Technische Universität Berlin **Fachgebiet für Neuronale Informationsverarbeitung**

Übungsblatt 10

Prof. Dr. Klaus Obermayer und Ivo Trowitzsch

J

Vererbung, Polymorphie

Verfügbar ab:	**29.06.18**
Abgabe bis:	**08.07.18**

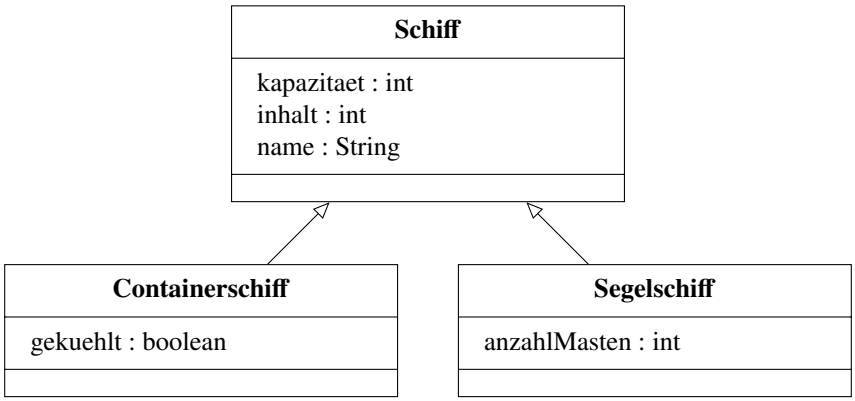
Aufgabe 1: Verständnis

1 Punkte

Betrachten Sie folgenden Klassen und Attribute:

Klassen	Attribute
	`String schnabelfarbe`
Fisch	`String fellfarbe`
Elefant	`double gewicht`
Tier	`boolean lebtImSalzwasser`
Saeugetier	`double ruessellaenge`
Vogel	`String bezeichnung`
Papagei	`double spannweite`
	`boolean kannSprechen`

Stellen Sie grafisch, d.h. in einer Baumstruktur, in welchem Vererbungsverhältnis die gegebenen Klassen zueinander stehen. Ordnen Sie dann die gegebenen Attribute den entsprechenden Klassen sinnvoll zu, ohne Attribute mehr als einer Klasse zuzuordnen.
Hinweis: Jedes Attribut darf nur einer Klasse zugeordnet werden (Brückentiere werden nicht beachtet).
Für eine Schiff-Klassen würde der Vererbungs-Baum ohne Berücksichtigung der Methoden und Sichtbarkeiten beispielsweise so aussehen:



Aufgabe 2: Fussballspieler 7 Punkte

Betrachten wir eine Fußballmannschaft. Diese besteht aus 11 Spielern: einem Torwart und 10 weiteren Spielern in beliebiger Kombination von Abwehr-, Mittelfeld- und Sturmspieler. Es soll nun eine Simulation von Mannschaften erstellt werden, welche aus diesen Spielern besteht. Gehen Sie dabei folgendermaßen vor:

1. Implementieren Sie die Klasse `Fussballspieler`. Diese soll die Attribute `name` und `alter` besitzen, auf welche nur in dieser Klasse zugegriffen werden kann. Außerdem besitzt die Klasse einen parametrisierten Konstruktor und entsprechende get-Methoden. Es soll keine(!) Möglichkeit geben, den Namen nach dem Anlegen zu verändern. Schreiben Sie weiterhin eine Methode `geburtstagFeiern`, welche das Alter um ein Jahr erhöht, sowie eine Methode `toString`, welche die relevanten Informationen als String zurückgibt.
2. Implementieren Sie die Klassen `Abwehrspieler`, `Mittelfeldspieler`, `Sturmspieler`, und `Torwart`. Diese sollen alle die Klasse `Fussballspieler` erweitern und eigene `toString`-Methoden, sowie get- und set-Methoden für ihre jeweiligen Attribute und einen parametrisierten Konstruktor besitzen, welcher Name, Alter und die jeweiligen klassenspezifischen Attribute übergeben bekommt.

   Ein `Torwart` besitzt das Attribut `gegentoreJeSpiel`, ein `Abwehrspieler` das Attribut `anzahlRoteKartenJeSpiel`, ein `Mittelfeldspieler` das Attribut `paesseJeSpiel` und ein `Sturmspieler` das Attribut `toreJeSpiel`.
3. Legen Sie jetzt eine Klasse `TestMannschaft` an. Erstellen Sie zunächst zwei Arrays aus jeweils 11 `Fussballspielern`. Befüllen Sie diese mit einer beliebigen Kombination aus 10 Feldspielern und einem Torwart. Lassen sie einen beliebigen Spieler Geburtstag feiern.
4. a) Schreiben Sie (in der Klasse `TestMannschaft`) eine Methode `roteKartenZaehlen`, welche ein Array aus `Fussballspielern` übergeben bekommt und die summierte `anzahlRoteKartenJeSpiel` der enthaltenen Abwehrspieler zurückgibt.
   b) Schreiben Sie eine Methode `ausgeben`, welche ebenfalls ein Array aus `Fussballspielern` übergeben bekommt und die `toString`-Methoden verwendet, um die Informationen aller Spieler auf der Konsole auszugeben.
   c) Schreiben Sie außerdem eine Methode `fussballspielen`, welche zwei Arrays aus `Fussballspielern` übergeben bekommt und auf der Konsole ausgibt, welche Mannschaft gewinnt. Es soll die Mannschaft gewinnen, welche die höhere Spielstärke hat. Denken Sie sich hierzu eine eigene Formel zur Berechnung der Spielstärke aus, welche alle Attribute der Spieler verwendet.

   Überlegen Sie sich für die beiden Methoden `roteKartenZaehlen` und `ausgeben`, inwiefern hier Polymorphie eine Rolle spielt. Schreiben Sie jeweils einen kurzen Kommentar, warum ihre Methode die richtigen (spezialisierten) Methoden der jeweiligen Klasse aufruft.
5. Testen sie die drei Methoden `roteKartenZaehlen`, `fussballspielen` und `ausgeben` in `main`.

Aufgabe 3: Bankkonto 7 Punkte

In dieser Aufgabe soll als erstes ein einfaches Bankkonto modelliert werden. Auf diesem aufbauend werden dann weitere Kontotypen hinzugefügt.

1. Ein Bankkonto muss den aktuellen Kontostand speichern können. Um die Implementierung möglichst einfach zu halten, soll der Kontostand in Cent gespeichert werden. Bei der Eröffnung des Kontos soll der Kontostand auf 0 gesetzt werden. Desweiteren müssen Methoden zur Ein- und Auszahlung implementiert werden. Eine Auszahlung soll nur dann erfolgen, wenn das Konto gedeckt ist. Dem Kontoinhaber sollte es möglich sein, seinen aktuellen Kontostand zu erfahren.
   Schreiben Sie eine Klasse `Konto`, welche die beschriebene Funktionalität darstellt.
2. Da eine Bank meistens mehrere Kontomodelle anbietet, soll nun ein weiteres, spezielleres Konto implementiert werden. Dieses Konto soll die selben Funktionen bieten wie das „Standard-Konto“ in 1. Es soll dem Inhaber zusätzlich einen Dispositionskredit einräumen (normalerweise 100 Euro), der beim Auszahlen berücksichtigt wird. Nachträglich kann dieser Kreditrahmen verändert werden. Der Kunde sollte seinen Kreditrahmen abfragen können.
   Implementieren Sie dieses Konto in der Klasse `DispoKonto` und verwenden Sie die von Ihnen vorher erstellte Klasse `Konto`.
3. Ein weiterer Kontotyp ist das „Jugendkonto“. Dieses Konto gibt es nur für Leute unter 18 Jahren, weshalb zusätzlich das Alter des Kontoinhabers gespeichert werden muss. Das Alter wird initial bei der Eröffnung des Kontos gesetzt. Es kann um ein Jahr erhöht werden und abgefragt werden. Zusätzlich kann bei diesem Konto ein Limit festgelegt werden, wie viel pro Auszahlung abgehoben werden darf. Initial soll dieses Limit bei 50 Euro liegen. Nachträglich ist das Limit auch veränderbar.
   Programmieren Sie die Klasse `JugendKonto`.
4. Für besonders gute Kunden bietet die Bank ein Konto mit Zinsen an. Dazu muss im jeweiligen Konto der aktuelle Zinssatz vermerkt werden. Der normale Zinsatz der Bank beträgt 1 %. Dieser sollte verändert sowie ausgegeben werden können. Die Berechnung der Zinsen und eine entsprechende Gutschrift soll ebenfalls möglich sein.
   Entwerfen sie eine entsprechende Klasse `ZinsKonto`.
5. Zur Überprüfung soll eine Test-Klasse `KontoTest` geschrieben werden, in welcher die implementierten Klassen getestet werden. Es soll jeweils mindestens ein Objekt der Klassen `Konto`, `DispoKonto`, `JugendKonto` und `ZinsKonto` erzeugt werden und für selbiges die implementierten Methoden sinnvoll verwendet werden, so dass ersichtlich wird, dass alle in 1.-4. genannten Anforderungen erfüllt sind.

Sämtliche Attribute sollen nicht öffentlich, die Methoden sollen jedoch nach außen hin sichtbar sein.
Vergessen Sie nicht Ihren Code sinnvoll zu kommentieren. Die Ausgabe Ihrer TestKonto-Klasse könnte z.B. so aussehen:

```
Konto eroeffnet
Aktueller Kontostand: 0 Cent
Zahle 100,03 Euro ein.
Aktueller Kontostand: 10003 Cent
Hebe 30,09 Euro ab.
Aktueller Kontostand: 6994 Cent
Hebe 99,99 Euro ab.
Auszahlung nicht moeglich.
Aktueller Kontostand: 6994 Cent
Zahle 50,68 Euro ein.
Aktueller Kontostand: 12062 Cent

DispoKonto mit Dispo (100,00 Euro) eroeffnet
Dispokredit: 10000 Cent
```

```
Aktueller Kontostand: 0 Cent
Zahle 100,03 Euro ein.
Aktueller Kontostand: 10003 Cent
Hebe 130,09 Euro ab.
Aktueller Kontostand: -3006 Cent
Hebe 99,99 Euro ab.
Auszahlung nicht moeglich.
Aktueller Kontostand: -3006 Cent
Setze Dispo auf 200,00
Dispokredit: 20000 Cent
Hebe 99,99 Euro ab.
Aktueller Kontostand: -13005 Cent
Zahle 50,68 Euro ein.
Aktueller Kontostand: -7937 Cent

JugendKonto eroeffnet
Alter des Kontoinhabers: 15
Aktuelles Limit: 5000 Cent
Aktueller Kontostand: 0 Cent
Kontoinhaber hat Geburtstag
Alter des Kontoinhabers: 16
Zahle 1200,03 Euro ein.
Aktueller Kontostand: 120003 Cent
Hebe 130,09 Euro ab.
Auszahlung nicht moeglich.
Aktueller Kontostand: 120003 Cent
Aendere Limit auf 500,00 Euro
Aktuelles Limit: 50000 Cent
Hebe 500,00 Euro ab.
Aktueller Kontostand: 70003 Cent

ZinsKonto eroeffnet
Aktueller Zinssatz: 1.0
Aktueller Kontostand: 0 Cent
Zahle 100,00 Euro ein.
Aktueller Kontostand: 10000 Cent
Verzinsen
Aktueller Kontostand: 10100 Cent
Hebe 20,99 Euro ab.
Aktueller Kontostand: 8001 Cent
Aendere Zinssatz auf 3,5%
Aktueller Zinssatz: 3.5
Verzinsen
Aktueller Kontostand: 8281 Cent
```