

Bericht Sprachkonzepte

Aufgabe 1

Fragestellung:

- Schreiben Sie alle Programmiersprachen (im weitesten Sinne) auf, die Ihnen bisher im Studium begegnet sind. Überlegen Sie zu jeder Sprache, welche Sprachmittel sie für den Umgang mit hierarchischer Gliederung, Abstraktion, Redundanz und Abhängigkeit bietet. Wir diskutieren Ihre Überlegungen in der Übungsstunde.

Lösung:

- [Übung 1](#)

Aufgabe 2

Teil a)

Fragestellung:

- Schreiben Sie ein Java-Programm, das in einem String Formatspezifikationen gemäß `java.util.Formatter` findet. Erstellen Sie dazu mit der Syntax von `java.util.regex.Pattern` einen regulären Ausdruck für eine solche Formatspezifikation.

Beispieleingaben:

`xxx %d yyy%n`

`xxx%1$d yyy`

`%1$-02.3dyyy`

Beispielausgaben:

`TEXT("xxx ")FORMAT("%d")TEXT(" yyy")FORMAT("%n")`

`TEXT("xxx")FORMAT("%1$d")TEXT(" yyy")`

`FORMAT("%1$-02.3d")TEXT("yyy")`

Lösung:

- [Uebung2.java](#)

Teil b)

Fragestellung:

- Erkennen Sie mit ANTLR 4 Lexer-Regeln Zeitangaben im digitalen 12-Sunden-Format. Beachten Sie auch die alternativen Schreibweisen 12 midnight und 12 noon

Lösung:

- [TimeSimple](#)

Wir hatten noch eine andere Grammatik die nicht ganz den Anforderungen entsprochen hat

- [Time](#)

Ausführung:

- [Ausführungsskript](#)
 - Befehl: `../Übung2/Uebung2.ps1 timeBeispiel.txt`

Aufgabe 3

Fragestellung:

- Denken Sie sich eine kleine Sprache aus. Definieren Sie deren Vokabular mit einer ANTLR4 lexer grammar und deren Grammatik mit einer ANTLR4 parser grammar.

Aufgabe 5

Teil a)

Fragestellung:

- Vervollständigen Sie das folgende Java-Programm, indem Sie die aufgerufenen Klassenmethoden ergänzen. Was an dem Java-Programm ist eindeutig prozeduraler Stil?

Lösung:

- [Aufgabe5_a.java](#)

Teil b)

Fragestellung:

- Stellen Sie das Programm aus 5a mit Hilfe von `java.util.streams` und Lambdas auf einen funktionalen Stil um. Ihr Programm darf nach der Umstellung keine Schleifen und Verzweigungen mehr enthalten.

Lösung:

- [Aufgabe5_b.java](#)

Teil c)

Fragestellung:

- Vergleichen Sie die Laufzeiten der Programme aus 5a und 5b.

Lösung:

- [Ausführungsskript](#)
- Befehl: `../Übung5/Uebung5.ps1 Test.txt`
- Output:

```
Aufgabe 5a): result = 152 (1335 microsec)
Aufgabe 5b): result = 152 (21612 microsec)
```

- Streams sind im Gegensatz zu Loops mit unserem Workload langsamer.

Aufgabe 7

Aufgabenstellung:

- Implementieren Sie eine Java-Anwendung, die für beliebige Java-Klassen und -Interfaces eine HTML-Seite im Format der Beispieldatei `aufgabe7.html` (siehe Moodle-Kursseite) generiert. Leiten Sie dazu aus `aufgabe7.html` eine Stringtemplategroup-Datei `aufgabe7.stg` ab. Die Java-Anwendung soll die gewünschten voll qualifizierten Klassen- und Interfacenamen als Aufrufparameter bekommen und mit Hilfe der Templates die HTML-Darstellung erzeugen.

Lösung:

- [Beispieldatei](#)
- Generierung des Class Arrays und Übergabe an Stringtemplate mit Hilfe der Template Engine:
[Aufgabe7.java](#)
- Stringtemplate: [aufgabe7.stg](#)
- [Ausführungsskript](#)
- Befehl: `../Übung7/Uebung7.ps1`
- Output: [output.html](#)

Aufgabe 8

Aufgabenstellung:

- Implementieren Sie eine kleine Anwendung mit einer Scriptsprache und analysieren Sie, welche typischen Eigenschaften einer Scriptsprache Sie dabei ausnutzen.

Lösung:

- Unser Python Skript: [Übung8.py](#)
- Wir haben die Feiertage Api verwendet und 2 Parameter übergeben. Einmal `"jahr": 2022` und `"nur_land": "BW"`. Die Rückgabe Daten haben wir dann in eine json Datei geschrieben und formatiert mit Hilfe der json Bibliothek von Python. Anschließend haben wir die Keys der json Datei mit Hilfe von einem Regex Pattern durchsucht.