

DOKUMENTATION

WORLD-CUP-BETTING

Jan Schneider, HfG, IoT3

<https://github.com/jan-patrick/World-Cup-betting>

Nach einigen generellen Überlegungen habe ich mich für eine ähnliche Struktur entschieden, wie sie jetzt noch immer implementiert ist. Unbestreitbar ist die Berechtigung der Klasse Land. Dort wird der Name, Tore und Punkte des Landes gespeichert. Alle Länder werden wiederum in einer Instanz der Klasse Gruppe gespeichert, die wiederum auch die Gruppenphasespiele enthält. Anfangs war Spiel eine eigene Klasse, aber letztlich sind Spiele nur eine Zusammensetzung aus Ländern mit Zusatzinformationen, weswegen daraus eine Liste wurde. Ebenfalls sind die Gruppenphasespiele in Gruppe gewandert, da diese Spiele eben nur dort benötigt werden und die Spiele danach (Achtel-, Viertel-, Halbfinale und Finale) davon unabhängig sind. Gruppen werden wiederum in der Main gespeichert, da auch dort später eine Auswertung derer nötig ist, zum Beispiel wer in die nächste Runde kommt.

Um alle Daten zu speichern und beim Schließen bzw nach dem erneuten Starten alle Eingaben nicht erneut einzugeben, werden sowohl alle Länder samt Punkten und Toren, alle Gruppen samt Spielen und Ergebnissen und allgemeine Informationen, implementiert ist der Turniername, in Textdateien gespeichert. Dies geschieht über eine eigene Klasse, Daten. Um auch dies ordentlich zu strukturieren ist dies aufgeteilt in drei Ordner: Allgemein, Gruppen und Länder. Diese liegen im Projektverzeichnis.

GitHub

Eine Grundüberlegung meinerseits war, dass die Interaktion mit meinem Programm wesentlich einfacher wäre, wenn das Programm auf Knopfdruck die aktuellsten Ergebnisse bereits kennt. Deshalb nutze ich das GitHub Repository als eine Art Vorlagenspeicherung. Einerseits besteht so die Möglichkeit, dass wenn lokale Daten beschädigt sind, diese einfach durch die Vorlagendaten ersetzt werden können. Andererseits kann so auf Knopfdruck immer der aktuellste Stand (sofern online

eingetragen) abgefragt und lokal ausgegeben werden. So ist im Idealfall keine Eingabe durch den Nutzer mehr nötig. Zudem ist eine Versionsverwaltung natürlich für mich zum Schreiben des Codes von Vorteil. Das Repository können Sie entweder direkt im Projekt öffnen (öffnet sich im Standardbrowser) oder über den oben stehenden Link.

Bedienung

Um das Projekt zu starten muss eine Instanz der Klasse Main erstellt werden. Diese lädt automatisch alle verfügbaren Daten aus den lokal gespeicherten Textdateien. Sollte dabei ein Fehler auftreten wird dieser zwar angezeigt und meist das Programm auch noch richtig gestartet, aber sollte dies wirklich nicht möglich sein können die Vorlagendaten heruntergeladen, gespeichert und im Programm verwendet werden. Sonst funktioniert die Bedienung wie gewohnt: In der Instanz der Klasse Main können alle möglichen Methoden aufgerufen werden. Diese Methoden rufen wiederum in der Klasse Interface ein eigenes Fenster auf, über welches anschließend mit dem Programm interagiert werden kann oder auch die Ausgabe stattfindet.

Falscheingaben

Da Eingabefehler ein ernstes Problem darstellen um die Lauffähigkeit des Programms zu bewahren, gerade da ich nun Textdateien verwende, anstatt wie ich es zu Beginn versucht habe, xml-Dateien einzusetzen, einfach da ich damit schneller und flexibler arbeiten kann, habe ich eine relativ restriktive Eingabenprüfung eingebaut.

Umlaute, Leerzeichen und die Großschreibung werden überprüft, entfernt, bzw umgewandelt, sodass eine beliebige Eingabe von Zeichen wie z.B. „ÄGYt= e)n“ trotzdem die Anzeige und das Speichern des Landes Ägypten erfolgt.

Werden Länder oder Gruppen falsch eingegeben wird es dem Benutzer auch mitgeteilt.

Sogar das Überschieben eines Spielergebnisses mit korrekter Punkteberechnung wird nach Bestätigung durchgeführt. Anderweitige Korrekturen können von dem Benutzer selbst durchgeführt werden, da dieser fast alle Freiheiten über das Hinzufügen oder Entfernen von Gruppen und Ländern hat, nur mit geringen Einschränkungen. Es müssen aber immer zwei Gruppen vorhanden sein, mindestens zwei Länder in einer Gruppe und der Turniername darf nicht zu lange oder zu kurz sein. Auch ein Reset aller lokal gespeicherten Spielergebnisse ist möglich, ohne die Vorlagendaten zu laden. So ist es im Prinzip möglich das komplette Turnier noch einmal neu mit Ergebnissen zu füttern.

Tests

Getestet wurde das Projekt durch Eingaben in die Methodenaufrufe und die jeweiligen Interfaces. Aufgrund der Struktur des konstanten Ladens und Speicherns in die Textdateien und dem Interface sind nur wenige Funktionen vorhanden, die durch einen klassischen Test überprüft werden können. Auch deshalb ist es eben möglich einen kompletten Reset des Programms durchzuführen, auch wenn eine Fehlermeldung nicht zwingend bedeutet, dass das Programm nicht mehr funktionieren würde.

Interface

Ausschlaggebend für den Einsatz eines Interfaces, bzw. die Entscheidung gegen die serielle Ausgabe, war die Grundüberlegung, dass für den Nutzer die einfache Übersicht des Gruppen und Spiele im Vordergrund stehen sollte. So kann über einen einfachen Methodenaufruf der aktuellste Übersichtsplan übersichtlich angezeigt werden.

Fazit

Generell habe ich, gerade auch mit wachsendem Code, versucht auf Codeduplizierung zu achten, weshalb ich beispielsweise die Methode zum Speichern von Dateien universell anwendbar auf alle Möglichkeiten (um-) geschrieben habe. Die Datenhaltung ist selbstverständlich geschlossen und die Kopplung so lose wie möglich.

Gleichzeitig benutze ich zum Teil sehr spezifische Methoden, die die Kohäsion teilweise sehr gut erfüllen, aber natürlich gibt es hier noch Potential. Generell läuft das Programm sehr gut, gerade was das Anzeigen der neuen und aktuellen Spielergebnisse betrifft, was natürlich an Reiz verloren hat, da die Gruppenphase bereits vorüber ist.

HashMaps habe ich im übrigen verwendet um immer eindeutig zu wissen wo welcher Eintrag steht, ohne dessen Inhalt zu kennen - ein Vorteil zur ArrayList bzw. Array.