



Abgabe: Mo, 14.01.08 zentrale Abgabe über <https://grundstudium.in.tum.de/info1abgabe>

Praktikum Grundlagen der Programmierung

Aufgabe 46 (Ü) IO und Exceptions

Ziel dieser Aufgabe ist es, eine statische Methode `int berechneWert(String datei)` zu schreiben, die den Wert eines in einer Datei gespeicherten arithmetischen Ausdrucks berechnet. Dabei beschränken wir uns auf arithmetische Ausdrücke, in denen lediglich **natürliche Zahlen** und **Addition** auftreten. Als Beispiel soll ein Aufruf der geforderten Methode für den Ausdruck `21 + 2 + 19` die Zahl 42 zurück liefern.

Dabei ist folgendes zu beachten:

- Ist das Format der Datei falsch, so soll bei einem Aufruf der Methode `berechneWert()` eine `FalschesFormatException` geworfen werden, die die Information enthält, beim Lesen des **wie vielen Wortes** ein Fehler aufgetreten ist.
- Existiert keine Datei namens `datei`, so soll eine `DateiNichtVorhandenException` geworfen werden.

Zur Lösung dieser Aufgabe können Sie die nachfolgend beschriebene Klasse `DateiLeser` verwenden, mit der eine Datei Wort für Wort gelesen werden kann. Die Klasse `DateiLeser` ist gegeben und muss **nicht** von Ihnen implementiert werden.

- Der Konstruktor `DateiLeser(String datei)` erzeugt ein `DateiLeser`-Objekt für die Datei `datei`. Existiert eine solche Datei nicht, so wird eine `DateiNichtVorhandenException` geworfen.
- Um Wort für Wort lesen zu können, wird die Methode `String leseNaechstesWort()` zur Verfügung gestellt. Diese liefert beim i -ten Aufruf das i -te Wort als String zurück. Existiert kein i -tes Wort, so wird `null` zurückgeliefert.

- a) Definieren Sie die oben beschriebene `FalschesFormatException`.
- b) Definieren Sie die oben beschriebene statische Methode `berechneWert()`.

Hinweis: Die statische Methode `int parseInt(String s)` der Klasse `Integer` liefert den `int`-Wert der als String repräsentierten Zahl `s` zurück. Falls der übergebene String `s` keinen `int`-Wert repräsentiert, wird eine `NumberFormatException` geworfen.

Aufgabe 47 (H) Terminkalender

(8 Punkte)

Ziel dieser Aufgabe ist es, einen Terminkalender zu programmieren, bei dessen Implementierung Java-Exceptions möglichst sinnvoll eingesetzt werden sollen. Wichtigste Klassen in Ihrem Klassenmodell sollten die Klassen `Termin` und `Kalender` sein.

Ein Termin hat einen Startzeitpunkt und eine Dauer. Der Startzeitpunkt und die Dauer sollen intern als `int`-Werte repräsentiert werden. Der Startzeitpunkt bezeichnet dabei die Minute seit dem 01.01.1970 00:00 Uhr. Weiterhin sind einem Termin eine Beschreibung und ein Ort zugeordnet. Die Klasse `Termin` soll folgende Konstruktoren und Methoden bereit stellen:

- `Termin(int start, int dauer, String beschr, String ort)`. Wird eine negative Dauer übergeben, so soll eine Exception ausgelöst werden.

- `Termin(String start, int dauer, String beschr, String ort)`. Wird eine negative Dauer oder ein `String start` übergeben, der kein Datum repräsentiert, so soll eine Exception ausgelöst werden. (Beispiel für einen `String`, der einen Zeitpunkt repräsentiert: "01.03.2008_13:24")
- `Termin(String start, String ende, String beschr, String ort)`. Repräsentiert `start` oder `ende` kein Datum oder liegt das Ende des Termins vor dem Anfang, so soll eine Exception ausgelöst werden.

Ein Objekt der Klasse `Kalender` repräsentiert einen Terminkalender. Folgende Methoden sollen zur Verfügung gestellt werden:

- `void trageTerminEin(Termin t)`, die den Termin `t` in den Kalender einträgt. Gibt es eine Überschneidung mit bereits eingetragenen Terminen, so soll eine Exception ausgelöst werden;
- `Termin nachfolger(Termin t)`, die den nach dem Termin `t` folgenden Termin zurückgibt. Existiert kein Termin, so soll eine Exception ausgelöst werden;
- `String termineTag(String datum)`, die die Liste aller Termine eines Tages liefert. Repräsentiert `datum` kein gültiges Datum, so soll eine Exception ausgelöst werden.
- `Termin terminTag(String datum, int i)`, die den `i`-ten Termin eines Tages zurückliefert. Repräsentiert `datum` kein gültiges Datum oder gibt es keinen `i`-ten Termin, so soll eine Exception ausgelöst werden.

Gehen Sie zur Lösung der Aufgabe wie folgt vor:

- Überlegen Sie sich eine geeignete Hierarchie von Exceptions und definieren Sie diese in Java.
- Implementieren Sie die Klasse `Termin`.
- Implementieren Sie die Klasse `Kalender`.
- Vervollständigen Sie Ihr Programm um Benutzer-Interaktion.
- Testen Sie Ihre Implementierung anhand geeigneter Beispiele.

Aufgabe 48 (H) Contentcrawler

(12 Punkte)

`FilterReader` sind spezielle Arten von `Reader`-Klassen, die ein Bearbeiten von Eingabeströmen ermöglichen. In dieser Aufgabe wollen wir schrittweise eine Kette von `FilterReader`-Klassen entwickeln, mit deren Hilfe die häufigsten Wörter eines Internetauftritts automatisch ermittelt werden können.

- Implementieren Sie eine Klasse `HtmlFilterReader` mit der abstrakten Basisklasse `FilterReader`. Überschreiben Sie die Methode `int read()` derart, dass alle HTML-Tags aus dem Strom ignoriert werden. Ein HTML-Tag beginnt immer mit dem Zeichen '`<`' und endet mit dem Zeichen '`>`'.
- Implementieren Sie eine Klasse `UrlExtractorFilter` mit der abstrakten Basisklasse `FilterReader`. Überschreiben Sie die Methode `int read()` derart, dass die Adressen aller Verweise zu HTML-Dateien innerhalb des Stroms in eine Liste geschrieben werden. Ein Verweis auf die Adresse `http://www2.in.tum.de/index.html` hat in HTML die Form: ``.
- Implementieren Sie eine Klasse `DictionaryFilterReader` mit der abstrakten Basisklasse `FilterReader`. Überschreiben Sie die Methode `int read()` derart, dass alle Wörter aus dem Strom in eine interne Datenstruktur der Klasse geschrieben werden. Berücksichtigen Sie dabei nur Wörter, die aus sinnvollen Buchstaben bestehen, mit einem Großbuchstaben beginnen und eine Minimallänge von 5 Buchstaben haben.
- Mit dem Statement `new InputStreamReader(new URL(url).openStream(), "UTF-8")` erzeugen Sie einen Eingabestrom auf die im `String url` angegebene Internetseite. Benutzen Sie die vorher entwickelten Filterklassen (und erweitern Sie diese gegebenenfalls), um eine Liste der Wörter, die sich aus der Internetseite und aus von dieser Seite verwiesenen Internetseiten extrahieren lassen, und deren Häufigkeiten zu erstellen. Die Anzahl der maximal zu besuchenden Seiten soll einstellbar sein.
- Schreiben Sie ein Java-Programm, das sich mit den Parametern `<starturl> <n>` starten lässt. Als Ergebnis soll die Liste der 10 am häufigsten vorkommenden Wörter der Internetseite `starturl`, sowie maximal `n` von dieser Seite verwiesenen Internetseiten ausgegeben werden.