

Programmieren 1

Selbststudium Semesterwoche 6

Kapitel 4.1 bis 4.3

1.

zu bearbeitende Aufgaben: 4.1

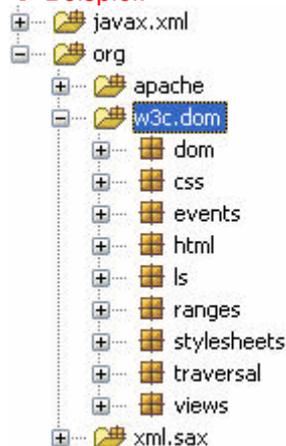
2.

Was verstehen Sie unter einem "Java-Package"?

→ Java-Programme und APIs bestehen aus Ansammlungen von vielen einzelnen Dateien, die innerhalb der objektorientierten Programmierung jeweils eine Klasse repräsentieren. Um dieser Klassensammlung eine Struktur geben zu können und so die Les- und Benutzbarkeit für Menschen zu verbessern, wurden Java-Packages eingeführt.

→ Dabei handelt es sich um eine hierarchische Struktur, in der die Klassen (= Dateien) abgelegt werden. Jedes Package hat eine oberste Ebene, in der beliebig viele Klassen und weitere Ebenen liegen können. Für jede untergeordnete Ebene gilt wiederum das gleiche. So entstehen Pakete, die Baumstrukturen abbilden und Klassen sinnvoll adressierbar machen.

→ Beispiel:



3.

Sie möchten in Ihrem Source-Code die Bibliotheksklasse ArrayList verwenden. Mit welcher Anweisung können Sie die Bibliotheksklasse ArrayList verfügbar machen?

→ In der Dokumentation von Java (<http://java.sun.com/j2se/1.4.2/docs/api/java/util/ArrayList.html>) findet man für jede Klasse eine Definition.

→ Die Klasse ArrayList hat den Package-Pfad java.util.ArrayList. Mit folgendem Befehl wird das Package im Code hinzugefügt:

```
import java.util.*;
```

4.

Machen Sie sich mit Hilfe der JDK Dokumentation etwas schlau über die Klasse ArrayList. Welche Methoden erachten Sie als besonders wichtig/interessant?

→ Die wichtigsten Methoden wurden aus der Dokumentation kopiert:

void	add (int index, Object element) Inserts the specified element at the specified position in this list.
boolean	add (Object o) Appends the specified element to the end of this list.
void	clear () Removes all of the elements from this list.
Object	get (int index) Returns the element at the specified position in this list.
int	lastIndexOf (Object elem) Returns the index of the last occurrence of the specified object in this list.
Object	remove (int index) Removes the element at the specified position in this list.
int	size () Returns the number of elements in this list.

→ Weitere Informationen: <http://java.sun.com/j2se/1.4.2/docs/api/java/util/ArrayList.html>

Kapitel 4.4 bis 4.7

5.

zu bearbeitende Aufgaben: 4.2 bis 4.9

6.

Erläutern Sie nachfolgende Deklaration:

```
private ArrayList<Balloon> list = new ArrayList<Balloon>();
```

→ In dieser Zeile wird ein neues Objekt von der Klasse "ArrayList" instanziiert und in die Variable "list" gespeichert. Die Arrayfelder des Arrays "list" sind vom Datentyp "Balloon".

7.

Inwiefern haben Abstraktion und ArrayList miteinander zu tun?

→ ArrayList verbergen sehr viel abstrakten Code und stellen stellvertretend dafür einfach zu benutzende Methoden zur Verfügung.

8.

Inwiefern unterscheiden sich die Methoden remove() und get() von ArrayList?

→ remove(int index) entfernt das Arrayelement einer bestimmten Position, die über den Parameter "index" gewählt wird. Rückgabewert ist das Objekt des entfernten Arrayelements.

→ get(int index) gibt ein Arrayelement einer bestimmten Position zurück, die über den Parameter "index" gewählt wird. Rückgabewert ist das abgefragte Arrayelement.

Kapitel 4.8. und 4.9

9.

zu bearbeitende Aufgaben: 4.12 bis 4.17, 4.20, 4.23

10.

Erläutern Sie möglichst ausführlich den Source-Code auf Seite 91 oben.

11.

Kann es passieren, dass der Schleifen-Rumpf einer while-Schleife gar nie zur Ausführung gelangt?

→ Ja, wenn die Schleife `while(false)` programmiert wird.→ Mit `Break;` und `Continue;` kann konditionell gestoppt bzw. fortgefahren werden.

```
while (<condition>)
{
    <statements>
    [break;]

    <statements>
    [continue;]
}
```

12.

Geben Sie für `no++` zwei alternative Anweisungen an.→ `no = no+1;`→ `no = no++;`→ `no += 1;`

13.

Eine ArrayList lässt sich mit Hilfe einer foreach-Schleife traversieren. Kennen Sie noch weitere Möglichkeiten?

→ for-Schleife

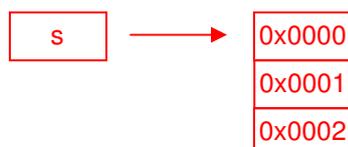
→ while-Schleife

→ do-while-Schleife

14.

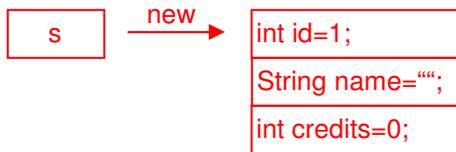
Ist `hasNext()` eine Methode von ArrayList oder Iterator? Wie ist der Rückgabewert von `hasNext()` zu interpretieren?→ `hasNext()` ist eine Methode von Iterator. Der Rückgabewert ist `boolean`, d.h. die Methode gibt zurück, ob der Array ein nächstes Element hat (`hasNext()=true`) oder ob das jetzige Element das letzte ist im Array (`hasNext()=false`).→ Weitere Information: <http://java.sun.com/j2se/1.4.2/docs/api/java/util/Iterator.html>**Kapitel 4.10**

15.

Eine Variable, deklariert für einen Klassentyp (Eine solche Variable wird auch Referenzvariable genannt.), kann den Spezialwert `null` beinhalten. Veranschaulichen Sie diesen Fall mit einer Zeichnung (Fall1). Wie sieht es aus, wenn die Variable ein Objekt speichert (Fall2)?Fall 1: Objekt nicht initialisiert (`private Student s = null;`)→ Die Variable `s` enthält lediglich eine Referenzadresse auf einen Speicherbereich im RAM. Der für die Instanzierung benötigte Arbeitsspeicher wird reserviert.

Fall 2: Objekt wurde initialisiert (`private Student s = new Student();`)

→ Die Variable `s` enthält noch immer nur eine Referenzadresse auf einen Speicherbereich im RAM. Im Arbeitsspeicher wurde jedoch ein konkretes Objekt angelegt.



Kapitel 4.12

16.

zu bearbeitende Aufgaben: 4.39 bis 4.43

17.

Was spricht für und was spricht gegen Arrays?

→ Vorteile:

- Kann mehrere Variablen vom selben Datentyp mit wenig Programmieraufwand speichern.
- Schneller Zugriff auf eine grosse Auswahl von Daten.

→ Nachteile:

- Ein Verschieben der Arrayelemente (z.B. Sortierung) führt nur über einen aufwendigen Kopiervorgang.
- Die Arraygrösse wird beim Erstellen festgelegt. Ein Array kann also "voll" sein und muss erweitert werden.
- Grosse Arrays belegen viel Speicherplatz im RAM.

18.

Wie lässt sich die Länge eines Arrays ermitteln?

→ `ArrayList` stellt dafür die Methode `"size()"` zur Verfügung. Diese liefert als Rückgabewert eine Integerzahl, welche die Anzahl Arrayelemente enthält.