

---

# 软件工程作业管理系统 概要设计

---

	人员	日期
拟制	张三 李四 王五	yyyy-mm-dd
评审人	•	yyyy-mm-dd
批准	•	yyyy-mm-dd
签发	•	yyyy-mm-dd

## 摘 要

本文是软件工程需求规格说明书模板，修改自于中国科学技术大学本硕博毕业论文 L<sup>A</sup>T<sub>E</sub>X 模板示例文件，该模板由 zepinglee 和 seisman 创建，遵循中国科学技术大学的论文写作规范，适用于撰写学士、硕士和博士学位论文。

本文档最后一章演示如何使用 L<sup>A</sup>T<sub>E</sub>X 的一些基本命令以及本模板提供的一些特殊功能，模板的选项及详细用法请参考模板说明文档 `ustcthesis.pdf`。请在提交之前把最后一掌实例注释掉。

**关键词：**软件工程    中国科学技术大学    学位论文    L<sup>A</sup>T<sub>E</sub>X 通用模板    学士  
硕士    博士    示例文档    模板说明文档

表 1    缩略词清单

缩略语	英文全名	中文解释
c	d	e

## 目 录

摘要 .....	
第 1 章 引言 .....	6
1.1 编写目的 .....	6
1.2 项目背景 .....	6
1.3 术语 .....	6
第 2 章 任务概述 .....	7
2.1 目标 .....	7
2.2 开发与运行环境 .....	7
2.2.1 开发环境的配置 .....	7
2.2.2 测试环境的配置 .....	7
2.2.3 运行环境的配置 .....	7
2.3 需求概述 .....	7
2.4 条件与限制 .....	9
第 3 章 总体设计 .....	10
3.1 软件描述 .....	10
3.2 处理流程 .....	10
3.2.1 总体流程 .....	10
3.2.2 系统基本流程 .....	10
3.2.3 客户端基本流程 .....	10
3.2.4 服务器端基本流程 .....	10
3.2.5 功能 1 具体流程 .....	10
3.2.6 功能 2 具体流程 .....	11
3.2.7 功能 3 具体流程 .....	11
3.3 功能结构设计 .....	11
3.3.1 整体结构 .....	11
3.3.2 用户端结构 .....	11
3.3.3 服务器端结构 .....	11
3.3.4 后台数据库维护模块结构 .....	11

3.4 功能需求与程序代码的关系 .....	11
第 4 章 接口设计 .....	13
4.1 外部接口 .....	13
4.1.1 支付宝接口 .....	13
4.2 内部接口 .....	13
第 5 章 数据结构设计 .....	14
5.1 逻辑结构设计 .....	14
5.1.1 服务器端数据结构 .....	14
5.1.2 客户端数据结构 .....	19
5.1.3 用户端数据结构 .....	21
5.2 物理结构设计 .....	21
5.3 数据结构与程序模块的关系 .....	22
第 6 章 数据库设计 .....	23
6.1 数据库环境说明 .....	23
6.2 数据库的命名规则 .....	23
6.3 逻辑设计 .....	23
6.4 物理设计 .....	23
6.4.1 数据库产品 .....	23
6.4.2 实体属性、类型、精度 .....	23
6.5 安全性设计 .....	23
6.6 数据库管理与维护说明 .....	24
第 7 章 界面设计 .....	25
7.1 客户端界面 .....	25
7.2 服务器端界面 .....	25
7.3 登录界面 .....	25
7.4 xxx 功能界面 .....	25
第 8 章 出错处理设计 .....	26
8.1 数据库出错处理 .....	26
8.2 某模块失效处理 .....	26
第 9 章 安全保密设计 .....	27

第 10 章 维护设计 .....	28
第 11 章 图片 .....	29
11.1 示例 .....	29
11.2 带图注的图 .....	29
第 12 章 表格 .....	31
12.1 A Simple Table .....	31
12.2 长表格 .....	31
第 13 章 算法环境 .....	33
第 14 章 代码环境 .....	35
第 15 章 引用文献标注 .....	36
15.1 著者-出版年制标注法 .....	36
15.2 顺序编码制标注法 .....	36
15.3 其他形式的标注 .....	37
参考文献 .....	38

## 图目录

11.1 测试图片 .....	29
11.2 带图注的图片 .....	30

## 表目录

1 缩略词清单 .....	
1.1 术语表 .....	6
2.1 开发环境的配置 .....	7
2.2 测试环境的配置 .....	8
2.3 运行环境的配置 .....	8
3.1 功能需求与程序代码的关系表 .....	12
5.1 数据结构与程序代码的关系表 .....	22
6.1 用户数据表 Users 设计 .....	24
6.2 订单数据表 Orders 设计 .....	24
12.1 这里是表的标题 .....	31
12.2 长表格演示 .....	31

## 第 1 章 引言

### 1.1 编写目的

在本项目的前一阶段，也就是需求分析阶段，已经将系统用户对本系统的需求做了详细的阐述，这些用户需求已经在上一阶段中对不同用户所提出的不同功能，实现的各种效果做了调研工作，并在需求规格说明书中得到详尽得叙述及阐明。

本阶段已在系统的需求分析的基础上，对即时聊天工具做概要设计。主要解决了实现该系统需求的程序模块设计问题。包括如何把该系统划分成若干个模块、决定各个模块之间的接口、模块之间传递的信息，以及数据结构、模块结构的设计等。在以下的概要设计报告中将对在本阶段中对系统所做的所有概要设计进行详细的说明，在设计过程中起到了提纲挈领的作用。

在下一阶段的详细设计中，程序设计员可参考此概要设计报告，在概要设计即时聊天工具所做的模块结构设计的基础上，对系统进行详细设计。在以后的软件测试以及软件维护阶段也可参考此说明书，以便于了解在概要设计过程中所完成的各模块设计结构，或在修改时找出在本阶段设计的不足或错误。

### 1.2 项目背景

随着 xxx 的不断发展...

### 1.3 术语

[列出本文档中所用到的专门术语的定义和外文缩写的原词组]

表 1.1 术语表

缩写、术语	解释
c	d



## 第 2 章 任务概述

本系统的目标是实现一个 xxx 系统，包括客户端、服务器端两个部分。  
客户端面向 xxx 用户，为用户提供 xx 和 xx 服务。

### 2.1 目标

实现 xxx 系统，实现需求规格说明书中所描述的 xx 功能、xxx 功能和 xxx 功能，并且保证系统的健壮性和数据安全。

### 2.2 开发与运行环境

#### 2.2.1 开发环境的配置

表 2.1 开发环境的配置

类别	标准配置	最低配置
计算机硬件	基于 x86 结构的 CPU 主频 $\geq 2.4\text{GHz}$ 内存 $\geq 8\text{G}$ 硬盘 $\geq 200\text{G}$	基于 x86 结构的 CPU 主频 $\geq 1.6\text{GHz}$ 内存 $\geq 512\text{M}$ 硬盘 $\geq 2\text{G}$
计算机软件	Linux (kernel version $\geq 4.10$ ) GNU gcc (version $\geq 6.3.1$ )	Linux (kernel version $\geq 3.10$ ) GNU gcc (version $\geq 5.4$ )
网络通信	至少要有一块可用网卡 能运行 IP 协议栈即可	至少要有一块可用网卡 能运行 IP 协议栈即可
其他	采用 MySQL 数据库	采用 MySQL 数据库

#### 2.2.2 测试环境的配置

#### 2.2.3 运行环境的配置

### 2.3 需求概述

功能需求包括：

表 2.2 测试环境的配置

类别	标准配置	最低配置
计算机硬件	基于 x86 结构的 CPU 主频 $\geq 2.4\text{GHz}$ 内存 $\geq 8\text{G}$ 硬盘 $\geq 200\text{G}$	基于 x86 结构的 CPU 主频 $\geq 1.6\text{GHz}$ 内存 $\geq 512\text{M}$ 硬盘 $\geq 2\text{G}$
计算机软件	Linux (kernel version $\geq 4.10$ ) GNU gcc (version $\geq 6.3.1$ )	Linux (kernel version $\geq 3.10$ ) GNU gcc (version $\geq 5.4$ )
网络通信	至少要有一块可用网卡 能运行 IP 协议栈即可	至少要有一块可用网卡 能运行 IP 协议栈即可
其他	采用 MySQL 数据库	采用 MySQL 数据库

表 2.3 运行环境的配置

类别	标准配置	最低配置
计算机硬件	基于 x86 结构的 CPU 主频 $\geq 2.4\text{GHz}$ 内存 $\geq 8\text{G}$ 硬盘 $\geq 200\text{G}$	基于 x86 结构的 CPU 主频 $\geq 1.6\text{GHz}$ 内存 $\geq 512\text{M}$ 硬盘 $\geq 2\text{G}$
计算机软件	Linux (kernel version $\geq 4.10$ ) GNU gcc (version $\geq 6.3.1$ )	Linux (kernel version $\geq 3.10$ ) GNU gcc (version $\geq 5.4$ )
网络通信	至少要有一块可用网卡 能运行 IP 协议栈即可	至少要有一块可用网卡 能运行 IP 协议栈即可
其他	采用 MySQL 数据库	采用 MySQL 数据库

## 2.4 条件与限制

本节至少要与需求说明文档中相关章节相一致。

## 第 3 章 总体设计

### 3.1 软件描述

系统包括前台和后台两个部分。

前台主要功能是：

后台主要功能是：

### 3.2 处理流程

#### 3.2.1 总体流程

此处应当有一个图和对应的描述。

#### 3.2.2 系统基本流程

此处应当有一个图和对应的描述。

#### 3.2.3 客户端基本流程

这只是举个例子，如果没有客户端则不需要此节。

#### 3.2.4 服务器端基本流程

这只是举个例子，如果没有服务器端则不需要此节。

#### 3.2.5 功能 1 具体流程

举个例子：交易处理流程

已登录用户在购物车中提交请求交易的 POST 请求,提交的表单中指明了交易中包括的所有商品、商家、付款信息、收货地址,输入输出处理系统接收到合法请求后,向商品信息系统请求数据,收到数据以后验证是否正确,然后向订单系统发起生成新订单的请求,订单系统负责更新商品信息系统、商家信息,通知商家接单,返回订单处理结果输入输出处理系统,输入输出处理系统依照结果产生 HTML 页面,并返回给用户。

### 3.2.6 功能 2 具体流程

此处应当有描述。

### 3.2.7 功能 3 具体流程

此处应当有一个描述。

## 3.3 功能结构设计

### 3.3.1 整体结构

此处应当有一个图和对应的描述。系统如果像微内核那样，划分成核心模块和若干个子系统，此处应当有图示及说明，然后后续几个节应当描述这几个子系统。如果系统像宏内核，那应当说明有哪些紧密联系的模块，并在后续几个节内描述这些模块。

### 3.3.2 用户端结构

此处应当有一个图和对应的描述。这只是举个例子。可能的内容包括用户端的具体模块、耦合情况等。

### 3.3.3 服务器端结构

此处应当有一个图和对应的描述。这只是举个例子。

### 3.3.4 后台数据库维护模块结构

此处应当有一个图和对应的描述。这只是举个例子。

## 3.4 功能需求与程序代码的关系

[此处指的是不同的需求分配到哪些模块去实现。可按不同的端拆分此表]

表 3.1 功能需求与程序代码的关系表

·	模块 1	模块 2	模块 3
需求 1	·	Y	·
需求 2	·	Y	·
需求 3	·	Y	·
需求 4	Y	·	·
需求 5	·	·	Y

注：各项功能需求的实现与各个程序模块的分配关系

## **第 4 章 接口设计**

### **4.1 外部接口**

#### **4.1.1 支付宝接口**

详细讲述不同的接口（查询状态、支付交易、获取回执等）

### **4.2 内部接口**

内部模块/系统之间的交互的接口。

## 第 5 章 数据结构设计

### 5.1 逻辑结构设计

#### 5.1.1 服务器端数据结构

##### 5.1.1.1 客户端请求

请求以 socket 的形式以字节的格式传输到服务器端口。

代码 5.1 客户端请求

```
1 struct command{
2     Byte b;
3 }
```

##### 5.1.1.2 登录请求

解析结果是登录请求，请求字段分别为用户 id 和密码，两个都加密后进行传输。

代码 5.2 客户端请求

```
1 struct LoginCommand{
2     commandID id;
3     String userid;
4     String password;
5 }
```

##### 5.1.1.3 作业/实验/通知查询请求

解析结果为作业实验查询请求，请求字段为用户 id 和请求查询的作业/实验名称。

代码 5.3 作业/实验/通知查询请求

```
1 struct WorkQueryCommand{
2     commandID id;
3     String userid;
4     String workid;
5 }
```



#### 5.1.1.4 作业/实验/通知发布请求

向服务器发布作业/实验/通知。解析为发布请求，请求字段为用户名，之后为字节流，内容是作业/实验/通知的详细信息，具体解析工作交由作业/实验/通知功能结构完成。

代码 5.4 作业/实验/通知发布请求

```
1 struct WorkPublishCommand{
2     commandID id;
3     String userid;
4     Byte s;
5 }
```

#### 5.1.1.5 作业/实验/通知修改请求

修改已经发布的作业/实验/通知。解析为修改请求，请求字段为用户名以及需要修改的工作名，之后为字节流，内容为需要进行修改的详细内容，具体解析工作交由作业/实验/通知功能结构完成。

代码 5.5 作业/实验/通知修改请求

```
1 struct WorkAdjustCommand{
2     commandID id;
3     String userid;
4     String workid;
5     Byte s;
6 }
```

#### 5.1.1.6 作业/实验提交请求

提交所完成的作业/实验。解析为提交请求，请求字段为用户名以及待提交的作业，之后为字节流，内容为需要提交的具体成果，具体解析工作交由作业/实验/通知功能结构完成。

代码 5.6 作业/实验提交请求

```
1 struct WorkCommitCommand{
2     commandID id;
3     String userid;
4     String workid;
5     Byte s;
6 }
```

#### 5.1.1.7 作业/实验查看请求

查看所完成的作业/实验。解析为提交请求，请求字段为用户名以及待查看的作业。

代码 5.7 作业/实验查看请求

```
1 struct WorkCheckCommand{
2     commandID id;
3     String userid;
4     String workid;
5 }
```

#### 5.1.1.8 成绩录入请求

解析结果是录入成绩请求，请求字段是用户 id，作业 id，之后是字节流，内容为具体学生和成绩的信息，具体解析交由成绩录入功能模块处理。

代码 5.8 成绩录入请求

```
1 struct ScoreCommitCommand{
2     commandID id;
3     String userid;
4     String workid;
5     Byte s;
6 }
```

#### 5.1.1.9 成绩查询请求

解析结果是录入成绩请求，请求字段是用户 id，待查询用户 id，作业 id。

代码 5.9 成绩查询请求

```
1 struct ScoreCommitCommand{
2     commandID id;
3     String userid;
4     String queryid;
5     String workid;
6 }
```

#### 5.1.1.10 成绩统计调整请求

解析结果是成绩统计调整请求，请求字段是用户 id，作业 id，后面跟有字节流，内容为具体的更改或者统计信息。

代码 5.10 成绩统计调整请求

```

1  struct ScoreAdjustCommand{
2      commandID id;
3      String userid;
4      String workid;
5      Byte s;
6  }
    
```

#### 5.1.1.11 资源上传请求

解析结果是资源上传请求，请求字段是用户 id，后面跟有字节流，内容为上传的目的地址，上传文件的相关信息和具体资源数据。具体解析交由资源上传模块解析。

代码 5.11 资源上传请求

```

1  struct ResoureUploadCommand{
2      commandID id;
3      String userid;
4      Byte s;
5  }
    
```

#### 5.1.1.12 资源下载请求

解析结果是资源下载请求，请求字段是用户 id 和资源 id。

代码 5.12 资源下载请求

```

1  struct ResourceDownloadCommand{
2      commandID id;
3      String userid;
4      String resourceid;
5  }
    
```

#### 5.1.1.13 日程同步请求

解析结果是日程同步请求，请求字段是用户 id，后面跟有字节流，内容是本地等待同步的日程信息。具体解析交由日程同步模块处理。

代码 5.13 日程同步请求

```

1  struct ScheduleSynCommand{
2      commandID id;
3      String userid;
4      Byte s;
5  }
    
```

#### 5.1.1.14 讨论区新建/回复请求

解析结果是讨论区相关请求，请求字段是用户 id，后面跟有字节流，内容为具体的主题/回复信息。具体解析交由讨论区模块处理。

代码 5.14 讨论区新建/回复请求

```
1  struct NewDisCommand{
2      commandID id;
3      String userid;
4      Byte s;
5  }
```

#### 5.1.1.15 讨论区查询请求

解析结果是讨论区查询请求，请求字段是用户 id 和需要查询的主题。

代码 5.15 讨论区查询请求

```
1  struct DisQueryCommand{
2      commandID id;
3      String userid;
4      String thesis
5  }
```

#### 5.1.1.16 新建博文请求

解析结果是新建博文请求，请求字段是用户 id，后面跟有字节流，内容为具体博文内容。具体解析交由博文模块处理。

代码 5.16 新建博文请求

```
1  struct NewBlogCommand{
2      commandID id;
3      String userid;
4      Byte s;
5  }
```

#### 5.1.1.17 学习笔记同步请求

解析结果是学习笔记同步请求，请求字段是用户 id，后面跟有字节流，内容为本地待提交的笔记内容，具体交由学习笔记同步模块解析。

代码 5.17 学习笔记同步请求

```
1  struct NoteSynCommand{
```

```

2      commandID id;
3      String userid;
4      Byte s;
5      }

```

## 5.1.2 客户端数据结构

### 5.1.2.1 作业/实验信息

代码 5.18 作业/实验信息

```

1      struct WorkInfo{
2          String caption;
3          String content;
4          Data deadline;
5          String[] userid;
6          String publisher;
7      }

```

作业实验的信息包括标题、详细内容、提交截止时间、需要提交作业的用户以及发布者。

### 5.1.2.2 通知信息

代码 5.19 通知信息

```

1      struct NoticeInfo{
2          String caption;
3          String content;
4          String[] userid;
5          String publisher;
6      }

```

通知信息包括标题，详细内容，需要接受消息的用户以及消息的发布者。

### 5.1.2.3 个人成绩

代码 5.20 个人成绩

```

1      struct ScoreEntry{
2          int score;
3          String userid
4      }

```

每条成绩记录包括用户和其相应的成绩。

#### 5.1.2.4 成绩表

代码 5.21 成绩表

```
1  struct ScoreList{
2      String Test;
3      ScoreEntry[] score;
4  }
```

成绩记录中包括考试/作业名称和各个用户的成绩。

#### 5.1.2.5 资源信息

代码 5.22 资源信息

```
1  struct ResourceInfo{
2      String name;
3      int type;
4      int size;
5      Byte content;
6  }
```

资源信息包括资源名称，资源的类型，资源大小，资源内容等等。

#### 5.1.2.6 日程信息

代码 5.23 日程信息

```
1  struct ScheduleInfo{
2      Data data;
3      int duration;
4      String name;
5      String place;
6      String description;
7  }
```

每条日程包括日程的时间地点，事件的持续时间，事件的名称和详细描述。

#### 5.1.2.7 讨论区主题

代码 5.24 讨论区主题

```
1  struct DisThesis{
2      String title;
3      String content;
4      Data creationtime;
5      Vector<ReplyInfo> reply;
6      String author;
```

```
7 | }
```

每个讨论区主题包含创建时间、标题、内容、作者以及回复信息。

#### 5.1.2.8 讨论区回复

代码 5.25 讨论区回复

```
1 struct ReplyInfo{
2     String content;
3     String author;
4     Data time;
5 }
```

每条回复包括回复内容、回复者和时间信息。

#### 5.1.2.9 博文信息

代码 5.26 博文信息

```
1 struct BlogInfo{
2     String title;
3     String author;
4     Data time;
5     int type;
6 }
```

每条博文记录包括具体内容、作者、时间以及是否公开。

#### 5.1.2.10 学习笔记

代码 5.27 学习笔记

```
1 struct NoteData{
2     String name;
3     Data time;
4     String author;
5     String content;
6 }
```

每个学习笔记包括标题、内容、创建时间和作者信息。

### 5.1.3 用户端数据结构

## 5.2 物理结构设计

各个数据结构无特殊物理结构要求。

### 5.3 数据结构与程序模块的关系

[此处指的是不同的数据结构分配到哪些模块去实现。可按不同的端拆分此表]

表 5.1 数据结构与程序代码的关系表

·	模块 1	模块 2	模块 3
结构 1	·	Y	·
结构 2	·	Y	·
结构 3	·	Y	·
结构 4	Y	·	·
结构 5	·	·	Y

注：各项数据结构的实现与各个程序模块的分配关系



## 第 6 章 数据库设计

### 6.1 数据库环境说明

本系统的数据系统采用 MySQL/PostgreSQL/Microsoft SQL Server 数据库系统。

其中 xxx 模块因为 xxx 而需要用到 Hadoop 架构。

### 6.2 数据库的命名规则

是否允许单词缩写，允许的单词缩写有哪些。

表名是单数还是复数。关联表如何命名。字符数限制等。

字段是否带上前缀（如 integer 类型则加上 i 前缀等）。

### 6.3 逻辑设计

是否需要满足某一种范式。

画个实体的逻辑关系表/图在此处。

### 6.4 物理设计

#### 6.4.1 数据库产品

用哪家数据库，是否分布式等。

#### 6.4.2 实体属性、类型、精度

##### 6.4.2.1 客户数据表设计

##### 6.4.2.2 订单数据表设计

### 6.5 安全性设计

备份和容灾设计。

表 6.1 用户数据表 Users 设计

字段名	类型	大小	说明	备注
ID	char	64	用户的唯一标识符	主键
pw	char	512	用户的登录密码	.

注：用户数据表 Users 设计

表 6.2 订单数据表 Orders 设计

字段名	类型	大小	说明	备注
ID	char	64	订单的唯一标识符	主键
user	char	64	对应用户	外键，来自 xx 表

注：订单数据表 Orders 设计

## 6.6 数据库管理与维护说明

对于数据库的维护，随时对数据库中的信息加以调试和保存备份。同样需要个工作人员进行系统的分析和用户的反馈，对系统进行升级以及功能的完善。同时保证系统安全有序的运行。

## 第 7 章 界面设计

### 7.1 客户端界面

此处应当有一个简略的图，重点是展示你与用户交互的逻辑。（processon 上画一个不花时间）

### 7.2 服务器端界面

此处应当有一个简略的图。

### 7.3 登录界面

此处应当有一个简略的图。

### 7.4 xxx 功能界面

此处应当有一个简略的图。

## 第 8 章 出错处理设计

### 8.1 数据库出错处理

多重备份时，应采取何种策略，先利用哪一份备份；系统是否暂停服务等。

### 8.2 某模块失效处理

是否整个系统暂停服务，还是维持最小服务状态、如何尽快恢复服务还是删库跑路等。

## 第 9 章 安全保密设计

可能的内容包括保密性、是否采取加密传输、密钥如何分发和管理等。

## 第 10 章 维护设计

可能的内容包括数据库的日常备份、压缩、维护等。

## 第 11 章 图片

本章展示图片相关用法。

### 11.1 示例



图 11.1 测试图片

### 11.2 带图注的图



图 11.2 带图注的图片

注：the solid lines represent the time histogram of the spontaneous activities of an old monkey cell(gray) and a young monkey cell (black). The bin-width is 1



## 第 12 章 表格

### 12.1 A Simple Table

表 12.1 这里是表的标题

a	b
c	d

注：这里是表的注释

### 12.2 长表格

表 12.2 长表格演示

名称	说明	备注
AAAAAAAAAAAAA	BBBBBBBBBBBBB	CCCCCCCCCCCCCCC
AAAAAAAAAAAAA	BBBBBBBBBBBBB	CCCCCCCCCCCCCCC
AAAAAAAAAAAAA	BBBBBBBBBBBBB	CCCCCCCCCCCCCCC
AAAAAAAAAAAAA	BBBBBBBBBBBBB	CCCCCCCCCCCCCCC
AAAAAAAAAAAAA	BBBBBBBBBBBBB	CCCCCCCCCCCCCCC
AAAAAAAAAAAAA	BBBBBBBBBBBBB	CCCCCCCCCCCCCCC
AAAAAAAAAAAAA	BBBBBBBBBBBBB	CCCCCCCCCCCCCCC
AAAAAAAAAAAAA	BBBBBBBBBBBBB	CCCCCCCCCCCCCCC
AAAAAAAAAAAAA	BBBBBBBBBBBBB	CCCCCCCCCCCCCCC
AAAAAAAAAAAAA	BBBBBBBBBBBBB	CCCCCCCCCCCCCCC
AAAAAAAAAAAAA	BBBBBBBBBBBBB	CCCCCCCCCCCCCCC
AAAAAAAAAAAAA	BBBBBBBBBBBBB	CCCCCCCCCCCCCCC
AAAAAAAAAAAAA	BBBBBBBBBBBBB	CCCCCCCCCCCCCCC
AAAAAAAAAAAAA	BBBBBBBBBBBBB	CCCCCCCCCCCCCCC
AAAAAAAAAAAAA	BBBBBBBBBBBBB	CCCCCCCCCCCCCCC

续下页

表 12.2 长表格演示 (续)

[illegible]

## 第 13 章 算法环境

模板中使用 `algorithm2e` 宏包实现算法环境。关于该宏包的具体用法，请阅读宏包的官方文档。

```

Data: this text

Result: how to write algorithm with LATEX2ε

1 initialization;
2 while not at end of this document do
3   read current;
4   if understand then
5     go to next section;
6     current section becomes this one;
7   else
8     go back to the beginning of current section;
9   end
10 end
    
```

算法 13.1: 算法示例 1

```

input : A bitmap  $Im$  of size  $w \times l$ 
output: A partition of the bitmap

1 special treatment of the first line;
2 for  $i \leftarrow 2$  to  $l$  do
3   special treatment of the first element of line  $i$ ;
4   for  $j \leftarrow 2$  to  $w$  do
5      $left \leftarrow \text{FindCompress}(Im[i, j - 1]);$ 
6      $up \leftarrow \text{FindCompress}(Im[i - 1, j]);$ 
7      $this \leftarrow \text{FindCompress}(Im[i, j]);$ 
8     if left compatible with this then //  $O(left, this) == 1$ 
9       if  $left < this$  then  $\text{Union}(left, this);$ 
10      else  $\text{Union}(this, left);$ 
11    end
12    if up compatible with this then //  $O(up, this) == 1$ 
13      if  $up < this$  then  $\text{Union}(up, this);$ 
14      // this is put under up to keep tree as
15      flat as possible
16      else  $\text{Union}(this, up);$ 
17      // this linked to up
18    end
19  end
20 foreach element  $e$  of the line  $i$  do  $\text{FindCompress}(p);$ 
21 end

```

算法 13.2: 算法示例 2

## 第 14 章 代码环境

模板中使用 listings 宏包实现代码环境。详细用法见宏包的官方说明文档。

以下是代码示例，可以在文中任意位置引用??。

代码 14.1 示例代码

```
1 #include <stdio.h>
2
3 int main( )
4 {
5     printf("hello, \world\n");
6     return 0;
7 }
```

## 第 15 章 引用文献标注

### 15.1 著者-出版年制标注法

<code>\citestyle{ustcauthoryear}</code>	
<code>\cite{knuth86a}</code>	$\Rightarrow$ Knuth (1986)
<code>\citet{knuth86a}</code>	$\Rightarrow$ Knuth (1986)
<code>\citet[chap.~2]{knuth86a}</code>	$\Rightarrow$ Knuth (1986, chap. 2)
<code>\citep{knuth86a}</code>	$\Rightarrow$ (Knuth, 1986)
<code>\citep[chap.~2]{knuth86a}</code>	$\Rightarrow$ (Knuth, 1986, chap. 2)
<code>\citep[see][ ]{knuth86a}</code>	$\Rightarrow$ (see Knuth, 1986)
<code>\citep[see][chap.~2]{knuth86a}</code>	$\Rightarrow$ (see Knuth, 1986, chap. 2)
<code>\citet*{knuth86a}</code>	$\Rightarrow$ Knuth (1986)
<code>\citep*{knuth86a}</code>	$\Rightarrow$ (Knuth, 1986)
<code>\citet{knuth86a,tlc2}</code>	$\Rightarrow$ Knuth (1986); Mittelbach et al. (2004)
<code>\citep{knuth86a,tlc2}</code>	$\Rightarrow$ (Knuth, 1986; Mittelbach et al., 2004)
<code>\cite{knuth86a, knuth84}</code>	$\Rightarrow$ Knuth (1984, 1986)
<code>\citet{knuth86a, knuth84}</code>	$\Rightarrow$ Knuth (1984, 1986)
<code>\citep{knuth86a, knuth84}</code>	$\Rightarrow$ (Knuth, 1984, 1986)

### 15.2 顺序编码制标注法

`\citestyle{ustcnumerical}`

<code>\cite{knuth86a}</code>	$\Rightarrow$	[2]
<code>\citet{knuth86a}</code>	$\Rightarrow$	Knuth <sup>[2]</sup>
<code>\citet[chap.~2]{knuth86a}</code>	$\Rightarrow$	Knuth <sup>[2]</sup> , chap. 2 <sup>1</sup>
<code>\citep{knuth86a}</code>	$\Rightarrow$	[2]
<code>\citep[chap.~2]{knuth86a}</code>	$\Rightarrow$	[2] chap. 2
<code>\citep[see][ ]{knuth86a}</code>	$\Rightarrow$	see <sup>[2]</sup>
<code>\citep[see][chap.~2]{knuth86a}</code>	$\Rightarrow$	see <sup>[2]</sup> chap. 2
<code>\citet*{knuth86a}</code>	$\Rightarrow$	Knuth <sup>[2]</sup>
<code>\citep*{knuth86a}</code>	$\Rightarrow$	[2]
<code>\citet{knuth86a,tlc2}</code>	$\Rightarrow$	Knuth <sup>[2]</sup> , Mittelbach et al. <sup>[3]</sup>
<code>\citep{knuth86a,tlc2}</code>	$\Rightarrow$	[2,3]
<code>\cite{knuth86a, knuth84}</code>	$\Rightarrow$	[1,2]
<code>\citet{knuth86a, knuth84}</code>	$\Rightarrow$	Knuth <sup>[1, 2]</sup>
<code>\citep{knuth86a, knuth84}</code>	$\Rightarrow$	[1,2]
<code>\cite{knuth86a, knuth84,tlc2}</code>	$\Rightarrow$	[1–3]

### 15.3 其他形式的标注

<code>\citealt{tlc2}</code>	$\Rightarrow$	Mittelbach et al. <sup>3</sup>
<code>\citealt*{tlc2}</code>	$\Rightarrow$	Mittelbach, Goossens, Braams, and Carlisle <sup>3</sup>
<code>\citealp{tlc2}</code>	$\Rightarrow$	<sup>3</sup>
<code>\citealp*{tlc2}</code>	$\Rightarrow$	<sup>3</sup>
<code>\citealp{tlc2, knuth86a}</code>	$\Rightarrow$	<sup>2,3</sup>
<code>\citealp[pg.~32]{tlc2}</code>	$\Rightarrow$	<sup>3</sup> pg. 32
<code>\citenum{tlc2}</code>	$\Rightarrow$	3
<code>\citetext{priv.\ comm.}</code>	$\Rightarrow$	[priv. comm.]
<code>\citeauthor{tlc2}</code>	$\Rightarrow$	Mittelbach et al.
<code>\citeauthor*{tlc2}</code>	$\Rightarrow$	Mittelbach, Goossens, Braams, and Carlisle
<code>\citeyear{tlc2}</code>	$\Rightarrow$	2004
<code>\citeyearpar{tlc2}</code>	$\Rightarrow$	2004

## 参考文献

- Knuth D E. May 1984. Literate programming[J]. *The Computer Journal*. 27(2):97–111.
- Knuth D E. 1986. Computers and Typesetting: A The  $\text{\TeX}$ book[M]. Reading, MA, USA: Addison-Wesley.
- Mittelbach F, Goossens M, Braams J, et al. 2004. The  $\text{\LaTeX}$  Companion[M]. 2nd ed. Reading, MA, USA: Addison-Wesley.