OneTutor®
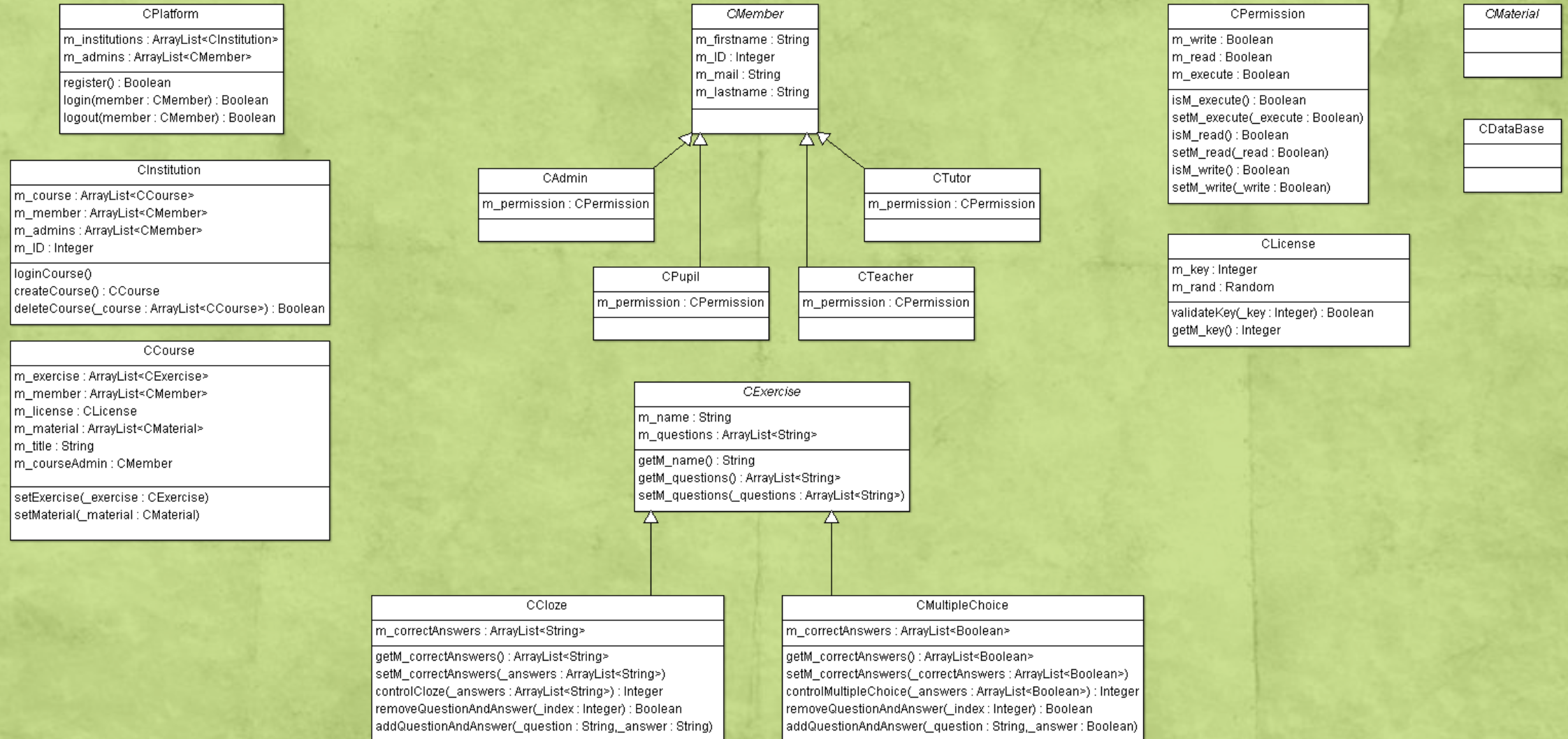
by OneTwoTech®

OneTutor®

- Nachhilfe- und Lernsoftware

- Zielgruppe: Schüler & Studenten

- Hilfe durch Nachhilfelehrer

UML

CPlatform

m_institutions : ArrayList<CInstitution>
m_admins : ArrayList<CMember>
register() : Boolean
login(member : CMember) : Boolean
logout(member : CMember) : Boolean

CInstitution

m_course : ArrayList<CCourse>
m_member : ArrayList<CMember>
m_admins : ArrayList<CMember>
m_ID : Integer
loginCourse()
createCourse() : CCourse
deleteCourse(_course : ArrayList<CCourse>) : Boolean

CCourse

m_exercise : ArrayList<CExercise>
m_member : ArrayList<CMember>
m_license : CLicense
m_material : ArrayList<CMaterial>
m_title : String
m_courseAdmin : CMember
setExercise(_exercise : CExercise)
setMaterial(_material : CMaterial)

CMember

m_firstname : String
m_ID : Integer
m_mail : String
m_lastname : String

CAdmin

m_permission : CPermission

CPupil

m_permission : CPermission

CTutor

m_permission : CPermission

CTeacher

m_permission : CPermission

CExercise

m_name : String
m_questions : ArrayList<String>
getM_name() : String
getM_questions() : ArrayList<String>
setM_questions(_questions : ArrayList<String>)

CCloze

m_correctAnswers : ArrayList<String>
getM_correctAnswers() : ArrayList<String>
setM_correctAnswers(_answers : ArrayList<String>)
controlCloze(_answers : ArrayList<String>) : Integer
removeQuestionAndAnswer(_index : Integer) : Boolean
addQuestionAndAnswer(_question : String,_answer : String)

CMultipleChoice

m_correctAnswers : ArrayList<Boolean>
getM_correctAnswers() : ArrayList<Boolean>
setM_correctAnswers(_correctAnswers : ArrayList<Boolean>)
controlMultipleChoice(_answers : ArrayList<Boolean>) : Integer
removeQuestionAndAnswer(_index : Integer) : Boolean
addQuestionAndAnswer(_question : String,_answer : Boolean)

CPermission

m_write : Boolean
m_read : Boolean
m_execute : Boolean
isM_execute() : Boolean
setM_execute(_execute : Boolean)
isM_read() : Boolean
setM_read(_read : Boolean)
isM_write() : Boolean
setM_write(_write : Boolean)

CLicense

m_key : Integer
m_rand : Random
validateKey(_key : Integer) : Boolean
getM_key() : Integer

CMaterial

CDataBase

Klassen

Plattform

Variablen
- ArrayList CInstitution a_institutions
- ArrayList CMember a_admins

Methoden
- boolean register()
- boolean login(Member _member)
- boolean logout(Member _member)

Institution

Variablen
- ArrayList CCourse a_courses
- ArrayList CMember a_member
- ArrayList CMember a_admins
- Int m_ID

Methoden
- void loginCourse(CCourse _course)
- CCourse createCourse()
- boolean deleteCourse(CCourse _course)

Klassen

Kurs

Variablen
- ArrayList CExercise a_exercises
- ArrayList CMaterial a_materials
- ArrayList CMember a_member
- CLicense m_license
- String m_name
- CMember m_admin

Methoden
- void setExercise(CExercise _exercise)
- void setMaterial(CMaterial _material)

Member

Variablen
- String m_firstname
- String m_lastname
- Int m_ID
- String m_mail

Methoden
- Getter & Setter
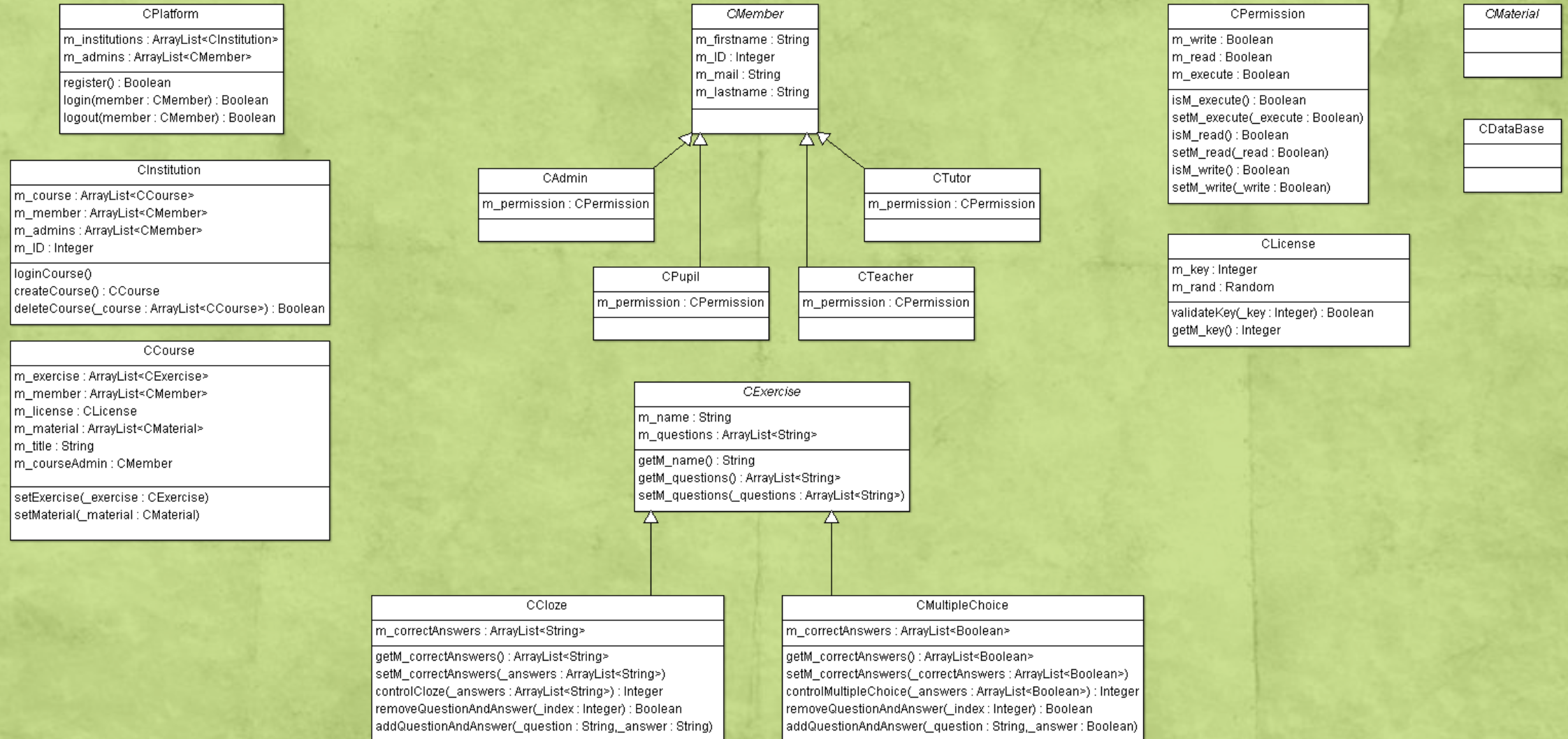
-> erbende Klassen:
- CAdmin, CPupil, CTeacher, CTutor (beinhalten Berechtigungen)

Übersicht am Beispiel

Plattform

Gymnasium Großdittmannsdorf

Schüler des Gymnasiums

Biologie · Deutsch · Informatik · Mathe

Lernvideos · Arbeitsblatt · Aufgaben · Probeklausuren

UML

CPlatform

m_institutions : ArrayList<CInstitution>
m_admins : ArrayList<CMember>

register() : Boolean
login(member : CMember) : Boolean
logout(member : CMember) : Boolean

CInstitution

m_course : ArrayList<CCourse>
m_member : ArrayList<CMember>
m_admins : ArrayList<CMember>
m_ID : Integer

loginCourse()
createCourse() : CCourse
deleteCourse(_course : ArrayList<CCourse>) : Boolean

CCourse

m_exercise : ArrayList<CExercise>
m_member : ArrayList<CMember>
m_license : CLicense
m_material : ArrayList<CMaterial>
m_title : String
m_courseAdmin : CMember

setExercise(_exercise : CExercise)
setMaterial(_material : CMaterial)

CMember

m_firstname : String
m_ID : Integer
m_mail : String
m_lastname : String

CAdmin

m_permission : CPermission

CTutor

m_permission : CPermission

CPupil

m_permission : CPermission

CTeacher

m_permission : CPermission

CExercise

m_name : String
m_questions : ArrayList<String>

getM_name() : String
getM_questions() : ArrayList<String>
setM_questions(_questions : ArrayList<String>)

CCloze

m_correctAnswers : ArrayList<String>

getM_correctAnswers() : ArrayList<String>
setM_correctAnswers(_answers : ArrayList<String>)
controlCloze(_answers : ArrayList<String>) : Integer
removeQuestionAndAnswer(_index : Integer) : Boolean
addQuestionAndAnswer(_question : String,_answer : String)

CMultipleChoice

m_correctAnswers : ArrayList<Boolean>

getM_correctAnswers() : ArrayList<Boolean>
setM_correctAnswers(_correctAnswers : ArrayList<Boolean>)
controlMultipleChoice(_answers : ArrayList<Boolean>) : Integer
removeQuestionAndAnswer(_index : Integer) : Boolean
addQuestionAndAnswer(_question : String,_answer : Boolean)

CPermission

m_write : Boolean
m_read : Boolean
m_execute : Boolean

isM_execute() : Boolean
setM_execute(_execute : Boolean)
isM_read() : Boolean
setM_read(_read : Boolean)
isM_write() : Boolean
setM_write(_write : Boolean)

CLicense

m_key : Integer
m_rand : Random

validateKey(_key : Integer) : Boolean
getM_key() : Integer

CMaterial

CDataBase

Kommende Schritte

- Ausarbeitung der Klassen CMaterial und CDatabase

- Programmierung restlicher Klassen mit Unit-Tests

Vielen Dank für Ihre Aufmerksamkeit