7.1.2015

Hampl & Kritzl

Rückwärtssalto

A05 - Metadata

Inhalt

[1. Angabe 2](#_Toc410645580)

[2. detaillierte Arbeitsaufteilung mit Aufwandsabschätzung 3](#_Toc410645581)

[2.1 Funktionale Anforderungen 3](#_Toc410645582)

[2.2 Nicht funktionale Anforderungen 3](#_Toc410645583)

[2.3 Organisatorische Anforderungen 4](#_Toc410645584)

[3.3 Gesamtsumme 4](#_Toc410645585)

[4. Designüberlegung 5](#_Toc410645586)

[4.1 Abbildung 5](#_Toc410645587)

[4.2 Überlegungen zur Struktur 6](#_Toc410645588)

[5. Arbeitsdurchführung 6](#_Toc410645589)

[6. Testfälle 7](#_Toc410645590)

[6.1 Starten des Programms 7](#_Toc410645591)

[6.2 Befehl „HELP“ 7](#_Toc410645592)

[6.3 Fehlermeldung 7](#_Toc410645593)

[6.4 Programm beenden 7](#_Toc410645594)

[6.5 Einloggen 8](#_Toc410645595)

[6.6 Beitreten eines Chatraums 8](#_Toc410645596)

[6.7 Senden und Empfangen von persönlichen Nachrichten 9](#_Toc410645597)

[6.8 Benutzer wechseln 9](#_Toc410645598)

[7. Lessons learned 10](#_Toc410645599)

[8. Quellenangaben 11](#_Toc410645600)

# 1. Angabe

Erstelle ein Java-Programm, dass Connection-Parameter und einen Datenbanknamen auf der Kommandozeile entgegennimmt und die Struktur der Datenbank als EER-Diagramm und Relationenmodell ausgibt (in Dateien geeigneten Formats, also z.B. PNG für das EER und TXT für das RM)

Verwende dazu u.A. das ResultSetMetaData-Interface, das Methoden zur Bestimmung von Metadaten zur Verfügung stellt.

Zum Zeichnen des EER-Diagramms kann eine beliebige Technik eingesetzt werden für die Java-Bibliotheken zur Verfügung stehen: Swing, HTML5, eine WebAPI, ... . Externe Programme dürfen nur soweit verwendet werden, als sich diese plattformunabhängig auf gleiche Weise ohne Aufwand (sowohl technisch als auch lizenzrechtlich!) einfach nutzen lassen. (also z.B. ein Visio-File generieren ist nicht ok, SVG ist ok, da für alle Plattformen geeignete Werkzeuge zur Verfügung stehen)

Recherchiere dafür im Internet nach geeigneten Werkzeugen.

Die Extraktion der Metadaten aus der DB muss mit Java und JDBC erfolgen.

Im EER müssen zumindest vorhanden sein:

* korrekte Syntax nach Chen, MinMax oder IDEFIX
* alle Tabellen der Datenbank als Entitäten
* alle Datenfelder der Tabellen als Attribute
* Primärschlüssel der Datenbanken entsprechend gekennzeichnet
* Beziehungen zwischen den Tabellen inklusive Kardinalitäten soweit durch Fremdschlüssel nachvollziehbar. Sind mehrere Interpretationen möglich, so ist nur ein (beliebiger) Fall umzusetzen: 1:n, 1:n schwach, 1:1
* Kardinalitäten

Fortgeschritten (auch einzelne Punkte davon für Bonuspunkte umsetzbar)

* Zusatzattribute wie UNIQUE oder NOT NULL werden beim Attributnamen dazugeschrieben, sofern diese nicht schon durch eine andere Darstellung ableitbar sind (1:1 resultiert ja in einem UNIQUE)
* optimierte Beziehungen z.B. zwei schwache Beziehungen zu einer m:n zusammenfassen (ev. mit Attributen)
* Erkennung von Sub/Supertyp-Beziehungen

# 2. detaillierte Arbeitsaufteilung mit Aufwandsabschätzung

## 2.1 Funktionale Anforderungen

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Arbeitspaket | Person | Schätzung(min) | Tatsächlich(min) | Erledigt | Testen |
| **Usereingabe** | **Hampl** | **0** | **0** | **x** | **x** |
| Parsen | Hampl | 10 | 5 | x | x |
| Überprüfen | Hampl | 10 | 5 | x | x |
| Hilfe | Hampl | 20 | 20 | x | x |
| **Datenbankverbindung** | **Kritzl** | **0** | **0** | **x** |  |
| MySQL | Kritzl | 30 | 60 | x |  |
| **Auslesen Metadaten** | **Kritzl** | **0** | **0** | **x** | **x** |
| Datenbank | Kritzl | 60 | 30 | x | x |
| Tabelle | Kritzl | 60 | 60 | x | x |
| Attribut | Kritzl | 60 | 120 | x | x |
| Referenz | Kritzl | 90 | 120 | x | x |
| **Export in EER** | **Hampl** | **0** | **0** |  |  |
| Syntaktisch | Hampl | 120 | 0 |  |  |
| optisch | Hampl | 120 | 0 |  |  |
| **Export in RM** | **Hampl** | **0** | **0** | **x** | **x** |
| Syntaktisch | Hampl | 30 | 120 | x | x |
| **Gesamt** |  | **610** | **540** |  |  |

## 2.2 Nicht funktionale Anforderungen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Arbeitspaket | Person | Schätzung(min) | Tatsächlich(min) | Erledigt |
| **Exception-handling** | **Kritzl** | **40** | **20** | **x** |
| **Build-Automation** | **Hampl** | **45** | **120** | **x** |
| **Testen** | **Hampl/Kritzl** | **150** | **360** |  |

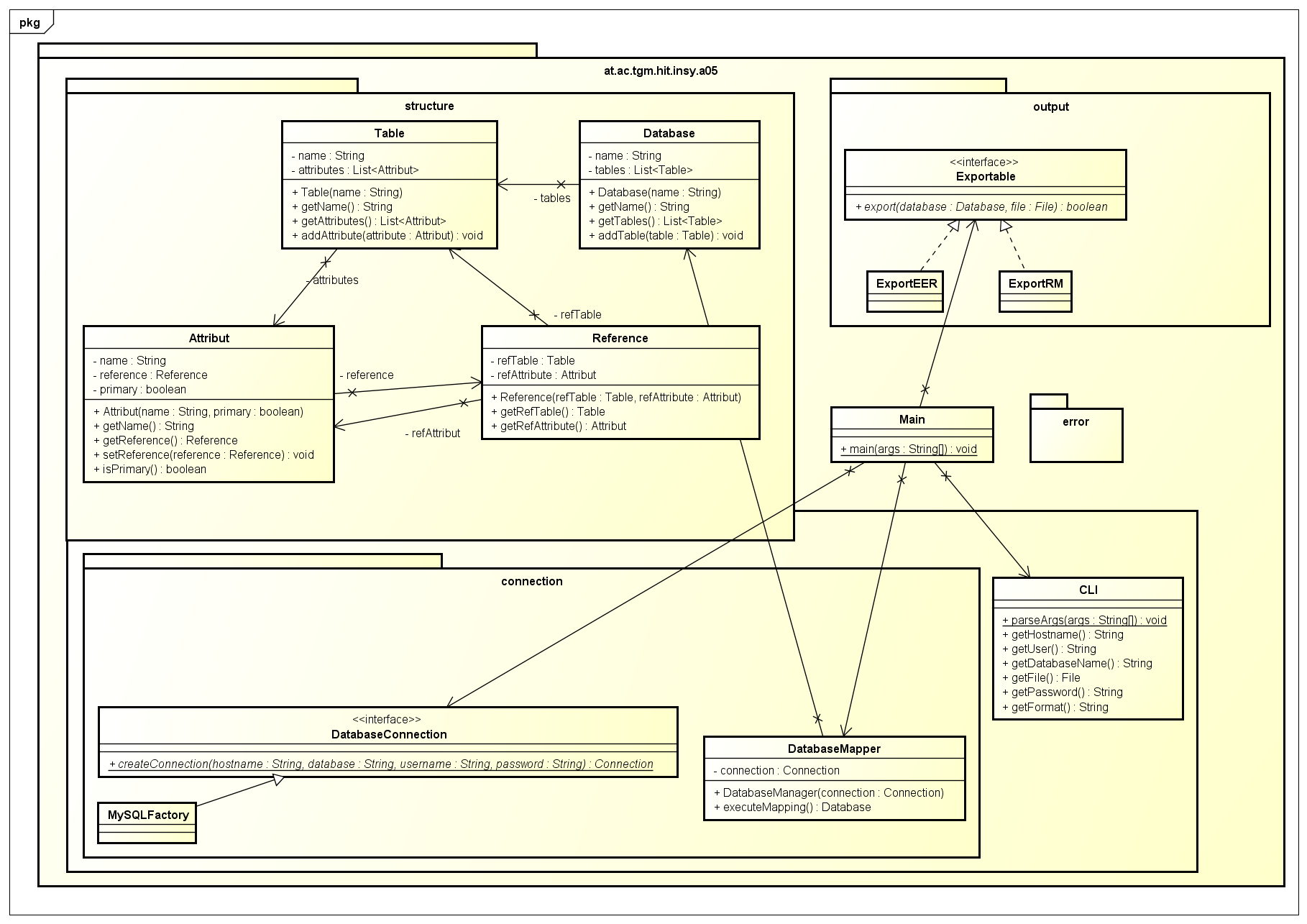
## 2.3 Organisatorische Anforderungen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Arbeitspaket | Person | Schätzung(min) | Tatsächlich(min) | Erledigt |
| **Dokumentation** | **Kritzl/Hampl** | **0** |  |  |
| JavaDoc | Kritzl/Hampl | 40 | 20 |  |
| Protokoll | Kritzl/Hampl | 120 |  |  |

# 4. Designüberlegung

## 4.1 Abbildung

Das UML-Diagramm wurde mit dem Programm „Astah“ erstellt.



## 4.2 Überlegung

* Das package „structure“ stellt eine Datenbank und deren Inhalte in einer objektorientierten Form dar. Damit ist die Datenbank, deren Tabellen, deren Attribute und deren Eigenschaften wie Primary, Unique und Not Null genauso wie die Foreign Keys der Attribute gespeichert. Dadurch kann sehr einfach auf die einzelnen Eigenschaften zugegriffen werden und nicht immer umständlich direkt mit der Datenbank kommunizieren zu müssen.

# 5. Arbeitsdurchführung

Da wir während der Implementierung auf einige Verbesserungen bezüglich der Struktur gekommen sind, sieht unser finales UML-Diagramm folgendermaßen aus:

# C:\Users\Martin Kritzl\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Klassendiagramm.png

# 7. Lessons learned

# 8. Quellenangaben