实验一 面向过程的分析与设计

1. 实验目的

掌握软件需求结构化分析方法；掌握软件体系结构模型及面向数据流的设计方法；掌握使用过程设计工具描述模块的详细数据结构和算法；完成系统面向过程的分析与设计文档。

1. 实验要求

1. 描述系统背景；

2. 进行功能分析；

3. 画出数据流图及建立数据字典；

4. 软件体系结构设计描述；

5. 用面向数据流的方法设计系统软件结构；

6. 设计数据库结构；

7. 描述总体设计中各个模块的详细数据结构和算法

8. 攥写面向过程的分析与设计文档。

1. 实验原理
2. 需求分析的基本要点

传统软件工程使用结构化方法完成分析用户需求的工作。需求分析是发现、求精、建模、说明规格和复审的过程。需求分析的第一步是进一步了解用户当前所处的情况，发现用户所面临的问题和对目标系统的基本需求；接下来应该与用户深入交流，对用户的基本需求反复细化逐步求精，以得出对目标系统的完整、准确和具体的需求。具体地说，应该确定系统必须具有的功能、性能、可靠性和可用性，必须实现的出错处理需求、接口需求和逆向需求，必须满足的约束条件，并且预测系统的发展前景。

为了详细了解并正确理解用户的需求，必须使用适当的方法与用户沟通。访谈时用户沟通的历史悠久的技术，至今仍被许多系统分析员采用。从可行性研究阶段得到的数据流图出发。为了促使用户与分析员其心协力共同分析需求，人们研究出一种面向团队的需求收集法，称为简易的应用规格说明技术，现在这种技术已经称为信息系统领域使用的主流技术。实践表明，快熟加你软件原型是最准确、最有效和最强大的需求分析技术。

快速原型应该具备的基本特征是“快速”和“容易修改”，因此，必须用适当的软件工具支持快速原型技术。通常使用第四代技术、可重用的软件构件及形式化规格说明与原型环境，快速地构建修改原型。

为了更好地理解问题，人们常常采用建立模型的方法。结构化分析实质上就是一种建模活动。在需求分析阶段通常建立数据模型、功能模型和行为模型。

除了创建分析模型之外，在需求分析阶段还应该写出软件需求说明书，经过严格评审并得到用户确认之后，作为这个阶段的最终结果。通常主要从一致性、完整性、现实性和有效项等4个方面复审软件需求说明书。

人们习惯使用E-R图建立数据模型、使用数据流图建立功能模型、使用状态图建立行为模型。我们应该掌握这些图形的基本符号，并能正确地使用这些符号建立软件系统的模型。

数据字典描述在数据模型、功能模型和行为模型中出现的数据对象及控制信息的特征，给出它们的准确定义。因此，数据字典称为把3中分析模型粘合在一起的“粘合剂”，是分析模型的“核心”。为了提高可理解性，还可以用层次方框图或Warnier图等图形工具辅助描绘系统的数据结构。为了减少冗余，简化修改步骤，往往需要规范数据的数据结构。

算法也是重要的，分析的基本目的是去顶系统必须做什么，算法定义了转变的规则。没有对算法的了解就不能确切知道系统的功能。IPO图是描述算法的有效工具。

1. 分析建模

为了更好地理解复杂事物，人们常常采用建立事物模型的方法。所谓模型。就是为了理解事物而对事物做出的一种抽象，根据模型的形态可分为物理模型和逻辑模型。逻辑模型是对事物的一种无歧义的书面描述。通常，逻辑模型由一组图形符号和组织这些符号的规则组成。

尽管目前有许多不同的用于需求分析的结构化分析方法，但是，多数方法都遵守下属准则：

* 必须理解并描述文图的信息域，根据这条准则因该建立数据模型；
* 必须定义软件应完成的功能，这条准则要求建立功能模型；
* 必须描述作为外部是结果的软件行为，这条准则要求建立行为模型；
* 必须对描述目标系统信息、功能和行为的模型进行分解，用层次的方式展开细节。

（3） 软件总体设计的基本要点

总体设计也称为总体设计或初步设计，基本目的时用比较抽象概括的方式确定系统如何完成预定的任务，确定系统的物理配置方案，确定组成系统的每个程序的结构。总体设计主要有两个阶段组成，首先进行系统设计或方案设计，从数据流图出发设想完成系统功能的若干种合理的物理方案，仔细分析和比较这些方案，并且和用户共同选定一个最佳方案。然后进行软件的结构设计，确定软件有哪些模块组成及这些模块之间的动态调用关系。层次图和结构图是描绘软件结构的常用工具。

进行软件结构设计时应该遵循的最主要的原则是模块儿独立，也就是说，软件应该有一组完成相对独立的子功能的模块儿组成，这些模块儿彼此之间的接口关系应该尽量简单。

抽象和求精是一对互补的概念，也是人类解决复杂问题时最常用、最有效的方法。在进行软件结构设计时一种有效的方法就是由抽象到具体的构造出软件的层次结构。

自顶向下、逐步求精是进行软件结构设计的常用途径。但是，如果已经有了详细的数据流图，也可以使用面向数据流的设计方法，用形式化的方法由数据流图映射出软件结构。应该记住，这样映射出来的只是软件的初步结构。还必须根据设计原理并且参考启发式规则，认真分析和改进软件的初步结构，以得到质量更高的模块儿和更合理的软件结构。

再进行详细的过程设计和程序编写之前，首先进行结构设计，其好处是可以在软件开发的早期站在全局的高度对软件结构进行优化。在这个时期进行优化付出的代价不高，却可以使软件质量得到重大改进。

（4）系统分析与设计关系

系统分析的基本任务是定义用户所需要的软件任务，也就是回答系统必须“做什么”这个问题。系统设计的基本任务是设计实现目标系统的具体方案，也就是回答“怎样做”这个问题。虽然分析与设计的任务性质不同，但是两者之间却有非常密切的关系。

软件工程师必须依靠用户对软件的需求来设计软件，因此，结构化分析的结果是进行结构化设计的最基本，最重要的输入信息。

结构设计的任务是确定程序有哪些模块儿组成及这些模块相互之间的关系。在需求分析阶段画出的数据流图是进行结构设计的主要依据，为结构设计提供了最基本的输入信息。

数据设计把需求分析阶段创建的信息模型转变成实现软件所需要的数据结构。在实体—联系图中定义的数据和数据之间的关系。以及数据字典中给出的详细的数据定义，共同为数据设计活动奠定了坚实的基础。

接口涉及的结果描绘了软件内部、软件协作系统之间，以及软件与使用它的人之间的通信方式。接口意味着信息的流动（数据流和控制流），因此数据流提供了进行接口设计所需要的基本信息。

过程设计决定程序中包含的每个模块的实现算法。需求分析阶段画出的IPO图（表）为过程设计奠定了基础。

虽然结构化分析为结构化设计提供了最基本、最要的输入信息。但并不是说可以简单的把结构化分析的结果映射成结构化设计的结果。实际上，结构化实际过程综合了下述诸多元素：从以往开发类似软件的经验中获取得的直觉和判断力，指导软件模型演化的一组原理（也称为准则）和启发规则，评定软件质量的一组标准以及导出最终的设计结果的迭代过程。

软件工程师在软件设计过程中所做出的决策，将最终决定软件开发能够成功。更重要的是，这些设计决策将决定软件维护的难易程度。

软件设计之所以如此重要，是因为设计是软件开发过程中决定软件产品质量的关键阶段。设计为我们提供了可以进行质量评估的软件表示（软件模型），设计是把用户需求准确的转变成最终的软件产品的唯一方式。软件设计是后续的一切软件开发和维护步骤的基础，如果不进行设计。就会产生构造出不稳定的软件系统的风险，稍作改动，系统就可能崩溃，这样的系统很难维护、很难测试。直到软件工程过程的后期（例如，编码结束），才能评价系统的质量，但是，此时才发现软件的质量问题已经太晚了。

（5）面向数据流的设计方法

面向数据流的设计方法的目标是给出设计软件结构的一个系统化的途径，这种设计方法定义了一些映射规则，利用这些映射可以把数据流图变换成软件结构。

1） 数据流的类型

面向数据流的设计方法把信息流映射成软件结构，信息流的类型决定了映射的方法。信息流有下属两种类型。

a) 变换流

如果信息沿输入通路进入系统，同时有外部形式变成内部形式，进入系统的信息通过变换中心，经过精加工处理后再沿输出通路变换成外部形式离开软件系统。当信息流具有这些特征时，就叫做变换流。

b) 事物流

原则上所有信息流都可以归结为变换流。但是如果信息沿输入通路到达一个称为事务中心的处理T,这个处理根据输入数据的类型在若干个候选的动作序列中选取出一个来执行，则这类数据流应该划分为一类特殊的数据流，称为事务流。

2） 设计步骤

面向数据流的设计方法主要有下述七个设计步骤：

a) 第一步：复查基本系统模型；

b) 第二步：复查并精华数据流图；

c) 第三步：确定数据流图时具有变换特征还是事物特征；

d) 第四部：确定数据流边界；

e) 第五步：完成“第一级分解”；

f) 第六步：完成“第二级分解”；

g) 第七部：优化。

* 1. 软件详细设计的基本要点

详细设计阶段的关键任务是确定怎样具体地实现用户需要的软件系统，也就是要设计出程序的“蓝图”。除了保证软件可靠性性之外，使将来编写出的程序可读性好、易于理解、易于测试、易于修改和维护，是详细设计阶段重要的目标。结构化程序设计技术是实现上述目标的基本保证，是进行详细设计的逻辑基础。

人机界面设计是接口设计的一个重要组成部分。对于交互式系统来说，人机界面设计和数据设计结构设计及过程设计一样重要，人机界面的质量直接影响用户对软件产品的接受程度，因此对人机界面设计必须给予足够重视，在设计人机界面的过程中必须充分重视并认真处理好系统响应时间、用户帮助设施、出错详细处理和命令交互等四个问题。人机界面设计是一个迭代过程，通常，先创建设计模型，接下来用原型实现这个设计模型并有用户试用和评估原型，然后根据用户意见修改原型，直到用户满意为止。总结人们在设计人机界面过程中积累的经验，得出了一些关于用户界面设计的指南，认真遵循遵守这些指南有助于设计出有好的、高效的人机界面。

过程设计应该在数据设计、体系结构设计和接口设计完成之后进行，它的任务是设计解题的详细步骤（即算法），它是详细设计阶段应完成的主要工作。过程设计的主要工具可分为图形、表格和语言三类，这三类工具各有所长，我们应该根据需要选择适当的工具。

1）人机界面设计

人机界面的设计质量，直接影响用户对软件产品的评价，从而影响软件产品的竞争力和寿命，因此，必须对人机界面设计给予足够的重视。

由于对人机界面的评价，在很大程度上是由人的主观因素决定，因此，使用基于原型的系统化的设计策略，是成功的设计人机界面的关键。

用户界面设计是一个迭代的过程，也就是说，通常先创建设计模型，再用原型实现这个设计模型，并有用户使用和评估，然后根据用户意见进行修改，直到用户满意为止。

2)过程设计

过程设计应该在数据设计结构设计和接口设计完成之后进行，它是详细设计阶段应该完成的主要任务过程。

过程设计的任务还不是具体的编写程序，而是要设计出程序的蓝图，以后程序也将根据这个蓝图写出实际的程序代码，因此，过程设计的结果基本上决定了最终的程序代码的质量。考虑程序的代码质量时必须注意，程序的“读者”有两个，那就是计算机和人。在软件的生命周期中，设计测试方案、诊断程序错误、修改和改进程序等都必须首先读懂程序。实际上对于长期使用的软件系统而言，人读程序的时间可能比写程序的时间还要长得多，因此，衡量程序的质量不仅要看他的逻辑是否正确，性能是否满足要求，更重要的是要看它是否容易阅读和理解。过程设计的目标不仅仅是逻辑上正确的实现每个模块的功能，更重要的是设计出的处理过程应该尽可能简明易懂。结构程序设计技术是实现上述目标的关键技术，因此是过程设计的逻辑基础。

3)过程设计工具

描述程序处理过程的工具被称为过程设计工具，它们可以分为图形、表格和语言三类。无论是哪种工具，对它们的基本要求都是要提供对设计的无歧义的描述，也就是应该指明控制流程、处理功能、数据组织及其他方面的实现细节，从而在编码阶段能把对设计的描述直接翻译成程序代码。此外，这类工具应该尽可能直观、易学、易懂。

* 程序流程图

程序流程图又称为程序框图，它是历史最悠久使用最广泛的描述过程设计的方法，然而它也是用的最混乱的一种方法。

* 盒图（N—S图）

出于要有一种不允许违背结构程序设计精神的图形工具的考虑，提出了盒图又称为N—S图。

盒图没有箭头，因此不允许随意转移控制，坚持使用盒图作为详细设计的工具，可以使程序员逐步培养成结构化的方法思考问题和解决问题的习惯。

* PAD图

PAD图是问题分析图的英文缩写，它用二维树形结构的图来表示程序的控制流，将这种图翻译成程序代码比较容易。

* 判定表

当算法中包含多重嵌套的条件时，用程序流程图、盒图，PAD图过程设计语言PDL都不易清楚的描述，然而判定表却能够清晰地表示复杂的条件组合与应有应做的动作之间的对应关系。

一张判定表有四部分组成，左上列出所有条件，左下部是所有可能做的动作，右上部是表示各种条件组合的一个矩阵，右下部是和每一种条件组合相对应的动作。判定表右半部的每一列实质上是一条规则，规定了与特定的条件组合相对应的动作。

* 判定树

判定表虽然能清晰地表示复杂的条件组合与应做的动作之间的对应关系，但其含义却不是一眼就能看出来的，初次接触这种工具的人理解它需要有一个简短的学习过程。

判定树是判定表的变种，也能清晰地表示复杂的条件组合与应做的动作之间的对应关系。判定树的优点在于：它的形式简单到不需要任何说明，一眼就可以看出其含义，因此易于掌握和使用，多年来判定树一直受到人们的重视，是一种比较常用的系统分析和设计的工具。

* 过程设计语言

过程设计语言（PDL）也称为伪码，它是一个笼统的名称，现在有许多种不同的过程设计语言在使用，它是用正文形式来表示数据和处理过程的设计工具。

PDL具有严格的关键字外部语法，用于定义控制结构和数据结构；另一方面，PDL表示实际操作和条件的内部语法通常又是灵活自由的，可以适应各种工程项目的需要。因此，一般说来，PDL是一种”混杂”语言，它使用一种语言的词汇，同时却使用另一种语言(某种结构化的程序设计语言)的语法。

（7）面向过程的分析与设计文档

面向过程的分析与设计文档通常用自然语言完整、准确、具体地描述随目标系统的需求，这样的规格说明书具有容易书写、容易理解的优点。

1. 实验环境

安装了Windows系列操作系统、Office系列Word、Visio等软件的计算机。

1. 实验任务
2. 导出系统的详细的逻辑模型。通常用数据流图、实体—联系图、数据字典和主要的处理算法描述等来共同表示这个逻辑模型；使用面向数据流的方法设计软件系统的结构并使用过程设计工具描述概要设计中各个模块的详细数据结构和算法；
3. 攥写面向过程的分析与设计文档。
4. 实验内容与步骤
5. 问题背景及描述；
6. 功能分析；
7. 建立数据流程图及建立数据字典；
8. 设计软件整体结构；
9. 设计数据库；
10. 进行模块内部设计。

绘图工具网址：https://www.processon.com/

实验报告系统地址：210.42.225.33