

Übungsblatt Vererbung

Aufgabe 1: (Vererbung: Generalisierung)

Schreiben Sie eine abstrakte Klasse `Person` mit der abstrakten Methode

`gibTaetigkeitAus()`, die die Tätigkeit der `Person` auf die Konsole ausgeben soll.

Leiten Sie die konkreten Klassen `Student` und `Professor` ab.

Erzeugen Sie ein Feld von 100 Personen, abwechselnd `Student` und `Professor`. Rufen Sie bei allen 100 Personen die Methode `gibTaetigkeitAus()` auf.

Bei einem Studenten soll folgende Ausgabe erfolgen:

`Der Student studiert.`

Bei einem Professor soll folgende Ausgabe erfolgen:

`Der Professor lehrt.`

Aufgabe 2: (Vererbung: Generalisierung)

Schreiben Sie ein Programm, das einen Quizbogen für einen Nutzer als Text generiert. Ein Quiz soll aus zwei Arten von Quizfragen bestehen:

- (1) Textfragen, die ein einzelnes Wort als richtige Antwort haben. Eine Textfrage könnte wie folgt auf dem Quizbogen erscheinen:

Was ist die Hauptstadt von Deutschland?

Antwort:

<Zwei Leerzeilen für eine Antwort>

- (2) Multiple-Choice Fragen, die ein Array von möglichen Antworten speichern. Eine Multiple-Choice Frage könnte wie folgt auf dem Quizbogen erscheinen:

Fragentext:

Wie viele Protonen hat ein Wasserstoff-Atom?

Antwortmöglichkeiten

A: 4

B: 2

C: 1

D: 0

Legen Sie ein Quiz mit mehreren Text- sowie Multiple-Choice Fragen an und generieren Sie einen Quizbogen.

Aufgabe 3: (Vererbung: Spezialisierung)

Schreiben Sie ein Programm, das die Struktur eines Unternehmens abbildet. In dem Unternehmen gibt es mehrere Abteilungen. In jeder Abteilung arbeiten mehrere Angestellte und ein Abteilungsleiter.

Ein Angestellter soll einen Nachnamen, Vornamen, einen Identifikator, ein Grundgehalt, einen Gehaltsfaktor sowie ein Geburtsdatum haben. Für alle Attribute sollen getter- und setter zur Verfügung stehen. Der Gehaltsfaktor für einen Angestellten ist 1. Das Gehalt errechnet sich aus dem Gehaltsfaktor multipliziert mit dem Grundgehalt.

Ein Abteilungsleiter soll ein Angestellter sein und ebenfalls einen Nachnamen, Vornamen, einen Identifikator, ein Grundgehalt, einen Gehaltsfaktor sowie ein Geburtsdatum haben. Der Gehaltsfaktor beträgt bei einem Abteilungsleiter 2. Ferner soll ein Abteilungsleiter eine Methode befördern haben, in der ein Angestellter befördert werden kann. Dieser Angestellte soll dann seinen Gehaltsfaktor um 10% erhöht bekommen.

Schreiben Sie eine geeignete main-Methode, um ihr Programm zu testen.

Aufgabe 4: (Vererbung: Spezialisierung)

(Verwenden Sie die vorgegebenen Klassen SimpleChangeCalculator u. Coin von der e-learning Plattform)

An einer anderen (!) Fakultät häufen sich Beschwerden, dass der Eis-Automat in der Cafeteria zwar den richtigen Betrag Wechselgeld zurückgebe, dieser aber oft aus übermäßig vielen kleinen Münzen bestehe. Die Kollegen wenden sich jetzt an uns von der Fakultät Informatik/Wirtschaftsinformatik, da der Hersteller des Eis-Automaten auf Nachfrage folgende Auskunft gegeben hat:

Der bei Ihnen aufgestellte Eis-Automat ist ein Gerät der neusten Generation. Im Gegensatz zu früheren Geräten kann jede Funktion des Automaten mit Hilfe von Java konfiguriert werden. Bei der Geldrückgabe heißt das konkret: wir liefern nur einen Vorschlag für die Geldrückgabelogik aus, die Sie Ihren Bedürfnissen anpassen können!

Unsere Standardimplementierung finden Sie in der Klasse SimpleChangeCalculator. Darin enthalten ist eine Methode getChange, die zu einem vorgegebenen Geldrückgabebetrag die Aufteilung in Münzen errechnet. Dazu wird ein int-Array zurückgeliefert, das die Anzahl der einzelnen Münztypen wie folgt enthält:

An der Position 0: Anzahl der 1-Cent Münzen

An der Position 1: Anzahl der 2-Cent Münzen

An der Position 2: Anzahl der 5-Cent Münzen

An der Position 3: Anzahl der 10-Cent Münzen

An der Position 4: Anzahl der 20-Cent Münzen

An der Position 5: Anzahl der 50-Cent Münzen

An der Position 6: Anzahl der 1-Euro Münzen

An der Position 7: Anzahl der 2-Euro Münzen

Wir sichern Ihnen zu, dass der Rückgabebetrag richtig ist. Für die Stückelung der Münzen haben wir auf einen einfachen und bewährten Algorithmus zurückgegriffen.

Wenn Sie eine veränderte Stückelungslogik realisieren möchten, haben wir dies bereits für Sie vorbereitet: Sie können in einer eigenen Klasse die Methode `getChange()` überschreiben. Natürlich muss weiterhin sichergestellt sein, dass der richtige Betrag zurückgegeben wird. Allerdings können Sie selbst bestimmen, welche Münzen genau zurückgegeben werden.

Sie können sich vorstellen, dass diese Auskunft bei der anderen Fakultät erst einmal große Verwirrung gestiftet hat.

Ihre Aufgabe:

Implementieren Sie einen eigenen `ChangeCalculator` und überschreiben Sie die Methode `getChange` mit einer verbesserten Logik für die Stückelung der Geldrückgabe. Dabei soll jeweils ein `int`-Array mit der **minimalen** Anzahl an Münzen zurückgegeben werden, die den geforderten Betrag ergeben.