

Die `osgexam`-Klasse

Matthias Werner [matthias.werner@informatik.tu-chemnitz.de]

Version 0.2a

Zusammenfassung Die \LaTeX -Klasse `exam` von Philip Hirschhorn eignet sich exellent zur Erstellung von Klausurdokumenten mit \LaTeX . Die Klasse `osgexam` baut auf der `exam`-Klasse auf, passt sie an die Anforderungen für Klausuren an, die an der Professur für Betriebssysteme der TU Chemnitz existieren und stellt einige Erweiterungen sowie ein Buildsystem für verschiedene Versionen der Klausur zur Verfügung.

Inhaltsverzeichnis

1	Einführung	3
1.1	Zweck	3
1.2	Systemvoraussetzungen	3
1.3	Dokumenttypen und Modi	3
2	Klassens Optionen	4
3	Die verschiedenen Dokumenttypen	6
3.1	Aufgabenblatt: Modi <code>empty</code> , <code>single</code> und <code>dummy</code> . . .	7
3.2	Musterlösung: Modus <code>answers</code>	7
3.3	Seriendruck: Modus <code>series</code>	7
4	Macros zur Klausurgestaltung	8
4.1	Sprache	8

4.2	Titelseite	9
5	Einschränkungen, Probleme und Bugs	11
5.1	Einschränkungen gegenüber der <code>exam</code> -Klasse	11
5.2	Probleme	11
5.3	Bugs	12

1 Einführung

1.1 Zweck

Die `osgexam`-Klasse hat zwei Funktionen

1. Sie erweitert bzw. konfiguriert die `exam`-Klasse von Philip Hirschhorn entsprechend den Anforderungen der Professur Betriebssysteme der TU Chemnitz
2. Sie dient als Buildsystem für verschiedene Versionen einer Klausur, einschließlich der Generierung eines kompletten Klassensatzes von Klausuren aus den Daten einer Excel-Datei.

1.2 Systemvoraussetzungen

Die `osgexam`-Klasse verlangt Lua \LaTeX und läuft nicht mit pdf \TeX . Für den Seriendruck ist `latexmk` sowie ein Python3 mit installierten `openpyxl` sowie notwendig. Letzteres kann z. B. über

```
>pip3 install openpyxl
```

in der Shell installiert werden.

1.3 Dokumenttypen und Modi

Mit dieser Klasse können verschiedene Arten von Dokumenten erzeugt werden, die mit verschiedenen Modi verbunden sind:

- Nichtpersonalisiertes (also leere) Aufgabenblatt (Modus `empty`)
- Personalisiertes Aufgabenblatt mit vorgegebenen Daten (Modus `single`)
- Personalisiertes Aufgabenblatt mit Dummy-Daten (Modus `dummy`)
- Serie von personalisierten Aufgabenblättern entsprechend den Daten in einer Exceldatei (Modus `series`)
- Musterlösung (Modus `answers`)

Es ist vorgesehen, den jeweiligen Modus interaktiv zu setzen. Während der Entwicklung einer Klausur ist die Interaktivität vermutlich störend,

daher kann sie mit der `developmet`-Option (siehe 2) abgeschaltet werden. Jedoch stehen dann einige der Modi nicht mehr zur Verfügung.

Um auch während der Entwicklung auf diese Modi zurückgreifen zu können, kann in Quelldokument noch vor `\documentclass` der jeweilige Modus mit einem `TEX`-Befehl definiert werden. Beispielsweise kann der `series`-Modus auf folgende Weise unter Umgehung des Auswahlmenüs erzwungen werden:

```
\def\mode{series}
\documentclass{osgexam}
% ...
```

2 Klassenoptionen

Die Klasse übernimmt alle Options von `exam`, fügt aber eine Reihe neuer hinzu.

Modi und Dateien

- development** `true, false, answers` (true)
Der Buildmechanismus wird bei den Werten `true` oder `answers` deaktiviert; das Dokument kann also wie ein normales `LATEX`-Dokument ohne Nutzerinteraktion übersetzt werden. Der Wert `answers` entspricht dem Setzen der `exam`-Option `answers`, es wird also die Musterlösung generiert.
Wird die Option nicht oder auf den Wert `false` gesetzt, wird interaktiv die Art des zu erzeugenden Dokumentes abgefragt (siehe 1.3).
- prefix** `<Prefix>`
Setzt den Prefix für die Ausgabedatei(en). Da der Dateiname einer Ausgabedatei durch den Dokument`typ` (siehe 1.3) festgelegt wird, hilft das Setzen eines Prefix den Zusammenhang mit dem Quelldokument herzustellen.
- xlsfile** `<Excel-Datei>`
Excel-Datei für die Seriengenerierung von einem Klassensatz von Aufgabenbättern, siehe 3.3.

outpath *⟨Pfad⟩* (pdf)

Dateipfad zu einem Verzeichnis für die generierten Dateien (außer im Developer-Mode). Damit kann das Verzeichnis mit den Aufgabenstellungen von den generierten Dateien freigehalten werden. Dies kann insbesondere bei der Seriengenerierung die Übersichtlichkeit steigern.

Der Pfad wird relativ zum aktuellen Arbeitsverzeichnis interpretiert. Wenn ein absoluter Pfad angegeben werden soll, muss er wie üblich mit einem Schrägstrich (/) beginnen.

Sprache

lang *de, en, {de, en}, {en, de}* (de)

Legt die Sprache(n) des Dokuments fest. Es sind nur Deutsch (**de**) und Englisch (**en**) vorgesehen. In Kombination mit den Sprachmacros (siehe 4.1) können aus der sonst gleichen Quelldatei verschiedene Sprachvarianten generiert werden.

quotes *babelshorthands, ascii, latex* (babelshorthands)

Mit dieser Option wird gesteuert, wie Anführungszeichen eingegeben werden sollten.

babelshorthands Anführungszeichen werden als Shorthand entsprechend der (deutschen) Einstellung für **babel** erwartet, also als "```" / "`'`". Babel-Shorthands funktionieren unabhängig von der Spracheinstellung, also auch in englischen Textabschnitten.

ascii Anführungszeichen werden als einfache ASCII-Anführungszeichen erwartet ("`"`"). D

latex Der Standardmechanismus von \LaTeX ("```" / "`'`") wird angenommen. Diese Einstellung erfolgt auch, wenn ein sonstiger Wert gewählt oder die Option ganz weggelassen wird.

*(Note: Bei Wahl von **babelshorthands** oder **ascii** wird (ähnlich wie in **babel**) der Kategoriecode der Anführungszeichen geändert, was ggf. mit anderen Paketen interferieren kann.)*

Gestaltung der Titelseite

solutiontime *⟨Zeit in Minuten⟩* (90)

Legt die zur Verfügung stehende Lösungszeit fest. Der hier angegebene Wert wird im Standardtitelblatt verwendet. Statt als Option kann die Lösungszeit auch über das Macro `\solutionTime` in der Preamble gesetzt werden.

`passedwith` *⟨Punktegrenze für deas Bestehen in %⟩* (50)

Es wird festgelegt, ab wieviel Prozent der Punkt die Klausur bestanden ist. Der berechnete Wert wird auf der Titelseite angegeben.

`seat` Auf der Titelseite der Klausur wird der Sitz mit angegeben. Wir eine Klausur parallel in mehreren Räumen geschrieben, empfiehlt es sich, hier (auch) den Raum anzugeben.

`impolite` Die Willkommens-Formel und der Erfolgswunsch wird unterdrückt. Damit kann ggf. etwas Platz auf der Titelseite der Klausur gespart werden, falls zusätzliche oder geänderte Klausurregeln (siehe 4.2) zusätzlichen Platz beanspruchen.

3 Die verschiedenen Dokumenttypen

Wenn ein Dokument mit der `osgexam`-Klasse übersetzt wird, ohne dass die `development`-Option gesetzt ist, erscheint im Terminal ein Textmenü:

```
*****
No mode provided.
Please enter a proper mode (default is 'e')
- [e]mpty exam
- [a]nswers (solution)
- [p]ersonalized exam series from MS Excel file
- [i]ndividually personalized exam
- [d]ummy personalized exam

- [q]uit

\mode =
```

Durch Auswahl des entsprechenden Eintrags wird die Übersetzung erneut – diesmal im ausgewählten Modus – gestartet.

3.1 Aufgabenblatt: Modi `empty`, `single` und `dummy`

Bei den Modi `empty`, `single` und `dummy` wird jeweils eine einzelne Klausur generiert. Die Modi unterscheiden sich durch die Personalisierung der Klausur. Während in `empty` keine Personalisierung vorgenommen wird, wird in `dummy` die Klausur mit generischen Daten (Max Mustermann, ...) und im `single` (Menüpunkt “individually personalized exam”) mit konkreten Personendaten personalisiert. Letztere werden interaktiv abgefragt.

Der Name der Ausgabedatei ist `<prefix>-<Studenten-ID>.pdf` in Fall von `single`, und `<prefix>-dummy.pdf` bzw. `<prefix>-empty.pdf` im Fall von `single` bzw. `empty`. Die Ausgabedatei wird ins Verzeichnis `<outpath>` geschrieben.

3.2 Musterlösung: Modus `answers`

In diesem Modus wird eine Musterlösung erstellt, ähnlich wie mit der `answers` in der Klasse `exam`. Die Musterlösung enthält auf allen Seiten ein Wasserzeichen mit dem Wort “Musterlösung” bzw. “Standard Solution”, je nachdem welches die (primäre) Sprache des Dokuments ist.

3.3 Seriendruck: Modus `series`

`xlsfile`

Um den Seriendruck von personalisierten Klausuren zu nutzen (Modus `series`), muss eine Excel-Datei mit in der `xlsfile`-Option angegebenen Namen und den entsprechenden Daten vorbereitet werden.

Die Datei muss auf dem ersten Arbeitsblatt eine Tabelle mit den Identifikationsnummer der Studierenden, ihren Namen und Vornamen, sowie ggf. den zugewiesenen Platznummern enthalten. Die Spalten der Tabelle werden über einen Tabellenkopf identifiziert, wobei die Kopfeinträge unterschiedlich lauten können:

- Identifikationsnummer: “*Matrikelnummer*”, “*Matr-Nr*”, “*id*”, “*student id*”
- Name: “*Name*”, “*surname*”, “*family name*”

- Vorname: “*Vorname*”, “*first name*”, “*forename*”, “*given name*”
- Platz: “*Platz*”, “*Sitz*”, “*seat*”, “*place*”

Es werden alle Einträge unter der Kopfzeile bis zur ersten Leerzeile verarbeitet, d. h. entsprechend viele Dateien in `<outpath>` angelegt. Diese sind wieder nach dem Schema `<prefix>-<Studenten-ID>.pdf` benannt. Man beachte, dass das Erzeugen eines Klassensatzes von Klausuren durchaus einige Zeit in Anspruch nehmen kann.

4 Macros zur Klausurgestaltung

4.1 Sprache

Die `osgexam`-Klasse stellt den Erstellern von Klausuren eine Reihe von Macros zur Sprachunterstützung bereit.

`\deen` `{<deutsche Textvariante>}{<englische Textvariante>}`

Wenn nur eine Sprache bei der `lang`-Option angegeben ist, wird diese ausgegeben. Sind zwei Sprachen ausgewählt, wird zunächst die Erstsprache ausgegeben und anschließend die Textvariante der Zweitsprache in der Folgezeile in einer alternativen Farbe.

`\sdeen` `{<deutsche Textvariante>}{<englische Textvariante>}`

Dieses Macro funktioniert wie **`\deen`**, nur dass statt eines Zeilenumbruchs ein Schrägstrich (“/”) die Textvarianten trennt.

`\ldeen` `{<deutsche Textvariante>}{<englische Textvariante>}`

Dieses Macro gibt *nur* die Variante in der Erstsprache (bzw. der einzigen gewählten Sprache) aus.

`\cdeen` `{<deutsche Textvariante>}{<englische Textvariante>}`

Dieses Macro funktioniert wie **`\ldeen`**, richtet seine Ausgabe aber nicht nach der Erstsprache des Dokuments, sondern nach der derzeit umgebenden Sprache. Es kann also *innerhalb* anderer Sprachmacros verwendet werden und gibt stets die Sprache seines Kontextes aus.

\xdeen $\{\langle\textit{deutsche Textvariante}\rangle\}\{\langle\textit{englische Textvariante}\rangle\}$

Das Macro **\xdeen** funktioniert ähnlich wie **\deen**, kann aber noch eine variable Anzahl weiterer Argumente haben. In den jeweiligen Sprachvarianten wird nach dem Vorkommen von “@” gefolgt von einer Zahl $\langle i \rangle$ gesucht, und jeweils durch das i . Argument ersetzt. Das Macro dient dazu, Aufgabenparameter in den verschiedenen Sprachvarianten einheitlich zu halten. Der Quellcode...

```
\documentclass[lang={en,de}]{osgexam}
% ....
\xdeen{Gegeben seien die zwei Koordinaten (@1) und
(@2).}{Given two coordinates (@1) and (@2).}{1,0}{0,3}
```

ergibt beispielsweise

Given two coordinates (1,0) and (0,3).
Gegeben seien die zwei Koordinaten (1,0) und (0,3).

Die Ergebnis der Verwendung von Parameternnummern, die nicht-existierenden Argumenten entsprechen, ist nicht definiert.

4.2 Titelseite

Die **osgexam**-Klasse legt automatisch eine Titelseite an, so dass sich ein Klausurautor nicht mehr um deren Layout kümmern braucht. Die Titelseite erlaubt es, die Klausur in einen A4-Umschlag mit Adressfenster zu stecken so dass die Personalisierung im Adressfenster zu sehen ist.

Die Titelseite besteht aus folgenden Elementen:

- Punkte-/Bewertungstabelle, in die die möglichen Punkte für jede Aufgabe angegeben sind und in die die erreichten Punkte bei der Bewertung handschriftlich eingetragen werden können.
- Die Personalisierung mit Studenten-ID, Name, Semester und ggf. Sitzplatz
- Das Fach, in dem die Klausur geschrieben wird
- Eine Liste mit Regeln, die bei der Prüfung zu beachten sind

Durch die **osgexam**-Klasse werden einige Macros zur Verfügung gestellt, um den Inhalt der Titelseite zu modifizieren. Sie müssen in der Preamble genutzt werden.

\course [*\langle Abkürzung \rangle*] {\i \langle Kursname \rangle}

Der Name der Lehrveranstaltung und ggf. eine Abkürzung wird gesetzt. Falls nötig, kann der dieser Name mit **\insertCourseName** und die Abkürzung mit **\insertCourseAcronym** in der Klausur genutzt werden.

\term [*\langle Semester-Kurzform \rangle*] {\i \langle Semester \rangle}

Dies definiert das Semester, in dem die Klausur geschrieben wird, sowie ggf. eine Kurzform. Falls benötigt, kann auf das hier definierte Semester in der Klausur mit **\insertCourseSemester** und auf seine Kurzform mit **\insertCourseSemesterShort** zurückgegriffen werden.

\solutionTime {\i \langle Zeit \rangle}

Dies setzt die Lösungszeit, die jedoch auch über die Option **solutiontime** eingestellt werden kann.

\addextrapoints [*\langle Bezeichnung \rangle*] {\i \langle Punkte \rangle}

Nimmt in den Punktesspiegel Zusatzpunkte auf, die keiner einzelnen Aufgabe der Klausur zugeordnet werden können, beispielsweise Punkte einer Vorleistung oder Extrapunkte für Form etc.

\addFrontPageRule {\i \langle Regeltext \rangle}

Dieses Macro fügt eine Prüfungsregel auf der Titelseite hinzu. Durch die **osgexam**-Klasse sind bereits Standardregeln vorgegeben, die aber modifiziert werden können. Dazu sind die Regeln intern durchnummeriert, beginnend mit 1. Mit Hilfe von **\replaceFrontPageRule**{\i \langle Nummer der Regel \rangle}{\i \langle Regeltext \rangle} kann eine Regel modifiziert werden, und mit Hilfe von **\removeFrontPageRule**{\i \langle Nummer der Regel \rangle} ganz gelöscht werden. Gelöschte Regeln werden nicht mehr angezeigt, behalten aber ihre Nummer, so dass sich weitere Modifikationen stets auf die ursprüngliche Nummerierung beziehen.

5 Einschränkungen, Probleme und Bugs

5.1 Einschränkungen gegenüber der exam-Klasse

Während die Klasse `exam` die eierlegende Wollmilchsau ist und vieles zulässt, ist `osgexam` auf einen bestimmten Einsatzfall festgelegt. Entsprechend können nicht alle Möglichkeiten von `exam` genutzt werden. Dieser Abschnitt zählt die wichtigsten Einschränkungen auf:

- Das Layout der Titelseite einschließlich der verwendeten Punkte-/Bewertungstabelle ist vorgegeben und kann (ohne größeren Aufwand) nur im Rahmen der Klassenoptionen und der im Abschnitt 4.2 beschriebenen Macros geändert werden.
- `osgexam` gibt das Layout für Fragen vor, hat einen eigenen Punktmechanismus und ein angepasstes Format für die Punktedarstellung. Entsprechend können Punkte nur Zahlen sein. D. h. z. B., dass eine Prozentangabe für eine Aufgabe (wie auf Seite 37 der `exam`-Dokumentation dargestellt) nicht möglich ist. Auch unterstützt `osgexam` keine halben Punkte.¹

5.2 Probleme

Seit MikTeX/MacTeX2020 funktioniert `kvoption-patch` nicht mehr, das dazu diente, auch komplexere Klassenoptionen mit `keyval/kvoptions` zu parsen. Daher wurde `osgexam` auf `pgfkeys/pgfopts` umgestellt. Die Version 0.1 läuft jedoch nicht auf neueren TeX-Installationen.

Die `osgexam`-Klasse ruft andere Programme auf, namentlich den Python-Interpreter und `latexmk`. Daher muss die Übersetzung stets mit der Aktivierung der Shell-Escape-Funktion vorgenommen werden, also z.B.:

```
>lualatex -shell-escape myexam.tex
```

Alternativ kann `latexmk` mit der im Paket zur Verfügung gestellten `latexmkrc`-Datei genutzt werden:

```
>latexmkrc myexam.tex
```

¹Die Notwendigkeit, halbe Punkte zu vergeben, deutet ohnehin in der Regel auf ein schlechtes Klausurdesign hin.

Warning:

Werden andere L^AT_EX-Übersetzungen in dem Verzeichnis durchgeführt, in dem sich die zur Verfügung gestellte `latexmkrc`-Datei befindet, wird dafür ebenfalls die Shell-Escape-Funktion aktiviert. **Dies kann ein Sicherheitsrisiko darstellen.**

5.3 Bugs

Ich bin sicher, dass die Klasse eine ganz Reihe von Fehlern hat. Zum Zeitpunkt der Erstellung dieser Dokumentation ist mir jedoch keiner bekannt. Nutzer der Klasse werden gebeten, für gefundene Bugs ein Issue auf GitHub anzulegen: <https://github.com/tuc-osg/osg-exam/issues>