GYMANSIUM OTTOBRUNN

Oberstufenjahrgang 2017/19

Seminarfach Informatik

Seminararbeit

Kalender auf Basis von XML

Verfasser: Eric Brendel

Seminararbeit: Peter Brichzin

Bewertung: …………. Punkte

Unterschrift des Seminarleiter: ………………………………..

Inhaltsverzeichnis

[Vorwort 1](#_Toc524545924)

[Einleitung 1](#_Toc524545925)

[XML und DTD 2](#_Toc524545926)

[XML 2](#_Toc524545927)

[Was bedeutet das für den Programmierer? 2](#_Toc524545928)

[Welchen Nutzen kann der Programmierer daraus ziehen? 2](#_Toc524545929)

[DTD 3](#_Toc524545930)

[Notwendigkeit und Vorteile einer DTD 3](#_Toc524545931)

[Terminkalender 3](#_Toc524545932)

[Produkt und Funktionen 3](#_Toc524545933)

[Beginn des Projekts 3](#_Toc524545934)

[User Stories 4](#_Toc524545935)

[Tasks und Prototypen 4](#_Toc524545936)

[Zeitliche Planung 4](#_Toc524545937)

[Zusammenarbeit der Klassen (Klassendiagramm) 4](#_Toc524545938)

[DTD (eigene Sprache) 4](#_Toc524545939)

[Schnittstelle Java XML (Datenspeicherung) 4](#_Toc524545940)

[GUI mit JavaFX 4](#_Toc524545941)

[Einschätzung des Projekts (Fazit) 4](#_Toc524545942)

[Einschätzung des Projekts 4](#_Toc524545943)

[Einschätzung des Produkts 4](#_Toc524545944)

[Literaturverzeichnis 4](#_Toc524545945)

# Vorwort

Diese Seminararbeit beschreibt das Entstehen und die Abläufe eines in Java geschriebenen Programms. Um das Geschriebene verstehen zu können, ist es daher notwendig Grundkenntnisse in Java oder anderen Objektorientierten Programmiersprachen zu haben. Vorkenntnisse zum Thema XML sind nicht notwendig, da dieses Thema vor der eigentlichen Erläuterung der Projekts erklärt wird.

# Einleitung



# XML und DTD

Um einen guten Einstieg in die Thematik zu liefern und um das Verständnis für die folgende Seminararbeit zu verbessern handelt dieser kurze Abschnitt über Themen XML und DTD, die einen wichtigen Grundpfeiler für das mit der Arbeit zusammenhängende Projekt liefern.

## XML

Bei XML handelt es sich um die sogenannte „eXtenable Markup-Language“[[1]](#footnote-1). Das bedeutet zum einen, dass es sich nicht um eine Programmiersprache, wie beispielsweise Java oder C++, sondern um eine Auszeichnungssprache handelt. Das „extenable“ bedeutet ist Englisch und bedeutet erweiterbar. Es ist also möglich die Sprache zu erweitern, bzw. eine neue Sprache zu erstellen und sie an die eigenen Ansprüche anzupassen.

### Was bedeutet das für den Programmierer?

Der wesentliche Unterschied zwischen einer Programmier- und eine Markup-/Auszeichnungssprache besteht darin, dass es sich bei der Zweiten um eine beschreibende Sprache handelt. Beide Sprachen sind zwar künstlich und die haben Aufgabe den Computer etwas mitzuteilen und zu befehlen und werden außerdem in Form von Quelltext geschrieben, der Unterschied bestehen jedoch darin, dass mit Markupsprachen Dokumente erzeugt werden und mit Programmiersprachen Programme. Außerdem verarbeitet der Computer beide unterschiedlich verarbeitet: ein XML-Dokument wird von einem Parser analysiert, bei einem Programm werden Anweisungen von einem Interpreter oder dem Betriebssystem selbst abgearbeitet. Dies ist aber für das Verständnis dieser Seminararbeit unwesentlich.

Für den Programmierer bedeutet das im Wesentlichen, dass eine Auszeichnungssprache die Struktur von Daten beschreibt und Programmiersprachen für die Realisierung logischer Abläufe verwendet werden.   
Ein einfacher Unterschied ist zum Beispiel, dass es in Auszeichnungssprachen weder Bedingte Anweisungen von Wiederholungsanweisungen oder Methoden gibt, sondern nur Elemente und deren Attribute, beziehungsweise ihren Inhalt[[2]](#footnote-2).

### Welchen Nutzen kann der Programmierer daraus ziehen?

XML ist ein sehr einfacher und praktischer Weg Daten zu speichern. Man kann mit Hilfe von Bibliotheken sogar ganze Objekzustände ohne größeren Aufwand speichern oder auslesen. Ein Vorteil ist hierbei auch, dass die Dateien nicht an eine Programmiersprache oder ein Betriebssystem gebunden, sondern von allen Betriebssystemen und Programmiersprachen benutzt werden können. Hierfür ist jedoch eine entsprechende Bibliothek notwendig. Eine sehr ähnliche Sprache „HTML“, aus der XML abgeleitet wurde, wird zum Beispiel zum Darstellen von Webseiten genutzt. FXML, eine mit XML erstellte Sprache, wird verwendet um Java-Applikationen eine Benutzeroberfläche zu geben und somit die Nutzerinteraktion einfacher zu gestalten. Auch in diesem Projekt ist die GUI mit JavaFX und FXML erstellt, dies wird aber später genauer erläutert.

## DTD

Bei einer DTD handelt es sich um eine Dokumenttyp-Definition. Mit Hilfe dieser können Regeln für das Dokument definiert werden um dieses zu gestalten. DTDs dient also als eine Art Regelwerk für das XML-Dokument und definiert die darin vorhandenen Elemente und deren Attribute.[[3]](#footnote-3)

### Notwendigkeit und Vorteile einer DTD

Eine DTD ist nicht zwingender Weise für ein XML-Dokument notwendig. So kann ein Dokument auch problemlos ohne DTD eingelesen werden, dennoch bringt die Verwendung einige Vorteile mit sich:

* Ordnung/ Struktur in dem Dokument: mit einer DTD erstellt man Regeln (für eine Struktur), die befolgt werden müssen, daher ist ihr Dokument ordentlicher und strukturierter, kann also auch einfach eingelesen werden
* Keine Missverständnisse bei mehreren Programmieren: mit einer DTD kann sichergestellt werden, dass alle Programmierer bei dem Erstellen oder Verarbeiten des Dokuments der gleichen Struktur folgen. Damit sind Fehler und unterschiedlichen Aufbau der Dokumente ausgeschlossen.
* Mehrere Benutzer können die Referenzstruktur verwenden: Es gibt viele XML-Applikationen, die über eine DTD verfügen (z.B. FXML, CFD uvm.). Dank dieser können verschiedene Nutzer damit arbeiten ohne jedes Mal eine neue Sprache erstellen zu müssen.

Der Nutzen einer DTD ist also sehr groß. Allerdings sollte trotzdem überlegt werden, ob eine solche wirklich notwendig oder sinnvoll ist. Wenn das Dokument größer als die DTD wäre, ist es sicher eine bessere Wahl auf die zu verzichten. Gleiches gilt, wenn der Prozessor nicht fähig ist eine Gültigkeitsprüfung durchzuführen, was bei neueren Computern jedoch immer der Fall ist.

# Terminkalender

Im folgenden Abschnitt sollen die Entstehung, die Funktionen und die dafür geschriebene Software erläutert werden. Eine kleine, kürze Dokumentation ist auf dem GitHub-Repository[[4]](#footnote-4) zu finden. Hier werden alle Funktionen und Attribute kurz und bündig erklärt um das Verständnis beim lesen des Codes zu erhöhen, jedoch erfüllt die Seminararbeit den gleichen Zweck, nur ausführlicher.

## Produkt und Funktionen

## Beginn des Projekts

Die Planung ist bei einem Projekt dieser Größe maßgeblich. Neben der Planung der Programmierung und des Designs dürfen auf keinen Fall die Wünsche der Auftraggeber, Nutzer oder Kunden vernachlässigt werden.

Über die Aufgabenstellung beispielsweise eine persistente Datenspeicherung über XML verlangt. In Verbindung damit wurde von der Lehrkraft eine DTD erwünscht, damit das Projekt auf seiner eigenen Auszeichnungssprache basiert.

### User Stories

Da ein wirklicher Auftraggeber mit Wünschen in dem Sinne nicht vorhanden ist wurden Freunde und Verwandte gefragt, welche Funktionen bei einem familienfreundlichen Kalender gewünscht werden. Bei dieser Umfrage sind folgende Ergebnisse hervorgegangen:

Für die gefragten Personen war es wichtig, dass jeder seine Freizeit nutzten kann und diese somit fest in die Planung eingebunden ist. Dieser Wunsch wird durch die Möglichkeit wöchentliche Freizeitphasen oder spezifische Zeiten als Freizeit zu definieren. Möchte ein Nutzer jetzt einen solchen Termin belegen wird er informiert und gebeten dies zu regeln, bevor er den Termin dann setzten kann.

Eine weitere Funktion, die auf keinen Fall fehlen kann und in den Meisten Kalendarprogrammen Standard ist, sind Permanenttermine. So wird das Eintragung von wöchentlich sich wiederholenden Terminen deutlich einfacher gemacht und somit die Nutzerfreundlichkeit extrem gesteigert.

### Tasks und Prototypen

### Zeitliche Planung

Die zeitliche Planung der Seminararbeit fand im Rahmen der zweiten Zwischenaufgabe statt. Hier war genau dies gewünscht. Die Verfasser sollten die von ihnen bereits verfassten Meilensteine, Prototypen usw. mit Deadlines versehen. Das Resultat dieser Aufgabe war folgendes[[5]](#footnote-5):

## Zusammenarbeit der Klassen (Klassendiagramm)

Bei einem so großen Projekt mit vielen Klassen und Referenzen ist es sehr wichtig sich zu überlegen, nach welchem System man seine Klassen ordnen möchte. In diesem Fall wurde das XYZ -System verwendet. Hier gibt es drei Bereiche: die View- also die Visualisierung der Inhalt mit Hilfe einer GUI (Graphical User Interface /Benutzeroberfläche), die Logic- also die logischen Abläufe und die Verarbeitung und Auswertung der Daten, die über GUI oder die Datenspeicherung kommen und XYZ.

## DTD (eigene Sprache)

## Schnittstelle Java XML (Datenspeicherung)

## GUI mit JavaFX

# Einschätzung des Projekts (Fazit)

## Einschätzung des Projekts

## Einschätzung des Produkts

# Literaturverzeichnis

1. Aus „XML Standards“, von Tobias Hauser [↑](#footnote-ref-1)
2. <http://webkompetenz.wikidot.com/wp:auszeichnungs-programmiersprachen>, von Stefan Münz (deutscher Informatiker) [↑](#footnote-ref-2)
3. Aus „XML für Dummies“, von Ed Tittel und Frank Boumphrey [↑](#footnote-ref-3)
4. GitHub-Wiki: https://github.com/Reiswaffl/Kalendar/wiki [↑](#footnote-ref-4)
5. Trelloboard zur zeitlichen Planung: https://trello.com/b/Hg1mSbSR/w-seminar-familienkallender-eric-brendel [↑](#footnote-ref-5)