

# Übung zur Benutzung der TUDaExercise-Klasse

Marei Peischl



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

Sommersemester 2042 – v3.38 (2024-05-07)  
Übungsblatt 5

---

## Aufgabe 5.1: Titelei

---

Im Gegensatz zu den ganzen Titelseiten bei TUDaPub benötigt TUDaExercise eine platzsparende Möglichkeit für die Titelerzeugung. Daher verfügt TUDaexercise nur über einen Titelkopf. Für die Übergabe der Daten werden die folgenden Makros ausgewertet:

```
\title[Kurztitel für die Kopfzeile]{Titel}  
\subtitle{Untertitel}  
\author{Autor*in/Dozent*in}  
\term{Semester}  
\date{Datum}  
\sheetnumber{Nummer des Übungsblatts}
```

Mit Ausnahme des Titels können alle Felder auch leer bleiben. Sofern keine Blattnummer übergeben wird, werden die Übungsaufgaben beginnend mit 1, fortlaufend nummeriert.

---

## Aufgabe 5.2: Farbgebung

---

Die Farbgebung nutzt die selben Mechanismen wie die übrigen Klassen des TUDa-CI-Bundles. Die Optionen `color`, `accentcolor`, `textaccentcolor`, `identbarcolor` funktionieren wie gewohnt (vgl. TUDaPub). Darüber hinaus ist es auch hier möglich den farbigen Titelblock zu deaktivieren. Die Option `colorback=false` ermöglicht es für Versionen zum Ausdrucken die Tonersparende Variante einzustellen.

---

## Aufgabe 5.3: Eine Aufgabe erstellen

---

Aufgaben werden bei `tudaexercise.cls` durch die Umgebung `task` definiert. Das Notwendige Argument übernimmt dabei eine Überschrift. Es kann leer bleiben.

```
\begin{task}[<Optionen>]{Überschrift}  
  Text der Aufgabe  
\end{task}
```

Querverweise zwischen den Aufgaben sind wie üblich mit `\label` und `\ref` möglich.

### Die Verwendung des Makros `\task`

Neben der Umgebungsstruktur, verfügt `tudaexercise` auch über die Möglichkeit eine Aufgabe nur mit `\task` einzuleiten. Dieser Mechanismus existiert um zu älteren Versionen Kompatibel zu bleiben. Sie sollte jedoch nach Möglichkeit nicht mehr verwendet werden!

---

**Aufgabe 5.4: Teilaufgaben**

---

---

5.4a) Grundstruktur

---

tudaexercise unterstützt mit der `subtask`-Umgebung Aufgaben in Teilaufgaben zu untergliedern. Es wird derzeit nur eine Ebene unterstützt. Für weitere Untergliederung ist es möglich das Dokument zusätzlich mit `\section` in Abschnitte zu unterteilen.

```
\begin{subtask}[title=Titel, points=5]
```

```
\end{subtask}
```

Teilaufgaben benötigen – im Gegensatz zu Aufgaben – keine Überschrift. Daher unterscheidet sich die Syntax für die Übergabe von der `task`-Umgebung. Es gibt jedoch eine Variante um diese Unterscheidung zu umgehen, vergleiche Aufgabe 5.10.

---

5.4b) Anpassung der Überschriften

---

Ab Version 3.13 unterstützt tudaexercise zusätzlich eine Umschaltung bezüglich des Layouts der Subtask-Überschriften. Es existieren drei Optionswerte für die Dokumentenklasse:

**subtask=ruled** Voreinstellung, die Überschriften sind durch Linien vom Fließtext abgetrennt. (vgl. dieses Dokument)

**subtask=plain** Keine Linien um die Überschrift.

**subtask=runin** Falls sehr kleine Aufgaben gesetzt werden, kann der Umbruch nach der Überschrift unterbunden werden. In diesem Fall ist die Nutzer:in selbst dafür verantwortlich, dass eine ggf. notwendige Optische Abtrennung zwischen Titel und Aufgabentext stattfindet, z.B. durch Hervorhebung des Titeltexes.

---

**Aufgabe 5.5: Kopfzeilenanpassungen**

---

Der Mechanismus für die Kopfzeilen wurde angepasst, sodass sein Inhalt frei wählbar ist.

```
\ConfigureHeadline{
  headline={Inhalt der Kopfzeile}
}
```

Es existiert nun eine Unterscheidung zwischen linken und rechten Seiten, sofern `twoside=true`.

```
\ConfigureHeadline{
  even={Inhalt der Kopfzeile für linke Seiten},
  odd={Inhalt der Kopfzeile für rechte Seiten},
  oneseite={Inhalt der Kopfzeile im einseitigen Modus}
}
```

Einstellungen, die gleiche Felder betreffen, überschreiben sich gegenseitig.

5.5a)

---

Die Inhalte sind frei wählbar und werden automatisch in einer Box mit der gleichen Breite, wie der Text gesetzt. Der Inhalt wird linksbündig platziert. Es ist möglich Grafiken oder Tabellen innerhalb dieser Box zu platzieren.

---

5.5b)

---

Für die Kopfzeilen existieren einige vorgefertigte Elemente, die dort platziert werden können.

```
\ShortTitle
\StudentID
\StudentName
```

Darüber hinaus kann auch jeglicher  $\LaTeX$ -Content dort eingefügt werden.

Zusätzlich sind einige Werte vorbelegt und platzieren die Einträge nach dem selbsterklärenden Namensschema: `title-name-id`, `title-name`, `title`, `name-id`, `name`

---

5.5c)

---

Da die Kopfzeile häufig für die Angabe von Studentendaten genutzt wird, ist es mit der Klassenoption `headontitle` auch möglich die Kopfzeile ebenfalls auf der Titelseite einzufügen.

---

### **Aufgabe 5.6: Lösungen**

---

tudaexercise verfügt über einen Mechanismus um Lösungsvorschläge innerhalb der Dateien mit zu verwalten. Hierfür existiert die Umgebung `solution`.

Die Ausgabe der Lösungen wird entweder global

```
\documentclass[... ,
  solution=true,
]{tudaexercise}
```

oder lokal für einzelne Aufgaben konfiguriert:

```
\begin{task}[solution=true]{Überschrift}
Aufgabentext
  \begin{solution}
  Lösungstext
  \end{solution}
\end{task}
```

Die Voreinstellung ist `solution=false`.

---

Lösungsvorschlag:

---

Beispiel für einen Lösungsvorschlag

**Aufgabe 5.7: Abfrage der Einstellung „solution“**

Ab Version 3.04 existiert zusätzlich die Möglichkeit die Einstellung `solution` abzufragen.

Die Umgebung `solution*` verhält sich wie `solution` ohne die Beschriftung. Darüber Hinaus wurden die Makros ergänzt.

```
\IfSolutionT{<falls solution=true>}
\IfSolutionF{<falls solution=false>}
\IfSolutionTF{<falls solution=true>}{<falls solution=false>}
```

In der aktuellen Einstellung (`solution=false`) erzeugt der obige Block die folgende Ausgabe:

```
<falls solution=false> <falls solution=false>
```

**Aufgabe 5.8: Punkte (10)**

Punkteangaben sind über den Optionsschlüssel `credit` möglich. Der Wert wird hinter den Aufgabentitel in runde Klammern gesetzt. Dies geschieht mithilfe des Makros `\creditformat`

Alternativ existiert der Schlüssel `points` dieser ergänzt zusätzlich zur Angabe auch den Text `\PointName` bzw. `\PointsName` je nachdem ob der Wert gleich 1. Der Schlüssel `points` akzeptiert daher nur numerische Werte ab Version 3.13 werden Gleitkommazahlen unterstützt. Das Dezimaltrennzeichen ist hierbei ein Punkt. Korrektur auf ein Komma ist über eine Umdefinition von `\pointformat` möglich.

**5.8a) Beispiel für den Schlüssel `points` (1 Punkt)**

Da ältere Versionen von TUDaExercise keine Optionsverarbeitung für `subtask` erlaubten, ist hier ein zusätzlicher Mechanismus enthalten. Falls sowohl ein Titel, als auch spezifische Angaben notwendig sind, existiert hier zusätzlich der Schlüssel `title`. Geprüft wird darauf, ob das Argument ein Gleichheitszeichen enthält, ist dies der Fall, so wird davon ausgegangen, dass Optionen nach Schlüssel=Wert-Struktur übergeben werden.

**5.8b) Fallback-Modus, falls ein `=` im Titel detektiert wird, aber keine Zuweisung möglich ist.**

Falls bei der Optionsverarbeitung Unstimmigkeiten festgestellt werden, so wird das gesamte Argument als Titel verarbeitet.

**Aufgabe 5.9: Automatismen für Punktetabellen (Gesamt: 8 Punkte)**

Mit Version 3.13 erhält `tudaexercise` die Funktionalität für automatische Punktetabellen und Referenzen auf die Punkteangaben.

**5.9a) Referenzen aktivieren (3 Punkte)**

`tudaexercise` hat diese Funktionalität grundsätzlich deaktiviert, sie kann jedoch über die Klassenoption `points=true` eingeschaltet werden.

Der Mechanismus für die Referenzierung funktioniert ähnlich, wie die Objektreferenzen bei  $\LaTeX$ . Dadurch ist es notwendig das Dokument zweimal zu kompilieren. Falls `tudaexercise` im Durchlauf feststellt, dass sich die Werte des letzten Durchlaufes von den aktuellen unterscheiden, wird eine entsprechende Warnung erzeugt.

Für die Referenz stehen zwei Makros zur Verfügung. Eines funktioniert über den Aufruf der Aufgabennummer, der andere über `labels`:

```
\getPoints{5.9}
```

```
\refPoints{task:refPoints}% benötigt das \label{task:refPoints}, vgl. Quellcode
```

Beide Makros geben in diesem Fall den gleichen Wert: 8. Da die Variante `\refPoints` dynamisch von den Aufgabennummern abhängig ist diese Variante im Allgemeinen zu bevorzugen.

Neben der Referenz auf einzelne Aufgaben gibt `\getPointsTotal` die Summe der aller Aufgaben im letzten  $\LaTeX$ -Lauf zurück.

---

### 5.9b) Punkte aus Teilaufgaben aufsummieren (2 Punkte)

---

Der neue Mechanismus ermöglicht auch, dass der Punktwert einer Aufgabe aus der Summe der Punkte der enthaltenen Teilaufgaben berechnet wird.

Hierfür ist die Verwendung der Umgebungen zwingend notwendig. `\task` erlaubt es nicht die Teilaufgaben entsprechend einzugrenzen. Selbiges gilt für die Subtask-Umgebung.

Die Summe der Punkte wird nicht automatisch bei der Aufgabenbezeichnung eingetragen, jedoch existiert mit dem Optionswert `points=auto` eine entsprechende Funktionalität. Falls die Klassenoption `points=auto` gesetzt ist, wird diese automatisch für alle Aufgaben aktiviert. Der automatische Wert kann jedoch über eine manuelle Angabe eines Wertes überschrieben werden, z. B. falls eine Aufgabe nicht aus Teilaufgaben besteht.

Es erfolgt derzeit keine automatische Prüfung auf die Übereinstimmung zwischen der Summe der Teilaufgaben und der Gesamtpunktzahl.

Um die Unterscheidung der Angabe zwischen aufsummierten Werten und manuell gesetzten Werten zu ermöglichen existiert intern eine weitere Variante des Makros `\creditformat`.

```
\newcommand*\creditformatsum}[1]{\creditformat{#1}}
```

Es ruft intern lediglich das Makro `\creditformat` auf. Für dieses Beispieldokument wurde jedoch die Bedeutung überschrieben, sodass im Titel dieser Aufgabe gekennzeichnet wird, dass der Punktwerte eine Summe der Teilaufgaben entspricht:

```
\renewcommand*\creditformatsum}[1]{\creditformat{Gesamt: #1}}
```

---

### 5.9c) Automatische Punktetabellen (3 Punkte)

---

`tudaexercise` stellt mit der Möglichkeit der Referenz auf die Aufgabenwerte auch die Auflistung aller Punkte zur Verfügung.

Da die Struktur einer Punktetabelle nach persönlichen Vorlieben stark variiert ist das Makro sehr abstrakt konstruiert.

```
\mapPoints{<Code, der für jede Aufgabe ausgeführt werden soll>}
```

`\mapPoints` iteriert über alle Aufgaben und erlaubt als Argument beliebige Code-Elemente. Innerhalb des Codes können `#1` (Nummer der Aufgabe) und `#2` (zugehöriger Punktwert) verwendet werden. Aufgaben, deren Wert Null beträgt werden übersprungen.

Zum Beispiel erzeugt:

```

\begin{tabular}{@{}lr@{}}
  \toprule
  \mapPoints{Aufgabe #1&#2 Punkte\\}
  \midrule
  Gesamt&\getPointsTotal{} Punkte\\
  \bottomrule
\end{tabular}

```

die Ausgabe:

Aufgabe 5.9	8 Punkte
-------------	----------

Gesamt	8 Punkte
--------	----------

Falls auch Aufgaben mit einem Wert von 0 verarbeitet werden sollen, existiert noch eine gesternte Version `\mapPoints*`.

---

### Aufgabe 5.10: Einheitliche Syntax für Task/Subtask

---

TUDaExercise unterscheidet in der Standardeinstellung zwischen `task` und `subtask`. Dies ist dadurch begründet, dass `subtask` keine Angabe eines Titels erfordert.

Es wurde der Bedarf nach einer einheitlichen Syntax geäußert (siehe [https://github.com/tudace/tuda\\_latex\\_templates/issues/189](https://github.com/tudace/tuda_latex_templates/issues/189)).

---

#### 5.10a) Die gesternten Varianten der task/subtasks-Umgebungen

---

Die mit Version 3.0 implementierten Varianten `task*` und `subtask*` ermöglichen dies.

Beide Varianten stehen nur als Umgebung und nicht als Makro zur Verfügung:

```

\begin{task*}{Aufgabentitel}
  \begin{subtask*}{Titel der Teilaufgabe}
    Beschreibung
  \end{subtask*}
\end{task*}

```

---

#### 5.10b) Vertauschen der beiden Varianten

---

Zusätzlich erlaubt die Klassenoption `match-task-subtask-syntax` das Vertauschen von `subtask` und `subtask*`.

Die Umgebungen `task*` und `task` sind identisch.

---

### Aufgabe 5.11: Kompatibilitätsmodus

---

Die Klasse `tudaexercise` verfügt über einen Kompatibilitätsmodus, um den Umstieg von zuvor existierenden Templates, wie `TUDexercise` zu erleichtern. Um den Kompatibilitätsmodus zu aktivieren, existiert die Option `compat=true`.

Damit können Strukturen wie `examheader`, `examheaderdefault` und die Erzeugung von Aufgaben mithilfe von `\subsection` wie gewohnt verwendet werden. Da einige dieser Mechanismen jedoch der Philosophie der semantischen Auszeichnung widersprechen wird ihre Verwendung hier nicht genauer erklärt. Nutzer, denen diese Mechanismen nicht vertraut sind, wird empfohlen die durch TUDaExercise zur Verfügung gestellten Varianten zu nutzen.

Beim zugehörigen Release im Development Repository findet sich auch eine Beispieldatei, die die alten Mechanismen verwendet und spezielle Hinweise für Umsteiger enthält: [https://github.com/tudace/tuda\\_latex\\_templates/blob/master/example/Examples-TUDexercise-compat.tex](https://github.com/tudace/tuda_latex_templates/blob/master/example/Examples-TUDexercise-compat.tex)