

Ergebnisprotokolle

NETZWERKVERWALTUNGSTOOL

BRIAN KORDUAN, DAVIS FRÖSE, MARCO D'ADDONA

11IT1B

Inhalt:

- 01 - Erstellung des fertigen Entwurfes
- 02 - Abänderung des Entwurfes aufgrund einer Änderung des Arbeitsauftrages
- 03 - Fertigstellung der GUI sowie der Klasse CSVReadWrite
- 04 - Verbinden der Logik mit der GUI
- 05 - Anzeige der existierenden Komponenten in der GUI
- 06 - Bearbeiten und Löschen existierender Einträgen

Ergebnis Protokoll Nr. 01:

„Erstellung des fertigen Entwurfes“

Aktueller Stand:

Alle benötigten Klassen wurden erstellt. Somit existiert nun ein fertiger Entwurf

Es existieren nun folgende Klassen incl. Assoziationen und Vererbungen:

Gebäude, Raum, Komponente, Switch, Router, PC, Laptop, JSONReadWrite, Main, MainForm.

Testfälle:

Beschreibung	Erwartetes Ergebnis	Tatsächliches Ergebnis	Tester
Generierung von Objekten der einzelnen Klassen	Erfolgreicher Compiliervorgang	Erfolgreicher Compiliervorgang	Alle Gruppenmitglieder
Ausgabe einiger Attribute der Objekte in der Konsole	Ausgabe der Attribute	Ausgabe der Attribute	Alle Gruppenmitglieder

Weitere Vorgehensweise:

- Erstellung des grafischen User Interfaces (GUI)
- Implementierung der Methoden Read und Write in der Klasse JSONReadWrite zum Auslesen und Schreiben (befüllen) einer JSON Datei.
- Logik mit der GUI verbinden.
- Einfügen einer Tabelle mit den aktuellen Komponenten

Ergebnis Protokoll Nr.02

„Abänderung des Entwurfes aufgrund einer Änderung des Arbeitsauftrages“

Aktueller Stand:

Wir haben erfahren, dass der JSON Dateiformat nun noch nicht für den Austausch von Daten zwischen den Computersystemen verwendet werden soll.

Aufgrund dessen mussten wir uns in XML einarbeiten.

Da wir nur sehr wenige Kenntnisse in XML haben, mussten wir uns aufgrund der knappen Zeit für CSV als Datentyp um entscheiden. Dieser Datentyp war auch ursprünglich für diesen Zweck angedacht.

Diese Änderung der Anforderung hat uns sehr viel Zeit gekostet. Aufgrund dessen sind wir nicht viel weiter gekommen als wir es bereits bei Ergebnisprotokoll Nr. 01 waren.

Es wurde eine „Experimentierklasse“ erstellt, zum Testen, wie man eine CSV Datei mit Java erstellt.

Testfälle:

Beschreibung	Erwartetes Ergebnis	Tatsächliches Ergebnis	Tester
Generieren einer CSV Datei mit belanglosem Inhalt	Generierung der CSV Datei	Generierung der CSV Datei	Marco D'Addona

Weitere Vorgehensweise:

- Erstellung des grafischen User Interfaces (GUI)
- Implementierung der Methoden Read und Write in der Klasse CSVReadWrite zum Auslesen und Schreiben (befüllen) einer CSV Datei mit Objekten der entsprechenden Klassen.
- Logik mit der GUI verbinden.
- Einfügen einer Tabelle mit den aktuellen Komponenten

Ergebnis Protokoll Nr. 03

„Fertigstellung der GUI sowie der Klasse CSVReadWrite“

Aktueller Stand:

Die GUI wurde fertiggestellt sowie die benötigten Methoden der Klasse CSVReadWrite implementiert.

Somit ist es nun möglich, dass durch entsprechende Methodenaufrufe CSV Dateien generiert werden aus einer Liste (ArrayList) von Komponenten.

Ebenso wurde die GUI fertiggestellt.

Testfälle:

Beschreibung	Erwartetes Ergebnis	Tatsächliches Ergebnis	Tester
Erfolgreiches Compilieren der GUI	Anzeige der GUI	Anzeige der GUI	Brian Korduan, Marco D'Addona
Generierung der Datei data.csv mit	Generierung der csv Datei, in welcher Attribute von Objekten der Klasse Komponente gespeichert sind (Übergabe aus einer ArrayList).	CSV Datei wurde generiert mit allen Attributen der in der ArrayList übergebenen Objekte	Davis Fröse, Marco D'Addona
Generierung der Datei Gebaeude.csv	Generierung der csv Datei, in welcher Attribute von Objekten der Klasse Gebaeude gespeichert sind (Übergabe aus einer ArrayList).	CSV Datei wurde generiert mit dem meisten Attributen der Objekte. Fehlerhafte Anzeige der Übergebenen Räume der Gebäude.	Davis Fröse, Marco D'Addona

Probleme:

Leider hatten wir vergessen, die Klasse „JSONReadWrite“ (die nicht mehr benötigt war) zu löschen, welche eine Bibliothek, die bereits entfernt wurde.

Aufgrund dessen konnte das Programm nicht richtig kompilieren.

Ergebnis Protokoll Nr. 04

„GUI mit der Logik (CSVReadWrite) verbinden “

Aktueller Stand:

Die GUI kann nun Daten aus der Klasse CSVReadWrite lesen, neue CSV Dateien erstellen bzw. bestehende überarbeiten.

Testfälle:

Beschreibung	Erwartetes Ergebnis	Tatsächliches Ergebnis	Tester
Erfolgreiches Compilieren der GUI	Anzeige der GUI	Anzeige der GUI	Alle Gruppenmitglieder
Hinzufügen von Komponenten zu einem Raum über die GUI	Speicherung der neuen Werte in die CSV Dateien	Speicherung der neuen Werte in die CSV Dateien	Alle Gruppenmitglieder

Ergebnis Protokoll Nr. 05

„Anzeige der existierenden Komponenten in der GUI“

Aktueller Stand:

Unterhalb der Optionen wird nun eine Tabelle (JTable) angezeigt, welche mit den Daten aus der CSV Datei gefüllt ist.

Testfälle:

Beschreibung	Erwartetes Ergebnis	Tatsächliches Ergebnis	Tester
Anzeige der Daten aus der CSV Datei	Daten aus der CSV Datei werden ausgelesen und in der Tabelle angezeigt	Daten aus der CSV Datei werden ausgelesen und in der Tabelle angezeigt	alle Gruppenmitglieder

Weitere Vorgehensweise:

- Bearbeiten und löschen existierender Einträge

Ergebnis Protokoll Nr. 06

„Bearbeiten und Löschen existierender Einträgen“

Aktueller Stand:

Man kann nun Einträge direkt in der Tabelle bearbeiten und löschen.
Die CSV Datei wird anschließend aktualisiert.

Testfälle:

Beschreibung	Erwartetes Ergebnis	Tatsächliches Ergebnis	Tester
Einträge lassen sich in der Tabelle anklicken und bearbeiten	Tabelleneinträge lassen sich bearbeiten	Tabelleneinträge lassen sich bearbeiten	alle Gruppenmitglieder
Klick auf den Button „Änderung übernehmen“	Abänderung der Tabelle und Aktualisierung der .csv Datei	Abänderung der Tabelle und Aktualisierung der .csv Datei	alle Gruppenmitglieder

Weitere Vorgehensweise:

- Abgabe des Projektes