

Aufgabenblatt 7

Klasse `java.io.File`

Die Java-API bietet eine Klasse `File` an, die es erlaubt Verzeichnisse mit Dateien und Unterverzeichnissen zu durchlaufen. Machen Sie sich mit der Klasse vertraut. Beiliegend finden Sie eine Klasse `FileBrowser`, die eine kleine Anwendung mit der Klasse `File` zeigt.

Klasse `FileBrowser` – Version 1

Die Klasse `FileBrowser` soll um zwei Methoden erweitert werden:

- `void dirPrint(File file)`
Gibt alle Unterverzeichnisse und Dateien von `file` rekursiv aus.
Dabei soll Datei- bzw. Verzeichnisname, Pfadname und Größe der Datei ausgegeben werden.
- `void dirSearch(File file, String extension, List<String> list)`
durchsucht alle Unterverzeichnisse von `file` rekursiv nach Dateien mit der Endung `extension` und schreibt ihre Namen mit ihrem Pfad in die Liste `list`. Hierbei kann die Methode `String.endsWith()` hilfreich sein.

Beispielsweise fügt der Aufruf

```
File dir = new File("/Users/oliverbittel/NetBeansProjects");  
dirSearch(dir, ".java", list)
```

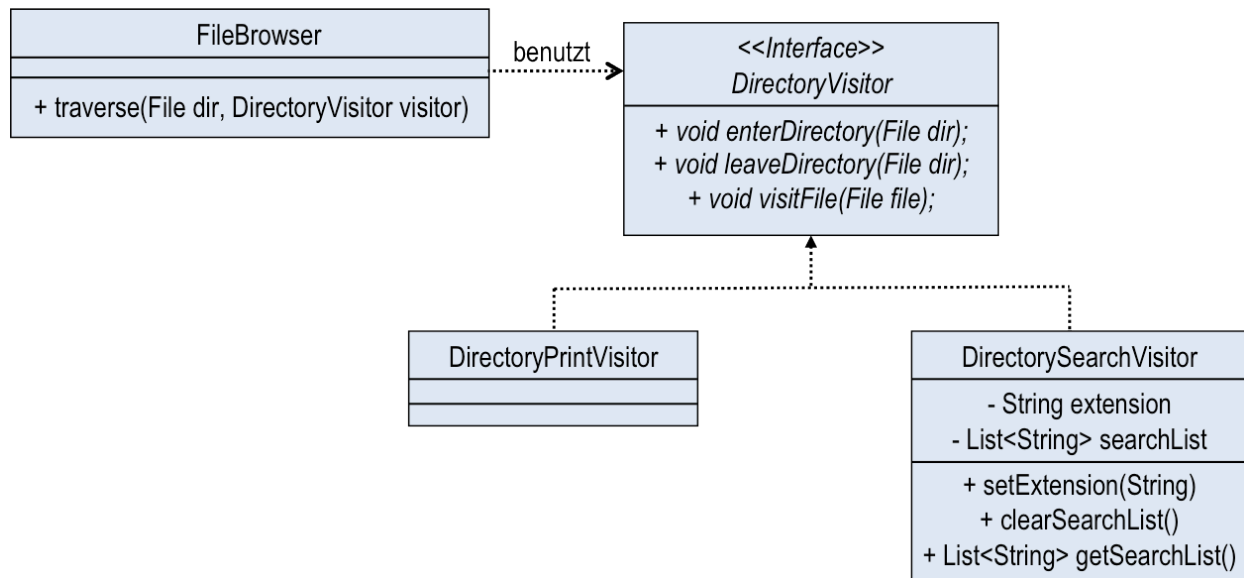
alle Java-Dateinamen des Verzeichnis `dir` in die Liste `list` ein.

- Schreiben Sie eine `main`-Methode, die beide Methoden ausgiebig testet.

Klasse `FileBrowser` – Version 2

Wenn Sie die Methoden `dirPrint` und `dirSearch` implementiert haben, werden Sie feststellen, dass in beiden Methoden in der gleichen Weise über die Verzeichnisstruktur traversiert wird. Mit dem Besucher-Entwurfsmuster (visitor pattern) lässt sich diese Codeduplizierung vermeiden. Setzen Sie einfach das Klassendiagramm auf der folgenden Seite in Java um.

Die Methode `traverse(dir, visitor)` durchläuft die Verzeichnisstruktur `dir` rekursiv. Beim Betreten bzw. Verlassen eines Verzeichnisses wird `enterDirectory` bzw. `leaveDirectory` aufgerufen. Stößt die Traversierung auf eine Datei, dann wird `visitFile` aufgerufen.



DirectoryPrintVisitor implementiert die Methoden enterDirectory, indem die Einrückungstiefe hoch gesetzt und der Verzeichnisname ausgegeben wird. leaveDirectory setzt die Einrückungstiefe eine Stufe zurück. visitFile gibt den Dateinamen aus. Um ein Verzeichnis rekursiv auszugeben (wie bei dirPrint), ist dann folgender Programm-Code durchzuführen:

```
File dir = new File("/Users/oliverbittel/NetBeansProjects");
traverse(dir, new DirectoryPrintVisitor());
```

DirectorySearchVisitor lässt die Implementierung der Methoden enterDirectory und leaveDirectory leer. Bei visitFile(file) wird die Datei file, falls ihr Name mit der Endung extension endet, in die Liste searchList abgespeichert.

Mit setExtension kann die gewünschte Endung eingestellt werden. Mit clearSearchList kann die Suchliste searchList zurückgesetzt und mit getSearchList gelesen werden.

Um ein Verzeichnis rekursiv nach Dateien mit der Endung „.java“ zu durchsuchen (wie bei dirSearch), ist dann folgender Programm-Code durchzuführen:

```
File dir = new File("/Users/oliverbittel/NetBeansProjects");
DirectorySearchVisitor v = new DirectorySearchVisitor();
v.setExtension(".java");
traverse(dir, v);
List<String> list = v.getSearchList();
```

Abgabe:

Der erstellte Code muss vorgeführt und erklärt werden können.