

Aufgabenstellung 3 (Gruppen A und E)

Fachliche Aufgabenstellung

Sie sollen eine Anwendung erstellen, mit der Sie den Platzbedarf von Verzeichnissen und Dateien auf einem Datenträger visualisieren können. Es sollen verschiedene Darstellungsweisen angeboten werden:

- *Tree-Map*: Darstellung hierarchischer Strukturen durch ineinander verschachtelte Rechtecke, deren Größe proportional zur Größe der berücksichtigten Daten ist
- *Kreisdiagramm*: kreisförmige Darstellung, deren einzelne Sektoren die darzustellenden Elemente und deren Größe repräsentieren
- *hierarchische Liste*.

Ausgewertet werden sollen

- Platzbedarf:
 - Summe Platzbedarf für die Dateien im Verzeichnis und Platzbedarf aller Unterverzeichnisse
- Anzahl der Dateien / Unterverzeichnisse.

Funktionale Anforderungen

Es bestehen folgende funktionale Anforderungen:

- Auswahl des Datenträgers, der analysiert werden soll
 - auf der Startseite
 - in einem Navigationsbereich
- schrittweise hierarchische Verfeinerung in den Darstellungsformen Tree-Map und Kreisdiagramm:
 - zunächst wird nur das Wurzelverzeichnis des Datenträgers wiedergegeben
 - durch Klick auf ein Rechteck / einen Sektor wird die Analyse auf der nächsten Hierarchieebene fortgesetzt
 - es muss eine einfache Möglichkeit zum schrittweisen Aufstieg angeboten werden
- bei der Anzeige der hierarchischen Liste wird zunächst ebenfalls schrittweise verfeinert, die gewonnenen Daten bleiben aber bei der Darstellung verfügbar, verschiedene Zweige der Hierarchie können ggf. gleichzeitig dargestellt werden
- der aktuelle Pfad, der analysiert wird, muss als *Breadcrumb-Navigation* angezeigt werden
 - zur Verbesserung der Übersichtlichkeit
 - um schnell einen anderen Bezugspunkt für die Analyse auszuwählen
- der Typ der Dateien, die bei der Analyse berücksichtigt werden sollen, soll einstellbar sein.

Eine Anmeldung von Benutzern ist nicht erforderlich.

Anforderungen an die Gestaltung der Benutzungsschnittstelle

Die Benutzungsschnittstelle muss so gestaltet werden, dass sie auch für ungeübte Benutzer verwendet werden kann. Der Aspekt Barrierefreiheit muss, soweit das fachlich möglich ist, berücksichtigt werden.

Beachten Sie die Anwendung der Gestaltgesetze.

Nutzen Sie die Möglichkeiten des CSS-Frameworks, um eine übersichtliche und leicht anwendbare Benutzungsschnittstelle zu implementieren.

Anforderungen an die Umsetzung

Die Anwendung wird als Desktop-Anwendung auf Basis von NW.js mit Web-Techniken realisiert:

- verwenden Sie die Ausführungsumgebung NW.js
- verwenden Sie HTML5
- verwenden Sie CSS und das CSS-Framework uikit
- verwenden Sie die JS-Bibliothek d3 zur Visualisierung der Daten
- Erzeugung des Markups mit der Template-Engine te / tmlocal.

Beachten Sie die Hinweise im Abschnitt "Hilfsmittel".

Anforderungen an die Dokumentation

Allgemeine Anforderungen

Verwenden Sie als *Ausgangspunkt* für die Strukturierung Ihrer Dokumentation folgende Gliederung:

1. Einleitung
2. Entwurf der Anwendung
 - 2.1 Benutzungsschnittstelle
 - 2.1.1 Anwendungsrahmen und einzelne Sichten
 - 2.1.2 Zustandsmodell
 - 2.2 Bereitstellung Daten zur Visualisierung
3. Implementierung der Anwendung
 - 3.1. Front-End
 - 3.1.1 Aufbau der Benutzungsschnittstelle
 - 3.1.2 Ablaufsteuerung
 - 3.1.3 Visualisierung der Daten
 - 3.2. Back-End
 - 3.2.1 Schnittstelle (API)
 - 3.2.2 Zugriff Dateisystem und Auswertung

Die Dokumentation wird als utf-8 kodierter Text mit der einfachen Auszeichnungssprache *markdown* erstellt. Mit Hilfe des Werkzeugs *pandoc* (siehe Hilfsmittel) erfolgt die Umsetzung in eine HTML-Datei:

```
pandoc -f markdown -t html5 -s -c iasp1.css --toc <IhreDatei> -o <IhreHTML5Datei>
```

Die Datei *iasp1.css* enthält zusätzliche CSS-Stilregeln und wurde Ihnen bereits mit der Aufgabe 1 zur Verfügung gestellt.

Beachten Sie folgende Anforderungen:

- das Werkzeug *pandoc* **muss** genutzt werden
- **die Erstellung der Dokumentation mit anderen *markdown*-Werkzeugen / Editoren wird nicht zugelassen**
- **die in *pandoc* verfügbaren Erweiterungen der Auszeichnungssprache *markdown* müssen genutzt werden**, z.B. bei der Verwendung von Tabellen
- die Zustandsdiagramme, die Sie mit *UMLet* erstellen, binden Sie als Graphiken ein
- exportieren Sie dazu die Diagramme als PNG-Dateien.

Zustandsmodell

Erstellen Sie das Zustandsmodell mit Hilfe von UML-Diagrammen des Typs *UML-State-Machine*. Wählen Sie bei *UMLet* in der Auswahlliste (rechts oben) diesen Diagramm-Typ aus.

Jede Sicht wird als *Zustand* modelliert. Die Zustandsübergänge repräsentieren die vorgesehenen Bedienungen z.B. von Verweisen oder Schaltern.

Eine ausführliche Beschreibung finden Sie in den Materialien zum Buch "UML@Classroom" (siehe <http://www.uml.ac.at/de/lernen>, dort: "Zustandsdiagramm").

Hilfsmittel

Sie **müssen** folgende Hilfsmittel verwenden:

Erstellung Zustandsmodell

Erstellen Sie Zustandsdiagramme mit dem Werkzeug *UMLet* (siehe <http://www.umlet.com>).

Erstellung Dokumentation

Verwenden Sie *pandoc* (siehe <http://pandoc.org/>) zur Konvertierung der Markdown-Datei in eine HTML5-Datei.

Implementierung

Verwenden Sie:

- NW.js (siehe <http://nwjs.io>)
 - Ausführungsumgebung zusammengesetzt aus:
 - node.js (Javascript-Interpreter V8 mit Runtime-Umgebung)
 - chromium (Webbrowser)
 - Version **0.14.5 (LTS)** / **SDK-Build**
- javascript
- jquery
- d3 (siehe <http://d3js.org>)
- inheritance.js, es.js
- te.js, tmlocal.js (für den direkten Zugriff auf Dateien angepasste Version von tm.js)
- uikit (CSS-Framework, siehe <http://getuikit.com>)

Testat

Um das Testat zu erhalten, müssen Sie

- Ihre Anwendung vorführen und Entwurf und Implementierung erläutern
- Ihre Dokumentation vollständig vorlegen und erläutern.