

Ausgabe: 19.01.2017
Abgabetermin: Dienstag, 31.01.2017, 12:00 Uhr

Prof. Norbert Siegmund
Nathalie Dittrich, Hans Lienhop

Abgabe bitte an folgende eMail schicken: hans.lienhop@uni-weimar.de

Aufgabe 1 Enumeration (6 Punkte)

Modellieren Sie eine Eisdielen, indem Sie die Eiskugeln, die dort verkauft werden, mit Hilfe von Enumerationen implementieren. Verwenden Sie für die komplette Aufgabe ein passendes Package für Ihr Team in folgendem Format:

```
de.uniweimar.prog1.team[1-13].eis
```

- Schreiben Sie einen Enumerationstyp für die Eissorten, die verkauft werden: Vanille, Schokolade, Kirsche, Himmelblau und Banane.
- Schreiben Sie eine Klasse `Eiskugel` mit einem Konstruktor, der die Sorte der Kugel entgegennimmt.
- Man kann wählen, ob man das Eis in einer Tüte oder einem Becher haben möchte sowie ob das Eis ohne Sahne, mit Sahne oder mit Schokoladenglasur serviert werden soll. Schreiben Sie für die Verpackungsart (Tüte oder Becher) sowie für die Garnierungsart je einen Enumerationstypen.
- Bei einer Bestellung wählt der Kunde die Anzahl der Kugeln, die Verpackungsart und die Garnierung aus. Schreiben Sie eine Klasse `Eis`, deren Konstruktor die Anzahl der Kugeln sowie Garnierungs- und Verpackungsart übergeben bekommt, und die eine Methode besitzt, um Eiskugeln hinzuzufügen.
- Eine Eiskugel kostet 1 Euro. Die Garnierung mit Sahne kostet 70 Cent und die Garnierung mit Schokolade 50 Cent.
- Schreiben Sie eine Klasse `Eisdielen`, die Methoden besitzt, um ein Array aller Eissorten zurückzugeben, um ein Eis zu bestellen (wobei ein `Eis`-Objekt zurückgegeben wird) sowie um den Preis eines `Eis`-Objekts zu berechnen.

Testen Sie Ihr Programm!

Aufgabe 2 Rekursion (3 Punkte)

Implementieren Sie eine rekursive Methode, welche einen Integer übergeben bekommt und die entsprechende Zahl aus der Fibonacci Reihe zurück gibt. Unter Fibonacci Reihe versteht man eine Reihe an Zahlen, wobei die aktuelle Zahl immer die Summe der beiden vorherigen Zahlen ist.

Also:

```
1 1 2 3 5 8 13 21 ...
```

Hier kann man nochmal nützliches zur Fibonacci Reihe nachlesen: https://en.wikipedia.org/wiki/Fibonacci_number

Aufgabe 3 Static (2 Punkte)

Erstellen Sie eine Klasse `Student`. Denken Sie sich sinnvolle Variablen aus und achten Sie auf die richtige Kapselung. Erweitern Sie Ihre Klasse zusätzlich um zwei sinnvolle statische Methoden und eine statische Variable.

Aufgabe 4 Vererbung (10 Punkte)

Modellieren und implementieren Sie verschiedene Arten von Bankkonten einer Bank. Ein Bankkonto kann von einem bestimmten Typ sein. Es kann ein: Festgeldkonto, ein Konto mit Überziehungszinsen (Dispokredit), ein Konto mit Kreditkarte, ein Konto mit Kreditkarte und Auszahlungslimit sowie ein Konto mit einer Eigenschaft Ihrer Wahl sein. Überlegen Sie sich, welche Attribute und Funktionen alle Bankkontentypen gemeinsam haben und worin die Unterschiede liegen.

- Implementieren Sie für die oben aufgezählten Kontentypen. Entwerfen Sie geeignete Attribute (z.B. Kontoinhaber) und Methoden (z.B. Ein-/Auszahlung) in den jeweiligen Klassen. Die Methoden müssen nicht über einen Rumpf verfügen (d.h., sie müssen, bis auf die Ein- und Auszahlungsmethoden nicht implementiert sein). Versuchen Sie eine größt mögliche Wiederverwendung durch den Einsatz von Vererbung zu erreichen.

- (b) Implementieren Sie eine Methode, die den jetzigen Kontostand ausgibt sowie den Kontostand nach der nächsten Zinsperiode. Beachten Sie, dass negative Dispozinsen, wenige Zinsen (Standardkonto), mittlere Zinsen (Kreditkartenkonto) und hohe Zinsen (Festgeldkonto) je nach Kontotyp anfallen.
- (c) Implementieren Sie ein Klasse `Test`, die in der Main-Methode verschiedene Konten anlegt, Ein-/Auszahlungen macht und die Zinsen für die nächste Zinsperiode berechnet.

Aufgabe 5 Generics (5 Punkte)

Implementieren Sie eine (optional: abstrakte) Klasse `Eis` sowie Kindklassen für verschiedene Eissorten. Denken Sie sich passende Attribute und Methoden aus. Implementieren Sie eine generische Klasse `Eistüte`, die eine Methode zum Hinzufügen für Eissorten hat sowie eine Methode, die den Inhalt der Eistüte (alle hinzugefügten Eissorten) ausgibt. Schreiben Sie eine Klasse `Test`, welche in der Main-Methode eine Eistüte anlegt und verschiedene Eissorten der Eistüte hinzufügt und dann die Ausgabe hierzu macht. Beachten Sie: Prüfen Sie, dass nur Objekte von Typ `Eis` bzw. deren Subtypen zur Eistüte hinzugefügt werden können.