

TGM

Segelverein

INSY

Thomas Taschner 4AHIT



2014/15

Inhalt

Aufgabenstellung	2
Beschreibung auf Moodle	2
Designüberlegung	3
Erster Ansatz	3
Konkrete Idee	3
Umsetzung	4
Detaillierte Arbeitsaufteilung (Aufwandsabschätzung, Endzeitaufteilung)	6
Aufwandabschätzung	6
Endzeitaufteilung	6
Fazit	6
Arbeitsdurchführung (Resultate/Niederlagen)	7
Resultate	7
Niederlagen	7
Testbericht	8
Coverage	8
Beschreibung	8
Quellenangaben	9

Aufgabenstellung

Beschreibung auf Moodle

Für Segler und Trainer sind Name (NAME) und Geburtsdatum (GEBURTSDATUM) bekannt. Sie werden beide identifiziert durch eine eindeutige Nummer (KEY). Mindestens zwei Segler, maximal jedoch vier Segler bilden eine Mannschaft. Für jede Mannschaft werden ein eindeutiger Name (NAME) und eine Altersklasse (AKLASSE) gespeichert. Jede Mannschaft wird genau von einem Trainer betreut. Ein Trainer kann jedoch mehrere Mannschaften betreuen.

Jeder Mannschaft sind Boote zugewiesen. Ein Boot kann mehreren Mannschaften zugewiesen sein. Ein Boot wird eindeutig durch eine Nummer (ID) identifiziert. Weiters sind zu jedem Boot ein Name (NAME), die Anzahl der Personen (PERSONEN) und der Tiefgang (TIEFGANG) bekannt. Es gibt Tourenboote und Sportboote. Tourenboote haben zusätzlich eine Bootsklasse (BOOTSKLASSE) und Sportboote haben zusätzlich eine Segelfläche (SEGELFLAECHE) gespeichert. Es ist außerdem bekannt welche Mannschaften mit welchen Sportbooten an welchen Regatten mit welcher Startnummer (STARTNR) teilgenommen haben.

Eine Regatta wird eindeutig identifiziert durch ihren Namen (NAME) und durch das Jahr (JAHR), in dem sie stattgefunden hat. Das Land (LAND) ist außerdem noch bekannt. Jede Regatta besteht aus mindestens drei jedoch maximal fünf Wettfahrten. Wettfahrten werden durch die zugehörige Regatta und das Datum (DATUM) identifiziert, außerdem wird die Länge (LAENGE) der Strecke gespeichert. Mannschaften können bei jeder Wettfahrt Punkte (PUNKTE) erzielen.

Designüberlegung

Erster Ansatz

Der bereits bestehende Code aus der Übung ‚The Exporter‘ wird übernommen, auf Postgres geportet, dem Parser ein Großteil der Argumente entfernt und zu guter Letzt die GUI entwickelt.

Konkrete Idee

Die Idee entspricht dem Ansatz. Eine genauere Erläuterung befindet sich unter ‚Umsetzung‘.

Umsetzung

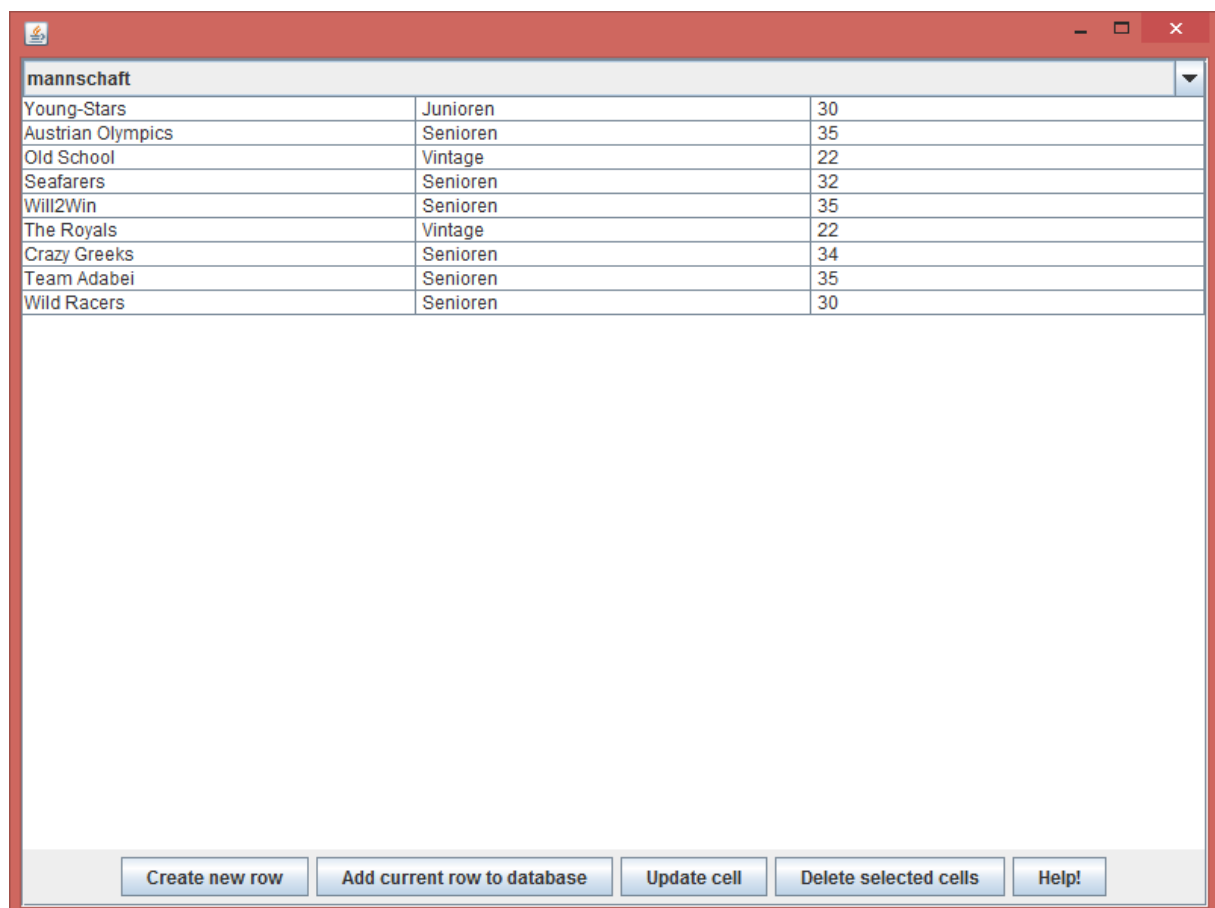
Die Datenbank wurde nach dem RM und dem ERD der Angabe umgesetzt. Einzig und allein die Checks, ob eine Mannschaft die maximale Personenanzahl erreicht hat, konnte nicht umgesetzt werden. Die Inserts konnten leider auch nur teilweise generiert werden. Zusätzlich zum Datafiller [] wurde auch noch ein Generator namens Spawner [] zur Generierung von Namen

Die Abfragen wurden mit Hilfe von extern generierten Datensätzen gelöst, da ich beim Generieren von Datensätzen mit Hilfe von Spawner ein paar Probleme hatte. Sobald Daten basierend auf anderen Daten generiert werden sollen, stehe ich etwas auf der Leitung.

Die MySqlConnection Klasse wurde übernommen und auf Postgres geportet, indem die vom Postgres Connector benötigten Strings angepasst wurden. Zusätzlich wurde die Klasse noch mit weiteren query Methoden ausgestattet. Dem Parser wurden nun nicht mehr benötigte Codezeilen entfernt. Es können nun nur noch die Hostadresse, der Benutzername, das Passwort und der Datenbankname angegeben werden, wie der beiliegenden README zu entnehmen ist.

Die GUI wurde von Grund auf entwickelt.

Ganz oben befindet sich eine JComboBox, dank der eine Tabelle gewählt werden kann. Sofern eine Wahl getroffen wurde, werden nun mit Hilfe eines JTables die ebenfalls editierbaren Datenbankeinträge dargestellt. Ganz unten befinden sich JButtons, mit einer selbsterklärenden Beschriftung. Alle GUI Elemente wurden auf einem JScrollPane eingebunden (scrollbares Panel).



Das Softwaredesign der GUI hätte ursprünglich noch besser, feiner werden sollen. Aufgrund eines Zeitmangels wurde aber schlussendlich alles in eine Klasse verfrachtet.

Detaillierte Arbeitsaufteilung (Aufwandsabschätzung, Endzeitaufteilung)

Aufwandabschätzung

Da bezüglich JDBC schon einige Erfahrung gesammelt wurden, sollte dieser Teil recht schnell erledigt sein. Der SQL Teil könnte etwas dauern, hauptsächlich aufgrund von schwachen SQL Kenntnissen. Erste Erfahrungen mit JTables wurden auch noch nicht gesammelt, somit könnte auch dieser Part etwas dauern. Die Tests mit Hilfe von Mock Objekten und GUI Tests könnten sich als interessant erweisen, da wir Mock Objekte nie wirklich im Unterricht besprochen haben (genau 1x und dann auch nur recht kurz) und ich die GUI mehr zu Coveragezwecken testen wollte.

Endzeitaufteilung

Task	Voraussichtlich benötigt	Tatsächlich benötigt
MySQLConnection porten	2 h	2 h
GUI entwickeln	10 h	15 h
Argumentparser umschreiben	1 h	30 Min.
Dokumentation schreiben	1 h	30 Min.
CREATE, START, DROP, INSERT, ... Script schreiben	2 h	10 h
SQL Queries entwickeln	5 h	12 h
Gesamt	21 h	40 h

Fazit

Die tatsächlich benötigte Zeit ist doppelt so groß, wie ursprünglich angenommen. Das Hauptproblem waren die GUI, die SQL Queries (bei denen ich sehr viel Zeit bedingt durch Unwissenheit verloren habe) und ebenfalls meine Schwierigkeiten mit dem Datafiller hatte. Nützliches Tool, aber noch etwas benutzerunfreundlich und zu schlecht dokumentiert. Hätte ich Python Kenntnisse, dann könnte ich mir durchaus vorstellen das Tool weiterentwickeln zu wollen. Das Tool Spawner war auch eine große Hilfe in der Generierung von Daten.

Arbeitsdurchführung (Resultate/Niederlagen)

Resultate

Das Umschreiben von bereits bestehendem Code hat wunderbar geklappt. Der SQL Part hat sich leider ein wenig in die Länge gezogen, da meine SQL Kenntnisse nicht sonderlich fortgeschritten waren und der Datafiller eben etwas umständlich von der Bedienung ist. Schlussendlich konnte ich nur ca. 5 Insert Scripts generieren lassen. Sobald Daten, die auf anderen Daten basieren generiert werden müssen, komm ich nicht weiter.

Niederlagen

Wie bereits erwähnt, konnte ich Daten „fortgeschrittener“ Natur nicht mehr generieren lassen, die Doku dieses Tools mangelt an Beispielen. Einfache Fälle zur Generierung sind vorhanden, sobald man aber ein wenig mehr machen möchte... Das Problem ist auch, dass das Tool nicht sonderlich verbreitet ist und es somit sonst kaum andere Anlaufstellen gibt. Die GUI hat aufgrund von mangelnder JTable Kenntnisse ein wenig Zeit beansprucht. Besonders das Verstehen des Zusammenspiels von Model und View hat etwas Einarbeitungszeit gekostet.

Testbericht

Coverage

Keine Tests vorhanden ☹️

Beschreibung

Keine Tests vorhanden ☹️

Quellenangaben

Wikipedia, APIs, offizielle Dokumentationen, Stackoverflow, diverse Foren,...