CLog Documentation

HHU-Programmierpraktikum SS2016 Projekt 5

Inhaltsverzeichnis

1 Verwendete Strukturen

1.1 Wrapper-Klassen

Wrapper-Klassen enthalten in der Regel eine private Variable des entsprechenden Datentyps, welche über den Konstruktor gesetzt werden kann. Da Getter- und Setter-Methoden nicht erlaubt sind, ist es nicht möglich, den gespeicherten Wert abzufragen oder nach der Erstellung zu manipulieren. Lediglich eine Ausgabe auf der Konsole (mittels `Ausgabe`) ist durch die Methode `ausgeben()` möglich und völlig ausreichend für die hier verwendeten Zwecke.

Wenn die Wrapper-Klassen Wrapper für Elemente eines Datensatzes sind, so enthalten sie in der Regel ebenfalls eine Methode `unterstreichen(char zeichen)` zum Ausgeben einer aus `zeichen` bestehenden Zeichenkette in der Länge des durch die Methode `ausgeben()` produzierten Strings, und eine Methode `vonEingabeEinlesen()`, welche statisch ist und die für ein Objekt des entsprechenden Typs von der Konsole einliest (den Nutzer dazu auffordert), ein Objekt des entsprechenden Typs daraus erzeugt und dieses zurückgibt.

1.2 „Vereinigungs“-Klassen

Ein Kernproblem ist die Restriktion von maximal 2 Attributen pro Klasse. Da recht viele Attribute gespeichert werden müssen, ist es nicht möglich, z.B. eine Klasse `Datum` zu erstellen, welche die Attribute `tag`, `monat` und `jahr` speichert.

Ich löse dieses Problem, indem ich erst eine Klasse `MonatJahr` erstelle, welche die Attribute `monat` und `jahr` speichert, und dann eine weitere Klasse `Datum`, welche die Attribute `monatJahr` und `tag` speichert, um diese Regel zu befolgen.

Etwaige Funktionsaufrufe von Methoden wie `ausgeben()` oder `unterstreichen()` (weiteres dazu später) bestehen dann aus den Aufrufen der entsprechenden Methoden bei den gespeicherten Objekten (z.B. `Datum.ausgeben()` ruft `MonatJahr.ausgeben()` und `Tag.ausgeben()` auf).

2 Packages

Im Programm sind die einzelnen Klassen in Packages sortiert:

- Package `data` enthält die Klassen, die einen Datensatz repräsentieren und die Datensätze verwalten:
  - `Datensatz` repräsentiert einen Datensatz
  - `DatensatzListe` speichert alle Datensätze in einer Liste, bietet Methoden zum Hinzufügen, Durchsuchen (und Ausgeben).
  - `DatensatzManager` verwaltet die Datensatzliste, also Datensätze hinzufügen, durchsuchen (und ausgeben), in eine Datei speichern und aus einer Datei lesen.
- Package `datumzeit` enthält alle Klassen, die zum dem Speichern von Datum und Uhrzeit verwendet werden:
  - `Jahr, Minute, Monat, Sekunde, Stunde` speichern die entsprechende Angabe als Integer.
  - `MonatJahr, StundeMinute` werden verwendet, um die jeweiligen Objekte zu vereinen (maximal 2 Attribute pro Klasse).
  - `Datum` stellt ein Datum dar (Vereint `MonatJahr` und `Tag`).
  - `Zeit` stellt eine Uhrzeit dar (Vereint `StundeMinute` und `Sekunde`)
- Package `eingabeausgabe` behandelt die Ein- und Ausgabe jeglicher Art, sowohl zum/vom Terminal als auch in/aus Dateien:
  - `Ausgabe` bietet Funktionen zum Ausgeben von Text in die Kommandozeile.
  - `Eingabe` bietet Funktionen zum Lesen von Text aus der Kommandozeile.
  - `DeSerializer` bietet Funktionen, um Java-Objekte in eine gegebene Datei zu speichern bzw. aus einer zu lesen.
  - `Path` repräsentiert einen Dateipfad.
- Package `fields` beinhaltet alle Felder, die in einem Datensatz gespeichert werden sollen (zusammengefügt aus gewrappten Strings und Primitives und deren zusammenführenden Klassen):

  - `DatumZeit, NameWohnort, TitelText` speichern jeweils die entsprechenden Objekte, um sie zu einem zu vereinen.
  - `NameWohnortDatumZeit` vereint `NameWohnort` und `DatumZeit` zu einem Objekt.
  - `NameWohnortDatumZeitTitelText` vereint `NameWohnortDatumZeit` und `TitelText` zu einem Objekt.
  - `Schlagworte` speichert eine `ArrayList⟨Schlagwort⟩`.
- Package `main` enthält lediglich die Main-Klasse, welche die Main-Methode enthält.
- Package `menues` enthält die Objekte, welche die Menüs repräsentieren. Jedes Menü-Objekt enthält dabei eine Methode `menue()`, welche die dem Menü zugeordnete Aktion ausführt (z.B. Daten Einlesen, Programm beenden etc.) und entsprechende Ein- und Ausgaben verwaltet. Die `menue()`-Methode endet, wenn eine jeweilige Aktion abgeschlossen ist.
- Package `textangaben` enthält Wrapper-Klassen und „Vereinigungsklassen“für alle weiteren Angaben eines Datensatzes (Textangaben, bis auf Datum + Zeit):
  - `Nachname, Schlagwort, Text, Titel, Vorname, Wohnort, Zeichen` speichern jeweils ein entsprechendes Element als `String` (bzw. `char` im Falle von Zeichen).
  - `Name` vereint `Vorname` und `Nachname` zu einem Objekt.

Beschreibung

3 User story

Der Nutzer startet das Programm im Terminal mit dem Aufruf `java Clog`. Dadurch wird in der Klasse `main.Clog` die Main-Methode `public static void main(String args[])` aufgerufen. Diese Methode enthält lediglich einen Aufruf an die Methode `menue()` der Klasse `menues.Menue0`, welche die Main-Loop enthält. Darin wird für den Nutzer lesbar das Hauptmenü ausgegeben, auf eine Antwort gewartet, die Antwort mittels eines Switches ausgewertet und die entsprechende Subroutine in einer der anderen Menüklassen aufgerufen.

```
Hauptmenü:
1) Clog-Eintrag erzeugen
2) Clog ausgeben
3) Clog laden
4) Clog speichern
5) Programm beenden
6) Alle Datensätze ausgeben
```

Abbildung 1: Hauptmenü

3.1 Menüpunkt 1: `Clog-Eintrag erzeugen`

Wählt der Nutzer Menüpunkt 1, so wird die Funktion `menue()` in der Klasse `menues.Menue1` aufgerufen. Diese behandelt das Erzeugen eines neuen Logeintrags. Dazu wird eine Aus-

gabe auf der Konsole erzeugt, welche den Nutzer über den gestarteten Vorgang informiert und die Bestätigung der Eingaben mit Enter anfordert.

3.2 Ausgaben und Eingaben

3.3 Lesen und Schreiben der Dateien