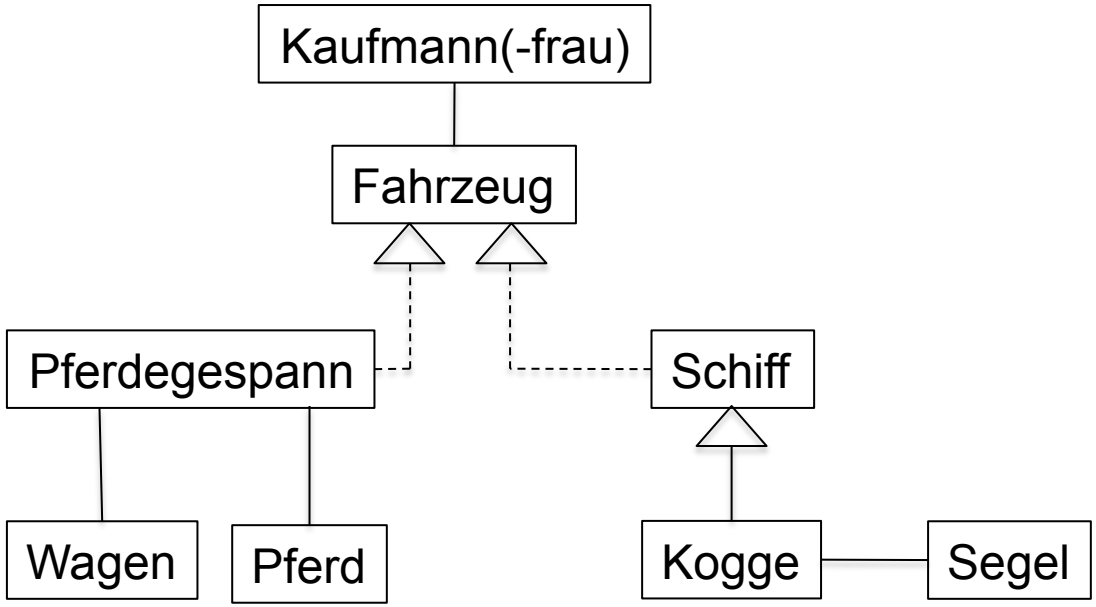
HAW Hamburg
Fakultät TI
Technik und Informatik

Assoziationen, Basisklasse Object, Rekursion

Lösungen zu den Übungsaufgaben

Übung: Beziehungen zwischen Klassen



Übung: Equals

- Konto

```
@Override
public boolean equals(Object anderesObjekt) {
  if (!(anderesObjekt instanceof Konto)) {
    // Kein kompatibler Typ: false
    return false;
  }
  Konto anderesKonto = (Konto) anderesObjekt;
  // Vergleiche Kontostand und Kontoinhaber
  return Math.abs(kontostand - anderesKonto.kontostand) < 1e-5
      && kontoinhaber.equals(anderesKonto.kontoinhaber);
}
```

- Person

```
  @Override
  public boolean equals(Object anderesObjekt) {
    if (anderesObjekt instanceof Person) {
      // Fall das andere Objekt auch eine Person ist, vergleiche
Namen.
      return name.equals(((Person) anderesObjekt).name);
    }
    // Ansonsten: in jedem Fall false
    return false;
  }
```

Übung: HashCode


```
  @Override
  public int hashCode() {
    // Buchstabencode wird als Zahlenwert zwischen 0 und 26 interpretiert. Falls
    // wahrheitswert true ist, wird auf den HashCode 26 addiert. Damit ist der
    // HashCode eindeutig und aus [0,52]
    return (int) buchstabe - (int) 'a' + (wahrheitswert ? 26 : 0);
  }

  @Override
  public boolean equals(Object anderesObjekt) {
    if (!(anderesObjekt instanceof WahrheitUndBuchstabe)) {
      return false;
    }
    // Da der HashCode eindeutig ist, darf ich ihn für die Gleichheit verwenden,
    // ansonsten nur um ungleiche Instanzen zu finden!
    return hashCode() == anderesObjekt.hashCode();
  }
}
```

Übung: Summe


```java
/**
 * Berechnung der ganzen Zahlen von 1...n. n muss >= 1 sein.
 */
public static int summe(int n) {
  if (n == 1) {
    return 1;
  }
  return n + summe(n - 1);
}
```

Übung: Rekursion statt Schleife


```java
/**
 * Rekursive Erzeugung einer Alphabet-Zeichenkette
 * ("abcdefghijklmnopqrstuvwxyz").
 */
public static String erzeugeAlphabetrekursiv() {
  return erzeugeAlphabetrekursivHilf(25);
}

/**
 * Rekursive Hilfsmethode für erzeugeAlphabetrekursiv.
 */
public static String erzeugeAlphabetrekursivHilf(int index) {
  if (index == 0) {
    return "a";
  } else {
    return erzeugeAlphabetrekursivHilf(index - 1) + (char) (index
'a');
  }
}
```

Hilfsmethode, da die rekursive Variante einen Parameter benötigt.

Übung: Addition


```java
/**
 * Rechner für Grundrechenarten mit rekursiver Implementierung.
 */
public class Rechner {

  /**
   * Addieren der beiden ganzen Zahlen zahl1 und zahl2. Beide müssen
>= 0 sein.
   */
  public int addiere(int zahl1, int zahl2) {
    if (zahl2 == 0) {
      return zahl1;
    } else {
      return addiere(zahl1 + 1, zahl2 - 1);
    }
  }

}
```