

中国地质大学（武汉）研究生课程论文

中国地质大学研究生课程论文

T_EX 模板

课 程 名 称: T_EX 模板教程

教 师 姓 名: 教师名字

学 生 姓 名: Timozer

学 号: 20131001333

所 在 院 系: 计算机学院

专 业: 计算机科学与技术

类 别: 硕士

2018 年 2 月 2 日

评 语

对课程论文的评语:

平时成绩:	课程论文成绩:
总 成 绩:	评 阅 人 签 名:

- 注: 1. 无评阅人签名成绩无效;
2. 必须用钢笔或圆珠笔批阅, 用铅笔阅卷无效;
3. 如有平时成绩, 必须在上面对评分表中标出, 并计算入总成绩.

摘 要

这是中国地质大学 (武汉) 研究生课程论文的 L^AT_EX 模板的说明文档. 主要对模板进行了介绍, 安装, 使用等方面的说明, 此外还写了模板开发过程的日志, 可以通过它来了解每一版中实现了哪些功能.

关键词: T_EX, L^AT_EX, 模板, 报告

目录

摘 要	i
第一章 介绍	1
1.1 TeX logos 测试	1
1.2 英文字体测试	1
1.3 这是代码测试	2
1.4 参考文献测试	2
1.5 插图环境测试	2
第二章 使用方法	4
2.1 安装	4
2.2 建立自己的文档	5
2.3 参考文献	5
第三章 实现细节	6
第四章 版本开发记录日志	7
4.1 Version 0.3	7
4.2 Version 0.4	7
4.3 Version 0.5	8
4.4 Version 0.6	11
参考文献	12
致谢	13
附录 A 测试附录	14

第一章 介绍

这份中国地质大学(武汉)研究生课程论文的 L^AT_EX 模板是我根据中国地质大学(武汉)研究生课程论文的 WORD 版来编写的. 值得一提的是, 在 WORD 版中只定义了封面的内容和下一页的评语表, 对其他格式并没有明确的要求, 因此, 在这份 L^AT_EX 模板中有关其他方面的格式都是没有参照标准的, 一切以我的审美为标准 :).

在此之前我好像并没有看到有同学写过地大的 T_EX 课程论文模板. 因此, 在此斗胆献丑. 才疏学浅, 难免有许多错误和不足的地方, 还望各位能够指出, 我会真心承认错误并积极改进 ^_^.

本模板的编码参考了很多其他大学的 T_EX 论文模板, 在此表示感谢. 目前仍然在开发中, 虽然用其写一个文档已经不成问题了, 但是还缺乏各种各样的让人用起来很舒服的功能. 因此, 希望有“牛人”可以添加新功能或是对本模板进行除错.

本模板旨在推广 T_EX 这一优秀排版软件在地大的应用, 为广大同学提供一个方便, 美观的课程报告模板, 减少写报告时候的各种各样格式方面的麻烦.

1.1 T_EX logos 测试

T_EX, L^AT_EX, L^AT_EX 2_ε, X_YT_EX, X_YL^AT_EX, LuaT_EX, LuaL^AT_EX, METAFONT, METAPOST, BibT_EX, A_MS-T_EX

1.2 英文字体测试

English Bold Fonts test

English Italic Fonts test

ENGLISHSMALLCAPSFONTS TEST

1.3 这是代码测试

看看间距

代码 1.1: cpp 代码

```
30 int sum(int _l, int _r)
31 {
32     return _l + _r;
33 }
```

看看间距

```
10 /* a + b = c
11     LaTeX
12 */
13 int main(){
14     return 0;
15 }
16
17 int abs(int a)
18 {
19     return a;
20 }
```

This is normal font size, \LaTeX {}

1.4 参考文献测试

这里是参考文献引用^[1].

1.5 插图环境测试



图 1.1: 1.jpg

第二章 使用方法

2.1 安装

本模板并不需要安装, 你可以在模板目录下写自己的报告文档, 也可以直接修改 `demo.tex` 文档 (当然这种方法不建议, 因为这个文档的内容在你以后需要的时候可以来查看)。

以下对本模板中的文件做一个说明:

cugrep.cls 这是本模板的文档类, 在使用模板的时候需要在导言区引入,
`\documentclass {cugrep}` ;

timozercode.sty 这是为方便代码引入而写的一个 code 宏包, 主要用于代码和伪代码的插入;

timozerfont.sty 这个文件用于文档中的字体设置, 中英文字体分开设置, 并且提供了从初号到小五的字体大小命令;

timozertexlogos.sty 这是宏包主要引入了提供 $\text{T}_{\text{E}}\text{X}$ logos 的宏包;

timozerbib.sty 对参考文献的格式做了一个设置;

demo.tex 本说明文档的 tex 源码文件;

README.md 本项目的说明文档;

Makefile make 程序使用的文件;

LICENSE 本项目的协议;

pyglangs.txt 代码支持的语言类型;

pygstyles.txt 代码高亮支持的风格。

2.2 建立自己的文档

2.3 参考文献

文献类型:

- article {author, title, journal, year}, {volume, number, pages, month, note}
- book {author/editor, title, publisher, year}, {volume/number, series, address, edition, month, note}
- booklet {title}, {author, howpublished, address, month, year, note}
- inbook {author/editor, title, chapter/pages, publisher, year}, {volume/number, series, type, address, edition, month, note}
- incollection {author, title, booktitle, publisher, year}, {editor, volume/number, series, type, chapter, pages, address, edition, month, note}
- inproceedings {author, title, booktitle, year}, {editor, volume/number, series, pages, address, month, organization, publisher, note}
- manual {title}, {author, organization, address, edition, month, year, note}
- mastersthesis {author, title, school, year}, {type, address, month, note}
- misc {none}, {author, title, howpublished, month, year, note}
- phdthesis {author, title, school, year}, {type, address, month, note}
- proceedings {title, year}, {editor, volume/number, series, address, month, organization, publisher, note}
- techreport {author, title, institution, year}, {type, number, address, month, note}
- unpublished {author, title, note}, {month, year}

第三章 实现细节

我想我可能会把这些实现细节写成博客文章发出来,不会在这里写了.

如果我写了这些文章,我会在这篇说明文档和 README.md 文档中更新状态的,敬请关注.

第四章 版本开发记录日志

4.1 Version 0.3

因为想不起版本 0.1–0.2 做了哪些功能, 所以在此就不介绍了.

这个版本主要是实现了中国地质大学 (武汉) 研究生课程论文 Word 模板中的封面和后面的评语表.

4.2 Version 0.4

目前在开发中, 暂时实现了摘要的添加. 因为这个 $\text{T}_\text{E}\text{X}$ 模板主要使用的是 `ctexbook` 文档类, 所以没有摘要这一环境, 暂时添加了摘要环境. 我不是太想使用摘要这个环境, 使用环境意味着摘要和关键词的格式需要自己调, 这很不舒服.

2018 年 1 月 26 日星期五 21 时 50 分 38 秒 CST

我新定义了一个命令 `\makeabstract`, 像 `\maketitle` 一样, 可以使用这个命令来生成摘要页面. 在使用之前, 你必须给变量 `\cugabstract`, `\cugkeywords` 赋值. 也许我可以起更好听的名字:).

2018 年 1 月 28 日星期日 17 时 01 分 20 秒 CST

为了使行文方便, 我重新定义了 `\frontmatter`, `\mainmatter`, `\backmatter` 这四个命令.

- `\frontmatter` 这个命令实现的功能是关闭 chapter 计数, 使用 roman 数字来标识页码;
- `\mainmatter` 这个命令实现的功能是打开 chapter 计数, 使用 arabic 数字来标识页码;
- `\backmatter` 这个命令实现的功能是关闭 chapter 计数.

折腾了一会儿页眉页脚, 包括首页页眉页脚. 但是成果比较感人, 首页之外的页面实现了页眉页脚的设置, 页眉左边为“中国地质大学 (武汉) 研究生

课程论文”, 右边为节标题, 节标题的格式有点儿不好, 需要改进. 另一个就是首页的页眉页脚设置好像有点儿问题, 只是让它在下面显示页码, 结果没有显示, 目前还不知道问题所在.

2018 年 1 月 29 日星期一 12 时 04 分 22 秒 CST

今天终于查到了首页页脚没显示的问题. 在 `\ctexset` 里面设置 `chapter` 的时候设置了页面的格式为 `empty`, 把后面的设置冲掉了. 现在更改为 `plain` 格式. 新的问题又来了, 首页页眉会显示一条横线, 看着不舒服, 去掉吧.

`ctex` 本身对 `appendix` 的设置就很不错了, 所以我这里就不对其重新定义了, 删除了原来对 `\appendix` 的重定义.

看了好多大学的论文模板, 发现附录一般都是放在最后的, 而参考文献, 致谢这些在附录的前面. 这就产生了一个问题. 在写文档的时候, 我们用 `\appendix` 命令来重置 `chapter` 计数, 并使用字母来表示章号, 而使用 `\backmatter` 来关闭 `chapter` 计数, 从而使 `\chapter` 命令产生的章标题 (参考文献, 致谢等) 不包含章号. 这样无形中的一个顺序使得结果和自己所想甚远. `\backmatter` 命令关闭 `chapter` 计数, `\appendix` 是不会打开的. 所以这里要重新定义 `\appendix`, 使得它会打开 `chapter` 计数.

0.4 版本开发搞一段落. 回顾一下该版本实现的功能:

- 添加了摘要命令 `\makeabstract`, 使得可以方便生成摘要;
- 定义了报告的写作顺序, `\frontmatter`, `\mainmatter`, `\backmatter`, `\appendix`, 严格按照这个顺序来定义报告的结构, 如果不按此来写, 很有可能不是你想要的结果;
- 重定义了 `\frontmatter`, `\mainmatter`, `\appendix` 等命令.

4.3 Version 0.5

2018 年 1 月 30 日星期二 17 时 07 分 04 秒 CST

今天正式开始开发 0.5 版本.

昨天在推送 0.4 版本后发现了一个问题, 字体配置有问题. 0.4 版本及之前的版本中, 字体配置是我自己在网上下载的字体, 而且也并没有将其安装到系统中, 只是将其放到了一个文件夹中, 然后使用 `fontspec` 宏包中的 `\defaultfontfeature` 命令指定其路径. 这就造成了只能使用该路径下的字体的问题. 关于如何添加多个路径以及把 `LaTeX` 默认搜索路径添加进来的操作, 我不会. 谷歌了很久都没有查到, 所以就放弃了, 如果你们谁知道如何解决这个问题, 烦请联系我, 不胜感激.

在放弃了那个方法后,我就使用了 \LaTeX 的默认搜索路径,并且把自己要使用的字体安装到系统中.这样,我就可以在 \LaTeX 中使用了.

这个问题解决后,突然发现又有新问题出现了(程序开发中解决一个 BUG 又产生了一堆新的 BUG 的即视感有木有 >_<).那就是英文字体中的加粗,斜体,小型大写字母命令不起作用了,无衬线和等宽字体的命令还能起作用,不是太清楚什么问题,在寻找中.

关于这个问题,我也是谷歌了很久的,烦.资料出奇的少.

最终找到一个类似的问题,他的解决方案是去掉了一个宏包,然后就可以了.我首先被这个人的想法惊到了,突然想起以前看过一篇文章,讲得是一个不懂电脑的电脑维修人员维修电脑的方法,他的修电脑方法和这个人类似,也是将电脑的零件各种拔插,本来的开不了机的电脑就可以开机了(ps.难道这种方法是计算机领域的一大特色^_^).

我决定采用他这种方法.首先不加入任何宏包,看这是不是 `ctexbook` 文档类本身的问题(ps.很明显是我想多了,人家那么多大神在维护,怎么可能出这种问题).显然不是,我又想会不会是我自己写的 `timozerfont.sty` 这个文件出了问题(ps.才疏学浅,出问题在所难免).在测试文档里引入了我的宏包,测试通过,并没有问题.那么就是我引入的宏包之间的冲突了.目前在排除中

....

在我将以下这些宏包关闭后,字体正常了.

- `{amstext, amsmath, amssymb, amsfonts, mathrsfs, bm, mathtools, newtx-text, courier}`
- `{graphicx, subcaption}`
- `{longtable, makecell, tabu, booktabs}`
- `{natbib}`
- `{timozercode}`, 这个是我自己写的一个宏包,用来方便输入代码的,具体用法应该在该模板大体完善的时候会写.

接下来一组一组的去开启这些宏包,使用排除法来找出产生冲突的宏包.

很幸运,在我开启第一组的时候就发现那几个命令工作不正常了.后面三组不用看了.

使用二分法,经过几次测试后,发现是 `newtxtext` 的问题.

我又对该宏包进行了单独测试,发现其和 `ctexbook` 文档类并不冲突,奇怪哉.难道是和我写的 `timozerfont.sty` 宏包有冲突?经过测试,确实是和该宏包冲突,在没有找到解决方法之前,我先把 `newtxtext` 宏包禁用了.

2018 年 2 月 1 日星期四 10 时 28 分 29 秒 CST

经过昨天的一番思想斗争以及查阅资料, 看宏包的说明文档, 我终于决定使用 minted 宏包来代替 lstlistings 宏包来实现代码插入.

在为代码添加背景色的时候, 发现 color 宏包不能正常使用, 换成了 xcolor 宏包.

今天实现了行内代码命令的定义, 提供了一个代码浮动体环境, 并且对代码环境的外观进行了设置. 基本任务完成, 如果有空, 我会考虑添加从文件中导入代码的功能.

2018 年 2 月 2 日星期五 12 时 17 分 51 秒 CST

昨天的代码环境有点儿小瑕疵, 就是不能自动换行, 今天修正这个纰漏.

0.5 版本开发告一段落, 在此对其做一个总结.

首先是字体配置的问题. 在开发前几个版本的时候没有注意到字体的问题, 主要是写文档的时候因为功能不多, 也就没怎么使用不同的字体, 这不, 一旦开始使用不同的字体时候问题就出来了. 主要的问题就是文档默认的中英文字体看着并不是很舒服, 其次就是一些字体命令竟然失效了.

文档默认的中英文字体在花费几个小时的比对后, 选出来了一些好看的并且是大部分文档都会采用的字体. 关于命令失效的问题, 找到了原因, 通过去掉一个宏包的方法修复好了.

接下来就是添加对代码的支持了. 因为我本身就是计算机专业的, 在写报告时候肯定会或多或少的需要在文档中展示一些代码, 这个时候就需要很方便, 而且不用我每次去查相关文档就能引入代码的方法了. 为此, 我基于 minted 宏包 (之前是 lstlistings) 重新写了一个简化的宏包 timozercode.

该宏包提供了行内代码的输入方式, 和代码段落以及代码浮动体的舒服方式, 基本的设置已经配好了, 只需要输入代码即可, 如果不满意我的设置, 可以自己去修改.

我们在写报告的时候不仅仅需要代码, 有时需要实现对伪代码的排版, 这个功能在该版本里面有了一个基本的设置, 你可以自己查看代码来找到使用的方法, 我还没有详细地去做这方面的工作, 下一版中应该会添加进来.

之后又添加了 $\text{T}_{\text{E}}\text{X}$ 系统的 logo 输入宏包, 该宏包主要就是预引入 $\text{T}_{\text{E}}\text{X}$ logos 的宏包, 方便大家在输入的时候不用关心其在哪个宏包里, 自己需要用到哪个宏包, 不用你操心, 我已经给你做了这个工作了, 没其他的作用,

最后就是将参考文献这个功能添加进来.

这就是 0.5 版本中实现的所有功能. 敬请期待下一版:).

4.4 Version 0.6

2018 年 2 月 2 日星期五 21 时 01 分 41 秒 CST

我试图为这个模板提供一些新的环境命令来方便我们输入内容, 比如说插图环境, 我想定义一个新的环境将插图环境包进去, 然后提供两个参数, 用来做插图的标题和标签, 这个环境自动将标题和标签放在闭合环境命令前面 (close stuff or end part), 很可惜, 我们并不能这么做, 具体原因我查到了, 如代码 4.1 所示.

代码 4.1: 为什么 newenvironment 的参数不能放在 close stuff 中

```
1 \newenvironment{foo}[1]%  
2 { ... open stuff ... }  
3 { ... close stuff ... }  
4  
5 % is defined internally as  
6 \def\foo#1{...}  
7 \def\endfoo{...}  
8 % the end part has been defined by no arguments  
9 \newenvironment{foo}[1]%  
10 {\def\fooNoI{#1} some code #1}  
11 {some code \fooNoI}
```

通过使用 xparse 宏包, 解决了这个问题.

重定义了 `\ref {arg}` 命令.

参考文献

- [1] Matthew D Zeiler and Rob Fergus. Visualizing and Understanding Convolutional Networks. 2013.

致谢

附录 A 测试附录

这里是附录, 看看是否会显示正常.