



Programmierung für Mathematik (HS13)

Übung 9

1 Aufgabe: Repetition: Vererbung

1.1 Lernziele

1. Vererbung erkennen und begründen können

1.2 Ausgangslage

Implementieren Sie die zwei Klassen `Square` und `Rectangle`. Diese zwei Klassen sollen die Methoden `perimeter()` und `area()` besitzen. Die Klassen sollen nebst den beiden Methoden auch einen passenden Konstruktor implementieren. Argumentieren Sie, ob dabei die Klasse `Rectangle` von der Klasse `Square` erben soll oder umgekehrt.

a) Aufgabenstellung

Implementieren Sie obigen Sachverhalt und testen Sie diese in einem `TestDriver`.

2 Aufgabe: Chain of Words

2.1 Lernziele

1. Das Wissen über Referenzen vertiefen.
2. Die Verwendung von Schleifen repetieren, insbesondere den Unterschied zwischen `while`- und `for`-Schleifen vertiefen.
3. `String`-Operationen und Konstruktoren wiederholen.

a) Aufgabenstellung: Die Elemente

1. Erstellen Sie eine Klasse namens `WordElement`.
2. Definieren Sie einen Konstruktor, welcher einen `String` als Parameter erwartet.
3. Deklarieren Sie nun eine private Instanzvariable des Typs `WordElement` und benennen sie diese mit `nextElement`.
4. Machen Sie dasselbe für eine zweite Variable namens `previousElement`.
5. Implementieren Sie Getter- und Setter-Methoden.
6. Fügen Sie desweiteren zwei Methoden hinzu, welche überprüfen, ob die jeweiligen Instanzvariablen mit einem Wert initialisiert wurden.

b) Aufgabenstellung: Die Kette

Im vorherigen Teil haben wir den Grundstein für unsere Kette gelegt: Die einzelnen Elemente verfügen über die Methoden, um sich aneinander zu ketten.

1. Erstellen Sie eine Klasse namens `Chain`.
2. Definieren Sie für diese Klasse eine private Instanzvariable namens `firstElement`.
3. Desweiteren muss eine Methode `addWord(String word)` definiert werden, welche keinen Wert zurückliefert. Diese Methode erstellt bei Aufruf jeweils eine neue Instanz der Klasse `WordElement` und übergibt dieser gleich das erhaltene Wort. Ist `firstElement == null`, so wird diese Variable mit dem neuen Objekt initialisiert. Ist dies nicht der Fall, so muss nun über eine `while`-Schleife das letzte Element der Wortkette ermittelt werden und danach das Objekt angefügt werden.
4. Erweitern Sie die Klasse dahingehend, dass eine Methode namens `getWordCount()` die Anzahl Wörter der Kette zurückgibt.
5. Ebenfalls muss eine Methode namens `getSentence()` erstellt werden, welche die gesamte Kette als `String` (mit Leerzeichen zwischen den Wörtern!) zurückgibt.
6. Erstellen Sie nun eine Methode `getReverse()`, welche bei folgendem Input:

```
1 Chain myChain = new Chain();
2 myChain.addWord("Anna");
3 myChain.addWord("liebt");
4 myChain.addWord("Florian");
```

diesen Output in Form eines Strings zurückliefert: `Thomas liebt Anna`.

7. Erstellen Sie eine Methode namens `getEverySecond()` welche zuerst eine lokale Variable mit der aktuellen Anzahl Wörter initialisiert und danach mit einer `for`-Schleife jedes zweite Wort zu einem String zusammenhängt. z.B. Die Wörter `Ich lebe und arbeite in der Schweiz` sollten wie folgt ausgegeben werden: `lebe arbeite der`.
8. Testen Sie die Funktionalität in einem `TestDriver`.

3 Aufgabe: Interface: ISize

3.1 Lernziele

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

3.2 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 `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.