

Das Paket *hse1-uebungen*

Testat

Name, Vorname:
Matr.Nr.:

Datum:
Unterschrift:

1 Einführung

1.1 Motivation

Dieses Dokument dient als Beispiel zur Nutzung des Pakets `hse1-uebungen.sty`, welches entworfen wurde, um als Vorlage für Praktikums- als auch Prüfungsaufgabenblätter für L^AT_EX an der Hochschule Emden/Leer zu dienen.

Ein zentraler Bestandteil hierbei ist die Verwaltung von Aufgabennummerierungen und Punkten, die für einzelne Aufgaben und Teilaufgaben vergeben werden. Hierzu werden neue interne Zähler erzeugt, die unabhängig von Standardbefehlen zur Dokumentstrukturierung, wie beispielsweise `\section` oder `\subsection`, agieren. Dabei werden die Punkte zu allen Fragestellungen innerhalb einer Aufgabe summiert und am rechten Seitenrand ausgegeben. Zusätzlich wird eine Endsumme aus allen zu erreichenden Punkten gebildet. Dazu wird ein Zähler aus dem Paket `TotCount` verwendet.

Aufgabe 1 Erste Schritte 8

Um das Paket nutzen zu können, muss dieses mittels

```
\usepackage [en]{hse1-uebungen2015}
```

in das Dokument eingebunden werden. Der erste optionale Parameter definiert mit `{en,de}` die Sprache der Begrifflichkeiten im Dokument, wobei standardmäßig der Parameter `de` gesetzt wird. Als Basis dient vorzugsweise 3

```
\documentclass [a4paper,12pt]{article}
```

Durch das Kommando `\exercise` wird eine neue Aufgabe angelegt. Als optionaler Parameter kann hier ein Titel zur Aufgabe übergeben werden. Beispiel: 5

```
\exercise {Erste Schritte}
```

Aufgabe 2 5 Textkopf

Das Dokument kann mit einem Textbaustein versehen werden, der einen standardisierten Textkopf erstellt. Die Daten zum Textkopf mit Angabe zum Namen der Veranstaltung, Namen des oder der Dozenten, werden durch das Kommando `\setexheader` gesetzt und der Textkopf wird mittels `\printexheader` erzeugt. Beispiel:

```
\setexheader{Praktikum Mikrocomputersysteme SS2011}
{Prof. Dr. C.~Koch}{carsten.koch@hs-emden-leer.de}
{HS Emden/Leer, Fachbereich Technik \hfill Elektrotechnik und Informatik}
{\today}
\printexheader
```

Zur Dokumentation von erfolgreich bearbeiteten Praktikumsaufgaben (Testat) besteht mittels `\printreceipt` die Möglichkeit, einen weiteren Textblock zu generieren, der Angaben zum Studierenden und dem Prüfungsdatum enthält. Ein Beispiel hierzu ist auf der ersten Seite dieses Dokuments dargestellt. 5

Aufgabe 3 28 Aufsummieren der Punkte

Für jede Aufgabe kann mit Hilfe des Befehls `\addpoints` die Anzahl der zu erreichenden Teilpunkte festgelegt werden. Diese werden am rechten Rand der Zeile dargestellt. 4

Nach dem Aufruf von `\exercise` werden alle mittels `\addpoints` definierten Punkte zu dieser Aufgabe aufsummiert. 10

Alternativ besteht die Möglichkeit mehrere Aufgaben in dem Stil einer Auflistung zu erstellen. Dazu folgendes Beispiel:

1. Ich bin eine Aufgabe 2
2. Ich bin auch eine Aufgabe 3
3. Und ich bin eine weitere Aufgabe 3

Der Quelltext dazu:

```
\begin{enumerate}
\exitem{Ich bin eine Aufgabe}{2}
\exitem{Ich bin auch eine Aufgabe}{3}
\exitem{Und ich bin eine weitere Aufgabe}{3}
\end{enumerate}
```

Dabei ruft `\exitem` bereits die Funktion `\addpoints` auf, sodass die Punkte jeder Aufgabe automatisch zugeordnet werden.

Aufgabe 3.1 Unteraufgaben

Zusätzlich besteht die Möglichkeit Unteraufgaben zu definieren, um eine zusätzliche Hierarchie-Ebene einzufügen. Dazu kann der Befehl `\subexercise [<name>]` verwendet werden. Dieser besitzt lediglich einen optionalen Parameter, weshalb hier eckige Klammern verwendet werden. Bei diesem handelt es sich um den Namen der Aufgabe. So wurde diese Unteraufgabe mit folgendem Quellcode generiert.

```
\subexercise[Unteraufgabe]
```

Die in Unteraufgaben definierten Punkte werden derzeit nicht separat gezählt, sodass nachfolgende Aufrufe von `\addpoints` nach wie vor der mit `\exercise` zuletzt definierten Aufgabe zugeordnet werden - in diesem Beispiel [Aufgabe 3](#) und nicht [Aufgabe 3.1](#). 1

Aufgabe 3.2

Ein weiteres Feature ist die Berechnung der Summe aller vergebenen Punkte. In diesem Dokument wurden beispielsweise insgesamt 46 Punkte vergeben. Dieser Wert kann mit der Funktion `\printexpoints` ausgegeben werden. 5

Hierzu muss das Dokument jedoch drei Mal übersetzt werden. Im ersten Durchlauf werden die Punkte zu jeder Aufgabe aufsummiert und in die `.aux`-Datei geschrieben. Bei dem zweiten Durchlauf werden diese Werte ausgelesen und zu jeder Aufgabe aufsummiert. Im letzten Durchlauf werden die Punkte aller Aufgaben aufsummiert und sind im Dokument verfügbar.

Aufgabe 4 Verweise auf Aufgaben 5

Möchte man innerhalb des Textes auf eine Aufgabe oder Unteraufgabe verweisen, können dazu die beiden Befehle

1. `\exref {<number of exercise>}` 3
2. `\subexref {<number of exercise>}{<number of subexercise>}` 2

verwendet werden. Damit ist es bspw. möglich ein Verweis auf [Aufgabe 3.2](#) mittels `\subexref{3}{2}`

zu erstellen. Für diese Funktionalität werden automatisch Labels beim Aufruf von `\exercise` und `\subexercise` mit der jeweiligen Aufgabennummer angelegt.

Aufgabe 5 Formatierung

Zur einheitlichen Formatierung (*Typesetting*) häufig verwendeter Textelemente wurden eigene Befehle definiert:

<code>\tsetfile</code>	Dateinamen und Pfade
<code>\tsetprog</code>	Elemente einer Programmiersprache, also Variablen, Funktionen, etc.
<code>\tsetsw</code>	Elemente einer Software, also Menüeinträge, Parameter, etc.
<code>\tsettitle</code>	Titel von Dokumenten

In der jeweils übergebenen Zeichenkette dürfen sich Leerzeichen und Sonderzeichen befinden. Da zur Umsetzung der L^AT_EX-Befehl `\detokenize` verwendet wurde, wird jedoch jedem Backslash ein Leerzeichen hinzugefügt. Das folgende Beispiel

''Lesen Sie die ersten beiden Kapitel im Tutorial
\tsettitle{Introduction to the Qsys System Integration
Tool} von Altera. Öffnen Sie anschließend in Qsys
die Datei \tsetfile{nios_basic.qsys} im Verzeichnis
\tsetfile{../sopc intro/qsys} und erweitern Sie das
System mit einem zusätzlichen On-Chip-Speicher \dots''

wird wie folgt dargestellt:

''Lesen Sie die ersten beiden Kapitel im Tutorial *Introduction to the Qsys System Integration Tool* von Altera. Öffnen Sie anschließend in Qsys die Datei `nios_basic.qsys` im Verzeichnis `../sopc intro/qsys` und erweitern Sie das System mit einem zusätzlichen On-Chip-Speicher ...''