



Übungsblatt 12

10.07., 11.07. und 14.07

Problem 12.1: Vererbung

1. Als Wiederholung der in Aufgaben 10.1-10.3 behandelten Techniken der objektorientierten Programmierung und als Basis für weitere Experimente bauen wir uns noch einmal einen virtuellen Zoo ähnlich zu dem von Aufgabe 7.2, aber diesmal mit Klassen statt mit Strukturen; außerdem wird es mehr Action geben, so dass wir Tiere statt Tierarten modellieren.
Um Ihr Programm einfach zu halten, deklarieren Sie – auch wenn man das eigentlich nicht tut – am besten erstmal alle Attribute und Methoden Ihrer Klassen als `public`, das lässt sich auch ohne Tutor bei Gelegenheit noch leicht reparieren.
 - (a) Schreiben Sie eine Klasse `tier` mit einem Attribut `art` vom Typ `string`.
Die Klasse soll einen Konstruktor mit einem `string` als Parameter haben, den der Konstruktor im Attribut `art` speichern soll. Geben Sie dem Parameter wieder einen Standardwert (vgl. Aufgabe 10.3, nur dass wir hier einen `String` brauchen, z.B. ein allgemeines "ein Tier" – den Artikel zum Artnamen dazu zu nehmen, macht es leichter, korrekte Sätze zu bilden).
Weiter soll die Klasse eine Methode `hallo` bekommen, mit der sich das Tier vorstellt – d.h., seine Tierart (und vielleicht ein wenig Text dazu, z.B. "Ich bin ") auf `cout` ausgibt.
Aus Gründen, die erst später deutlich werden, soll diese Methode einen Parameter `prefix` vom Typ `string` bekommen, der der Ausgabezeile vorangestellt wird. Wäre für eine Variable `fisch` vom Typ `tier` die Tierart z.B. "ein Fisch", könnte `fisch.hallo("Test: ");` eine Ausgabe folgender Art erzeugen:

```
Test: Ich bin ein Fisch!
```
 - (b) Ein Zoo hat hier auch wieder einen Vektor (hier z.B. `tiere` genannt) mit Elementen vom Typ `tier` – schreiben Sie eine entsprechende Klasse `zoo`.
Am Anfang soll der Zoo leer sein, dann brauchen wir keinen Konstruktor, aber es soll eine Methode `kaufe_tier` mit einem Parameter vom Typ `tier` geben, die das Tier hinten an den Vektor anfügt (vgl. Aufgabe 5.1).
Weiter soll es eine Methode `alle_hallo` geben, die für jedes der Tiere im Zoo dessen Methode `hallo` aufruft (mit dem leeren String "" als Parameter `prefix`).
 - (c) Legen Sie in der Funktion `main` drei Variablen für Tiere an: Einen Fisch, eine Robbe und einen Orca, und packen Sie die Tiere in einen Zoo.
Rufen Sie die Methode `hallo` des Orca auf und die Methode `alle_hallo` des Zoos.
2. Damit in unserem Zoo etwas passiert (und wir endlich ein Beispiel zur Vererbung bekommen), kommen nun Raubtiere.
 - (a) Leiten Sie dazu eine Klasse `raubtier` von der Klasse `tier` ab (`public`). Die Erweiterung besteht darin, dass ein Raubtier ein anderes Tier fressen kann, das machen wir hier mit einem weiteren Attribut `im_bauch` vom Typ „Zeiger auf `tier`“ (es ist ein Zeiger, weil sonst die Beute beim Fressen, also der Zuweisung, kopiert würde, was völlig unbiologisch wäre...).
Der Konstruktor soll das geerbte Attribut `art` wie gehabt behandeln und den Zeiger `im_bauch` auf den Nullzeiger `nullptr` (Breyman, Kap. 4.1, S. 198) setzen. Eine Methode `friss` mit

einem Parameter vom Typ „Zeiger auf tier“ beschreibt das Fressen: der Parameter wird im Attribut `im_bauch` abgespeichert.

- (b) Überschreiben Sie in der Klassendefinition `raubtier` die geerbte Methode `hallo` aus `tier`. Sie soll nun zuerst die geerbte Methode aufrufen (`tier::hallo(prefix);`) und dann, sofern der Zeiger `im_bauch` nicht der Nullzeiger ist, die Methode `hallo` des Tieres, für das dort ein Zeiger gespeichert ist.

Vorsicht: Wenn wir eine Variable, z.B. `t`, vom Typ *Zeiger auf tier* haben, müssen wir sie zum Aufruf der Methode `hallo` dereferenzieren (`(*t).hallo("")`) oder besser `t->hallo("")`).

Bei Aufruf des Beute-`hallo` soll der Parameter `prefix` um z.B. zwei Leerzeichen wachsen (`prefix + " "`), so dass wir in der Bildschirmausgabe den Text für das gefressene Tier etwas eingerückt bekommen.

Wenn das Raubtier nichts im Bauch hat, soll es drucken, dass es hungrig ist.

- (c) Bauen Sie die Funktion `main` so um, dass die Robbe und der Orca Raubtiere sind. Dann soll die Robbe den Fisch und der Orca die Robbe fressen. Rufen Sie wieder die Methode `hallo` des Orca auf und die Methode `alle_hallo` des Zoos. Beobachten Sie jeweils, ob die Methode `hallo` aus `tier` oder die aus `raubtier` verwendet wird.