

Übungen zur Objektorientierten Programmierung

Aufgabe 1

Programmieren Sie eine Klasse Adresse zum Verwalten von Adressen. Die Klasse sollte folgende Felder besitzen: strasse, hausnummer, ort und plz. Die Datenfelder sollen private sein. Des Weiteren soll die Klasse einen Konstruktor haben, der sich folgendermaßen gestaltet:

Adresse(String strasse, int hausnummer, String ort, int plz)

Darüber hinaus soll die Klasse alle erforderlichen Get- und set-Methoden besitzen. Abschließend implementieren Sie eine Methode drucken(), mit der die Adresse auf der Konsole ausgegeben werden kann. Abschließend überschreiben Sie die toString() – Methode, die eine Zeichenkette wie folgt zurück liefert: **Musterstr;12;Musterhausen;12345**

Aufgabe 2

Erstellen Sie die Klasse Kunde mit folgenden Feldern int kundennummer, String name, String vorname, Adresse adresse, double umsatzBisher. Benutzen für die Adresse die bereits erstellte Klasse. Auch die Klasse Kunde soll einen Konstruktor besitzen. Achten Sie aber darauf, dass jeder neue Kunde eine fortlaufende Kundennummer erhält und der Kundenumsatz mit 0 initialisiert wird. Des Weiteren sorgen Sie dafür, dass die Klasse Kunde auch eine Methode drucken() besitzt, mit der wie auch bei der Klasse Adresse die ausgegeben werden. Des Weiteren soll die Klasse über eine toString()-Methode verfügen, die ebenfalls die Informationen der einzelnen Werte zurück liefert, wie z.B.:

1258;Mustermann;Andreas; Musterstr;12;Musterhausen;12345;0

Aufgabe 3

Erstellen Sie die Klasse Artikel mit folgenden Feldern: String bezeichnung und double preis. Auch hier bitte die entsprechenden get- und set-Methoden implementieren. Die Methoden drucken() und toString() sind nach bekanntem Muster wie bei den Klassen Adresse und Kunde anzulegen.

Aufgabe 4

Erstellen Sie die Klasse Bestellung mit folgenden Feldern: Artikel artikel, int bestellmenge, double Gesamtpreis. Auch hier bitte die entsprechenden get- und set-Methoden implementieren. Achten Sie bitte darauf, dass bei einer Änderung der Bestellmenge der Gesamtpreis der Bestellposition aktualisiert werden muss. Die Methoden drucken() und toString() sind nach bekanntem Muster wie bei den Klassen Adresse und Kunde anzulegen.

Aufgabe 5

Erstellen Sie die Klasse Warenkorb, die folgende Felder enthält: Kunde kunde, Bestellung[] besellliste. Der Konstruktor Warenkorb(Kunde kunde) soll einen neuen Warenkorb mit Kundeninformationen und einem leeren Array mit der Länge 10 erzeugen. Darüber hinaus

besitzt die Klasse Warenkorb die Methoden kaufeArtikel(Bestellung bestellung) und druckeRechnung(). Die Methode kaufeArtikel(...) soll prüfen, ob im Warenkorb der bestellte Artikel bereits vorhanden ist. Ist der Artikel vorhanden, soll die Bestellmenge entsprechend aktualisiert werden. Anderenfalls ist die neue Bestellung im Array an der nächsten freien Stelle zu speichern. Sollte das Array voll sein, ist eine entsprechende Meldung auf der Konsole auszugeben.

Die Methode druckeRechnung() soll aus Kundeninformationen und den einzelnen Bestellungen (Anzahl, Gesamtpreis je Artikel) und Gesamtpreis aller bestellten Artikel bestehen. Abschließend sind die geänderten Kundeninformationen ausgegeben werden.

Aufgabe 6

Testen Sie Ihre Klassen mit einer Main Methode, indem Sie zwei Kunden und mindestens drei Artikel erzeugen. Lassen Sie die Kunden einkaufen und geben Sie die Rechnung aus.

Die Ausgabe einer Rechnung könnte wie folgt aussehen:

```
10001: Mustermann, Andreas  
Musterweg 11 in 12345 Musterhausen  
Aktueller Umsatz: 0.0 €
```

```
Artikelbezeichnung: Hose Artikelpreis: 12.99  
Bestellmenge: 2  
Gesamtpreis Artikel (Hose): 25.98 €  
Artikelbezeichnung: Schuhe Artikelpreis: 32.99  
Bestellmenge: 2  
Gesamtpreis Artikel (Schuhe): 65.98 €  
Artikelbezeichnung: T-Shirt Artikelpreis: 9.99  
Bestellmenge: 1  
Gesamtpreis Artikel (T-Shirt): 9.99 €  
Artikelbezeichnung: Socken Artikelpreis: 3.99  
Bestellmenge: 4  
Gesamtpreis Artikel (Socken): 15.96 €  
Gesamtpreis: 117.91 €
```

```
10001: Mustermann, Andreas  
Musterweg 11 in 12345 Musterhausen  
Aktueller Umsatz: 117.91 €
```