



Informatik I – Eprog HS13

Übung 8

1 Aufgabe: Interface: ISize

1.1 Lernziele

1. Sie können eine Schnittstelle definieren und diese in Klassen implementieren.
2. Sie kennen die Unterschiede zwischen Interfaces und abstrakten Klassen.

1.2 Fläche

1. Erstellen Sie eine Klasse `Car`, welche die Attribute `Model`, `Hersteller` und `Gewicht` besitzt und eine Klasse `SoccerField`, welche durch die Attribute `Ortschaft` und `Heimteam` charakterisiert wird. Zu allen Instanzvariablen sollen accessor-Methoden existieren.
2. Die soeben erstellten Klassen sollen nun in einem weiteren Schritt so erweitert werden, dass sie über eine gemeinsame Schnittstelle verfügen, über welche die Länge, die Breite und die Höhe von Autos und Fussballfeldern abgerufen werden kann. Erstellen Sie dazu ein Interface namens `ISize`, welches die benötigten Methoden anbietet.
3. Erweitern Sie die Klassen `Car` und `SoccerField` um das Interface `ISize`. Nehmen Sie die dazu notwendigen Erweiterungen an den Klassen vor, damit die Grösse von Objekten über die neue Schnittstelle ermittelt werden kann. Stellen Sie je einen Konstruktor zur Verfügung, welcher die dafür benötigten Daten erwartet.
4. Erstellen Sie im `TestDriver` 4 unterschiedliche Objekte beider Klassen und legen Sie diese in einem typsicheren `Array` ab. Ergänzen Sie ausserdem den `TestDriver` um eine Klassenmethode `getSquare()`, welche die Grundfläche der Objekte einer übergebenen Liste berechnet und diese der Reihe nach auf der Konsole ausgibt.
5. Überlegen Sie sich, weshalb wir für die Schnittstelle `ISize` ein Interface und keine abstrakte Klasse verwendet haben. Notieren Sie Ihre Überlegungen und nennen Sie alle Ihnen bekannten Unterschiede zwischen Interfaces und abstrakten Klassen.

2 Aufgabe: Interface: 3D Körper

2.1 Lernziele

1. Sie können eine gemeinsame Schnittstelle schreiben und andere Klassen diese implementieren lassen.
2. Sie verstehen anhand einem ersten Beispiel die Vorteile einer gemeinsamen Schnittstelle.

2.2 Dreidimensionale Körper

a) Geometrische Figuren

Modellieren Sie die folgenden geometrischen Figuren jeweils in einer eigenen Klasse. Die Implementierung soll dabei nur so detailliert sein, dass für jede Figur das Volumen abgefragt werden kann.

- Quader
- Reguläre Pyramide
- Gerader Kreiszylinder
- Kugel

Falls Ihnen die einzelnen Voluminaberechnungen nicht geläufig sind, hilft [diese Auflistung](#) weiter.

b) Volumina

Erweitern Sie die Klassen aus a) dahingehend, dass es möglich ist, unterschiedliche Instanzen in einem geeigneten Array abzulegen und in einer Schleife alle Volumina auszugeben.

3 Aufgabe: Repetition: Referenztypen

```
1 public class Number {
2     private int z;
3
4     public void setZahl(int x) {
5         z = x;
6     }
7
8     public int getZahl() {
9         return z;
10    }
11
12    public void change(Number x, Number y) {
13        x.setZahl(50);
14        y = x;
15    }
16
17 }
```

Listing 1: Number.java

```
1 public class NumberTest {
2
3     public static void main(String[] args) {
4         Number a = new Number();
5         Number b = new Number();
6
7         a.setZahl(0);
8         b.setZahl(100);
9
10        a.change(a, b);
11
12        System.out.println(a.getZahl());
13        System.out.println(b.getZahl());
14    }
15
16 }
```

Listing 2: NumberTest.java

Welche Ausgabe erfolgt auf der Konsole beim Ausführen dieses Programms? Begründen Sie Ihre Antwort stichwortartig oder in einigen wenigen Sätzen.
(Das Komma repräsentiert dabei eine neue Zeile.)

- 100, 50
- 50, 50
- 50, 100
- 0, 100

4 Aufgabe: Eine Adressbuch-Applikation - Teil 2

4.1 Lernziele

1. Sie sind in der Lage, eine komplett funktionsfähige Adressverwaltung der letzten Woche zu ergänzen und optimieren.
2. Sie können eine Benutzerschnittstelle implementieren.

4.2 Aufgabenstellung

In der letzten Übung (Woche 7) haben Sie einige Grundfunktionalitäten eines klassischen Adressbuchs programmiert. In dieser Aufgabe werden Sie einige Anpassungen am Code vornehmen, um die Applikation zu optimieren. Verwenden Sie für diese Aufgabe Ihre Lösung von letzter Woche. Vergleichen Sie diese zunächst mit der Musterlösung, um Folgefehler zu vermeiden.

a) Interface: Benutzerschnittstelle

In dieser Woche soll ein Benutzer text-basiert, also schlicht über die Kommandozeile, neue Kontakte anlegen können etc. Später werden wir eine graphische Benutzeroberfläche zu diesem Zweck implementieren. Aus diesem Grund ist es wichtig, dass die Benutzerschnittstelle zwar das Adressbuch kennt, letzteres aber umgekehrt keine Referenz auf die Benutzerschnittstelle halten darf. Wenn Sie dies beachten, können Sie später, ohne bestehenden Code zu modifizieren, Ihre Applikation erweitern.

1. Schreiben Sie eine Klasse `CommandLineInterface`, welche eine Instanz des Adressbuchs im Konstruktor erwartet und in einer Instanzvariablen speichert.
2. Die Klasse soll eine Methode `void run()` besitzen, die mittels eines `java.util.Scanner` Objektes Eingaben vom Benutzer entgegen nimmt.
3. Beim Aufrufen von `run()` soll dem Benutzer ein Menu präsentiert und seine Auswahl gespeichert werden:

```
1 Make your choice:
2 1. Add new contact.
3 2. Select existing contact.
4 3. Delete current selection.
5 4. List all contacts.
6 5. Quit the application.
```

4. Wenn der Benutzer '1' eingibt, dann soll er erst nach dem Vor-, dann nach dem Nachnamen des Benutzers gefragt werden. Die Eingaben sollen dann verwendet werden, um die `addContact(String firstName, String lastName)`-Methode des Adressbuches (siehe oben) aufzurufen.
5. Wenn der Benutzer '2' eingibt, dann soll eine durchnummerierte Liste aller Kontakte angezeigt werden. Für die Nummerierung sollen Sie die eindeutige Nummer eines jeden Kontaktes hernehmen. Anschliessend kann der Benutzer eine der Nummern eingeben und dann mit '3' *Delete the current selection.* auswählen. Dies soll bewirken, dass der entsprechende Kontakt aus dem Adressbuch gelöscht wird. Beispiel:

```
1 Make your choice:
2 1. Add new contact.
3 2. Select existing contact.
```

```

4      3. Delete current selection.
5      4. List all contacts.
6      5. Quit the application.
7      >2
8
9
10     Select one of the following contacts by number:
11     ---
12     1. Michael Wuersch
13     ---
14     2. Emanuel Giger
15     ---
16     Which contact do you want to select?
17     >1
18
19
20     Make your choice:
21     1. Add new contact.
22     2. Select existing contact.
23     3. Delete current selection.
24     4. List all contacts.
25     5. Quit the application.
26     >3
27
28     Deleted 'Michael Wuersch'.
29
30
31     Make your choice:
32     ...

```

6. Die Auswahl '4' und '5' sollte selbsterklärend sein: '4' gibt sämtliche Kontakte auf die Kommandozeile aus, während '5' das Adressbuch beendet.