

EPROG 2.Teilprüfung

Dauer 50min. Keine Unterlagen erlaubt. Loginname ist q<matrikelnummer> (also z.B. q0697801). Passwort ist Ihre Matrikelnummer.

Speichern Sie Ihre Lösungen in den dafür vorgesehenen Unterverzeichnissen ~/test/Beispiel1/ und ~/test/Beispiel2/.

Die Java-Dokumentation finden Sie im Verzeichnis ~/test/doc/

Aufgabe 1:

Vervollständigen Sie die Datei: `test/Beispiel1/String2.java`

Schreiben Sie dort die Methode

```
public double compress(),
```

die eine Lauflängenkompression auf dem String `internal` durchführt und das Ergebnis wieder in `internal` speichert. Dabei werden Zeichen die mehr als einmal nacheinander vorkommen als Anzahl der Vorkommen gefolgt vom Zeichen dargestellt (Bsp: A -> A, AAA -> 3A). Im Eingabestring dürfen dabei keine Zahlen vorkommen. Diese Vorbedingung muss nicht überprüft werden. Nach der Kompression wird er Kompressionsgrad, also das Verhältnis der Länge des unkomprimierten zur Länge des komprimierten Strings, zurückgegeben werden.

Beispiel: "abcccd" wird zu "ab3c2d".

Hinweis:

Die vorgefertigte Datei `Main.java` können Sie zum Testen Ihrer Klassen benutzen. Ihr Inhalt wird nicht beurteilt.

Aufgabe 2:

Erstellen sie ein einfaches Filesystem in dem Verzeichnisse und Textdateien verwaltet werden. Das Filesystem besteht aus dem Interface `DirectoryEntry` und den Klassen `File` und `Directory`. Erstellen Sie die Dateien `DirectoryEntry.java`, `File.java` und `Directory.java`, die die Interface- bzw. Klassendefinitionen beinhalten.

DirectoryEntry:

Dieses Interface `DirectoryEntry` ist das Basisinterface für alle Objekte im Dateisystem. Es soll die Methoden

```
String name()
```

und

```
int size()
```

und

```
String description()
```

zur Verfügung stellen.

File:

Die Klasse `File` implementiert das Interface `DirectoryEntry` und speichert zusätzlich einen Dateinhalt in Form eines Strings. Auf diesen kann mittels der Auskunftsmethode (Getter)

```
String getContent()
```

und der Methode (Setter)

```
void setContent(String)
```

zugegriffen werden.

Die Methode `size()` soll hier die Länge des momentan gespeicherten Strings (Dateinhalts) zurückliefern. Die Methoden `name()` und `description()` liefern bei `File` beide den Namen des Files (Datei) zurück. Erstellen Sie auch einen Konstruktor, der einen Parameter zum Setzen des Namens des Files besitzt.

Directory:

Die Klasse `Directory` implementiert ebenfalls das Interface `DirectoryEntry` und verwaltet(speichert) eine Liste von mit `DirectoryEntry` kompatiblen Objekten (Dateien oder Verzeichnisse).

Die Methode `size()` soll in diesem Fall die Anzahl der Einträge im Verzeichnis zurückgeben. Die Methode

```
void add(DirectoryEntry e)
```

fügt dem Verzeichnis einen `DirectoryEntry` hinzu. Die Methode

```
void list()
```

soll eine Inhaltsangabe des Verzeichnisses ausgeben. Die Methode `name()` liefert den Namen des Verzeichnisses. Die Methode `description()` soll bei `Directory` den Namen des Verzeichnisses in spitzen Klammern (zb: "`<root>`") zurückliefern. Erstellen Sie auch einen Konstruktor, der einen Parameter zum Setzen des Namens des Verzeichnisses besitzt. Siehe dazu das Beispiel in der Datei `Main.java`.

Hinweis:

Die vorgefertigte Datei `Main.java` können Sie zum Testen Ihrer Klassen benutzen. Ihr Inhalt wird nicht beurteilt.