

Angabe

zu Schreiben ist ein Programm, welches die Gesamtgrösse eines Verzeichnisses berechnet. Beim Aufruf wird dem Programm per Kommandozeilen-Parameter der Name desjenigen Verzeichnisses übergeben, für das die Grösse berechnet werden soll.

zum Beispiel:

```
c:\java> java DirSizeDemo c:\irgendwas
```

sollte dem Programm beim aufruf fälschlicherweise der geforderte Kommandozeilen-Parameter nicht übergeben worden sein, soll das Programm eine Fehlermeldung an die Konsole schreiben und beenden:

```
c:\java> java DirSizeDemo
Usage: java DirSizeDemo [-s] Path
c:\java>
```

Hier seht ihr ein Beispiel für eine mögliche Ausgabe:

```
c:\java> java DirSizeDemo c:\irgendwas
10 c:\irgendwas\blah.txt
100 c:\irgendwas\bleh.txt
52 c:\irgendwas\nochwas\muh.txt
14 c:\irgendwas\nochwas\undnoch\meh.txt
176 bytes
c:\java>
```

Das Programm listet offenbar alle gefundenen Dateien und deren jeweilige Grösse in Byte auf. Am Schluss wird die Summe und damit die Gesamtgrösse des Verzeichnisses ausgegeben. Beachtet, dass das Programm nicht nur diejenigen Dateien beachtet, die direkt im angegebenen Verzeichnis liegen, sondern auch Dateien in Unterverzeichnissen findet.

FREIWILLIG ZU IMPLEMENTIEREN:

Wird beim Aufruf des Programms der Optionale Parameter -s (für "summarize") mit angegeben, ist nur die Summe am Schluss auszugeben, die Auflistung jeder einzelnen Datei entfällt:

```
c:\java> java DirSizeDemo -s c:\irgendwas
176 bytes
c:\java>
```

Tipps

Das gesamte Programm mit allen Features in *einer* Hauruck-Aktion zu schreiben, ist sicher recht schwierig. Besser ist es vermutlich, wenn ihr euch schrittweise vortastet, und zuerst mal Teilerfolge anstrebt.

Versucht vielleicht der Reihe nach mal Folgendes:

- Draufkommen ob ein Kommandozeilen-Parameter übergeben wurde oder nicht. Falls nicht, Ausgeben der Fehlermeldung und Programm beenden. Ansonsten: Weiter im Programm.

(Das haben wir eh schon im Unterricht gemacht).

- Unter der Annahme, dass der übergebene Kommandozeilen-Parameter der volle (absolute) Pfad zu einer Datei war: Ausgabe der Grösse von nur dieser einen Datei.
Hier braucht ihr zum ersten Mal die Klasse `File` aus dem package `java.io`. (Import-Statement nicht vergessen!: `import java.io.File;`) Verwendet deren Konstruktor `File(String pathname)`, die Methode `long length()` und `String getAbsolutePathName()`.

```
c:\java> java DirSizeDemo c:\irgendwas\bleh.txt
c:\irgendwas\blah.txt ist 10 bytes gross.
```

- Unter der Annahme, dass der übergebene Kommandozeilen-Parameter der Pfad zu einem Verzeichnis war: Ausgabe der Grösse aller Dateien und Verzeichnisse, die direkt im angegebenen Verzeichnis liegen (noch keine Beachtung von Unterverzeichnissen).

```
c:\java> java DirSizeDemo c:\irgendwas
10 c:\irgendwas\blah.txt
100 c:\irgendwas\bleh.txt
0 c:\irgendwas\nochwas
```

Damit seid ihr schon mal recht weit.

Um die Geschichte mit den Unterverzeichnissen auch noch hinzubekommen, braucht ihr Rekursion. Lest euch dazu im Buch die Seiten 255 bis 258 durch, und versucht den Grundgedanken hinter der Rekursion zu verstehen. (Ihr werdet dabei vermutlich nicht ganz so grosse Sternderl in den Augen bekommen wie ich, aber hey... Rekursion ist schon was sehr Elegantes, ehrlich. Das isst schon wert, sich damit zu beschäftigen.)

Rekursion lässt sich umsetzen zum Beispiel mit Methoden, die sich selbst aufrufen. In unserem Fall macht es Sinn, eine Methode `m` zu schreiben, die uns von einer Instanz der Klasse `File` die Grösse berechnet und ausgibt. Notabene: Einer Instanz der Klasse `File` kann ein Verzeichnis oder eine Datei repräsentieren. Siehe dazu die Methoden `boolean isDirectory()` und `boolean isFile()`.

Für den Fall, dass es sich bei der Instanz um die Repräsentation einer Datei handelt, dann ist das ganz einfach: mit der Methode `length()` lässt sich die Grösse ja leicht feststellen.

Handelt es sich aber um ein Verzeichnis `x`, dann führt `length()` nicht zum Ziel, das liefert immer 0 (steht so in der Dokumentation und haben wir auch oben gesehen). *Der Trick ist jetzt, dass wir zur Feststellung der Gesamtgrösse dieses Verzeichnisses `x` uns einfach dessen Inhalt besorgen und für jedes Element wieder die Methode `m` (aus `m` heraus!) aufrufen und die dadurch berechneten Datei- und Verzeichnisgrößen zusammenzählen.* Das ist dann die Gesamtgrösse des Verzeichnisses `x`.

Hier ein bisschen Code dazu:

```
class DirSize {
    long getRecursiveSize(File file) {
        long length = 0;
        if (file.isDirectory()) {
            // - Liste der Dateien und Verzeichnisse holen
            // - für jedes Element davon getRecursiveSize() aufrufen
            // - Gesamtgrösse in length aufsummieren
```

```

    } else {
        length = file.length();
    }
    return length;
}

class DirSizeDemo {
    public static void main(String[] args) {
        String pathname = args[0];
        File file = new File(pathname);
        DirSize dirSize = new DirSize();
        long length = dirSize.getRecursiveSize(file);
        System.out.println(length + " bytes");
    }
}

```

Damit geb ich eh schon ne Menge her eigentlich... ;-)

(wenn ihr euch fragt, warum ich eine 2. Klasse `DirSize` einführe und nicht einfach die Methode `getRecursiveSize(...)` in der Klasse `DirSizeDemo` implementiere: fragt euch nicht. Da reden wir drüber, wenn wir über statische Methoden und Klassenvariablen sprechen. Betrachtet es inzwischen bitt einfach als notwendig.)

viel Erfolg!

lg

Fritz