Übungen zu Kapitel 13

13.1 Semantik finaler Methoden:

Betrachten Sie folgendes Coding. Was wird am Bildschirm ausgegeben? Entspricht dies der Intention des Oberklassen-Autors? Wie liesse sich der "Missstand" einfach beheben?

```
class Oben {
   public tuWas() { IO.writeln( "Original" ); }
   public final void fin() { tuWas(); }
}
class Unten extends Oben {
   public tuWas() { IO.writeln( "Überschrieben" ); }
   public static void main( String[] args ) {
      Unten u = new Unten();    u.fin();
   }
}
```

13.2 Vererbung & Assoziation:

Erörtern Sie das semantische Verhältnis zwischen den Begriffen: Person, Kunde, Kundenbetreuer, Auftrag, Auftragsposition und Rechnung. Inwiefern können diese durch Vererbung bzw. Assoziation verbunden werden?

13.3 Klassenhierachie:

Legen Sie die Klassenhierachie gemäß Abbildung 13.5 an. Es handelt sich jeweils um zweikomponentige Dinge, wobei die Komponenten `comp1` und `comp2` je nach Klasse unterschiedliche semantische Bedeutung haben.

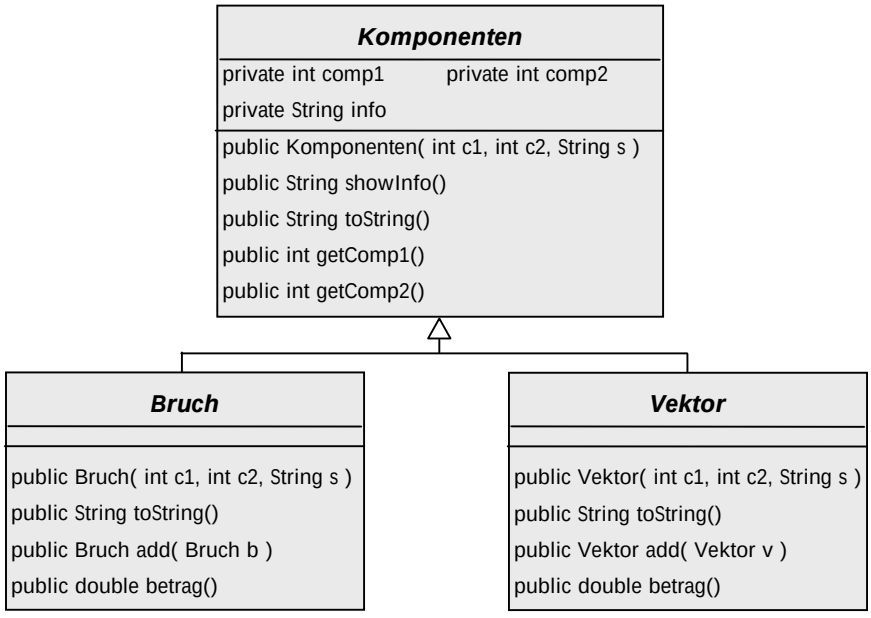

Abb.: Klassenhierachie Übung 13.3

In den Unterklassen-Konstruktoren soll der Oberklassen-Konstruktor mittels `super()`-Anweisung aufgerufen werden. Die Methode `showInfo()` soll die im String `info` gespeicherte Information zurückgeben. Die Methode `toString()` soll je nach Klasse einen anderen `String` liefern:

`Komponenten`:	"1.Komp. = ***comp1*** 2.Komp. = ***comp2*** "	
`Bruch`:	"***comp1***/***comp2*** "	z.B.: "2/3"
`Vektor`:	"(***comp1***, ***comp2***)"	z.B.: "(4, -17)"

Auserdem soll `toString()` den Text der Methode `showInfo()` aufnehmen.

Die Methoden `add()` addieren das übergebene Objekt zum aufrufenden Objekt gemäß Regeln der Bruchrechnung bzw. Vektoraddition für Vektoren der Ebene.

Die Methode `betrag()` berechnet den Betrag des Unterklassenobjekts.

`Bruch`: `double`-Bruchwert = ***comp1***/***comp2***
`Vektor`: `double`-Vektorlänge = Math.sqrt(***comp1*** *****comp1*** + ***comp2*** *****comp2***)

Implemetieren Sie diese Klassen! Legen Sie eine ausführbare Klasse `Vererbung` an, in deren `main()`-Methode Sie Objekte der Klassen erzeugen. Rufen Sie deren Methoden auf. Machen Sie aus `Komponenten` eine abstrakte Klasse.

13.4 Klasse `Object`*:*

Informieren Sie sich in der J2SE-API-Dokumentation [SUN05b] über die Methoden der Klasse `Object`.

.