

# 软件工程

实验指导与习题

南昌航空大学软件工程课程组



---

# 目录

## 第一篇 实验指导

实验 1 用PowerDesigner建模工具绘制数据流图 .....	1
1.1 实验环境.....	1
1.2 实验目的.....	1
1.3 实验内容.....	1
1.4 实验要求.....	1
1.5 实验步骤.....	2
实验 2 用PowerDesigner建模工具设计数据库 .....	7
2.1 实验环境.....	7
2.2 实验目的.....	7
2.3 实验内容.....	7
2.4 实验要求.....	7
2.5 实验步骤.....	7
实验 3 用Rational Rose绘制用例图 .....	12
3.1 实验环境.....	12
3.2 实验目的.....	12
3.3 实验内容.....	12
3.4 实验要求.....	12
3.5 实验步骤.....	12
实验 4 用Rational Rose工具绘制类图 .....	16
4.1 实验环境.....	16
4.2 实验目的.....	16
4.3 实验内容.....	16
4.4 实验要求.....	16
4.5 实验步骤.....	16
软件工程大作业.....	19
5.1 目的及要求.....	19
5.2 总体安排.....	19
5.3 任务.....	19
5.4 设计结果提交方式.....	20
5.5 教材和主要参考资料.....	20

## 第二篇 习题

第 1 章 概论.....	21
1.1 填空题.....	21

---

1.2 选择题.....	21
1.3 判断题.....	24
1.4 简答题.....	24
<b>第 2 章 软件工程过程模型.....</b>	<b>26</b>
2.1 填空题.....	26
2.2 选择题.....	26
2.3 判断题.....	28
2.4 简答题.....	28
2.5 应用题.....	29
<b>第 3 章 传统软件工程.....</b>	<b>30</b>
3.1 填空题.....	30
3.2 选择题.....	33
3.3 判断题.....	42
3.4 简答题.....	44
3.5 应用题.....	46
<b>第 4 章 面向对象基础.....</b>	<b>48</b>
4.1 填空题.....	48
4.2 选择题.....	49
4.3 判断题.....	51
4.4 简答题.....	52
4.5 应用题.....	53
<b>第 5 章 可行性分析及制定项目计划.....</b>	<b>54</b>
5.1 填空题.....	54
5.2 选择题.....	55
5.3 简答题.....	57
5.5 应用题.....	57
<b>第 6 章 面向对象分析.....</b>	<b>59</b>
6.1 填空题.....	59
6.2 选择题.....	60
6.3 判断题.....	63
6.4 简答题.....	64
6.5 应用题.....	65
<b>第 7 章 面向对象设计.....</b>	<b>66</b>
7.1 填空题.....	66
7.2 选择题.....	67
7.3 判断题.....	70
7.4 简答题.....	71
7.5 应用题.....	71

---

第 8 章 软件编码与测试 .....	73
8.1 填空题 .....	73
8.2 选择题 .....	74
8.3 判断题 .....	79
8.4 简答题 .....	80
8.5 应用题 .....	81
第 9 章 软件维护 .....	82
9.1 填空题 .....	82
9.2 选择题 .....	83
9.3 判断题 .....	87
9.4 简答题 .....	87
第 10 章 软件项目管理与质量保证 .....	89
10.1 填空题 .....	89
10.2 判断题 .....	90
10.3 简答题 .....	91
参考文献 .....	92



---

# 第一篇 实验指导





# 实验 1 用 PowerDesigner 建模工具绘制数据流图

## 1.1 实验环境

Windows XP、Sybase 公司的 PowerDesigner 工具软件。

## 1.2 实验目的

- (1) 了解 PowerDesigner 工具软件的组成及功能；
- (2) 掌握 PowerDesigner 中的 ProcessAnalyst (PA) 模块的建模方法及其工具的使用方法。

## 1.3 实验内容

- (1) 设计数据流图 (DFD)；
- (2) 利用外部实体、过程处理、数据流及数据存储绘制数据流图。

## 1.4 实验要求

建立一个 PAM (ProcessAnalyst Model) 的应用实例，绘制某大学教室预订系统的数据流图，注意：功能分解的细化程度应视具体情况而定，且数据流图适度分层。

项目的具体情况如下：

### (1) 背景

某大学需要设计一个教室预订系统，将空闲的教室提供给师生做学术报告或班级活动。师生可使用该系统预订所需的教室，以便让管理员在恰当的时间开放教室、准备好多媒体设备。

### (2) 功能要求

- ① 管理员可使用该系统增加用户、删除用户；
- ② 管理员可向系统中增加或删除供预订的教室，设置任一教室可使用的时间段；
- ③ 管理员可浏览任意时间段内的教室预订情况；

④ 师生可登录系统、修改密码；

⑤ 师生在登录后，可浏览给定时间段的空闲(供预订的且未被预订的)教室；选择教室和使用的时间段，并预订教室。

(3) 性能要求

① 教室占用的时间段精确到“课时”；

② 支持最多 10000 人同时在线使用该系统，确保系统稳定流畅。

## 1.5 实验步骤

### (1) PowerDesigner 简介

PowerDesigner 作为图形化的数据库模型设计工具软件，其集成特性灵活，采取模块化设计，共由以下六个模块组成：

- ◆ **ProcessAnalyst**：用于系统的需求分析，可用于设计和构造数据流图（DFD）和数据字典。它支持多种处理建模方法，用户可以选择适合自己应用环境的建模方法来描述系统的数据及对数据的处理。
- ◆ **DataArchitect**：用于对概念层和物理层的交互式数据库设计和构造。使用 DataArchitect 可以很方便地设计数据库的概念模型并对该 DBMS 自动生成物理模型。该模块还提供高质量的文档生成能力和逆向工程能力，可从现有的数据库中得到其物理模型和概念模型，并生成相应文档。
- ◆ **AppModeler**：应用建模工具，用于物理建模和应用对象及数据组件的生成。
- ◆ **MetaWorks**：用于团队开发、信息共享和模型管理。
- ◆ **WarehouseArchitect**：用于数据仓库和数据集的建模和实现。
- ◆ **Viewer**：用于以只读的、图形化方式访问整个企业的模型信息。

在本课程的实验部分，需要掌握 **ProcessAnalyst** 模块和 **DataArchitect** 模块的应用，鼓励同学在完成大作业的过程中了解其他模块。

**PowerDesigner 的模型有以下四种：**

- ◆ **PAM**：由 ProcessAnalyst 模块生成的处理分析模型。
- ◆ **CDM**：由 DataArchitect 模块生成的概念模型。
- ◆ **PDM**：由 DataArchitect 模块生成的物理模型。
- ◆ **WAM**：由 WarehouseArchitect 模块生成的数据仓库模型。

PowerDesigner 是将一个大型应用程序分成若干项目，每个项目可以包括模型，每个模型还可以包括若干个子模型。

PowerDesigner 中的对象包括项目、模型、子模型以及它们的各自属性。PowerDesigner 通过对这些属性的设置来描述其对象的特征，这些属性称之为对象的特性。一般情况下，对象的所有特性中只有名称和代码需要用户设置。

## (2) ProcessAnalyst (PA) 使用简介

ProcessAnalyst 是在数据库系统设计的需求分析阶段中，用于设计信息系统处理分析模型 PAM 的模块。PAM 的主要组成部分是数据流图（描述信息系统中所用到的基本数据及其处理分析过程）。

设计新系统的处理模型主要包括：数据流图、数据字典和处理逻辑表达工具等。本实验要求熟练使用 PowerDesigner 设计数据流图。数据流图是系统处理模型的主要组成部分，它只在模型上精确描述系统中数据和处理的关系，并详细描述系统的功能、输入、输出、数据存储等。而数据字典则是对数据的详细描述，是各类数据属性的清单。

数据流程图的设计采取自上而下的方法，首先将整个系统作为一个功能绘制顶层图，然后逐级细化，分解为多层次的数据流图。

下面以一个简单的教师管理系统为例演示 PowerDesigner 中的 ProcessAnalyst (PA) 模块的使用方法。

数据流图是系统处理模型的主要组成部分。其中心问题是把功能逐层分解为多个子功能。

在 Windows 桌面上双击 ProcessAnalyst 程序图标，打开 ProcessAnalyst 应用程序，在左上角出现如图 1.1 所示的工具箱。

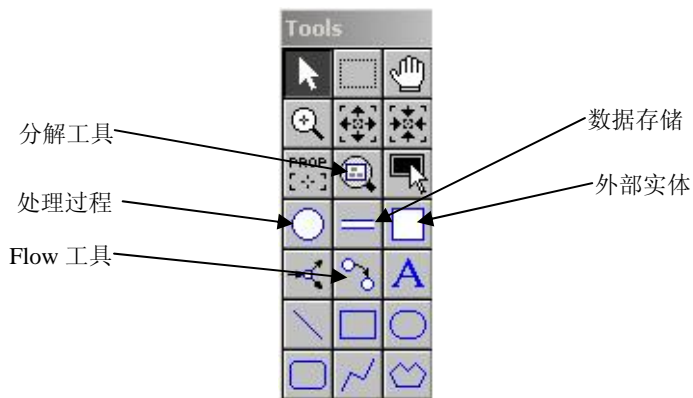


图 1.1 ProcessAnalyst 工具栏

#### (4) 数据流图设计

##### ① 建立根处理模型

S1: 建立一个处理过程 P1。单击工具栏中的 Process 工具，在模型工作区内单击会出现一个处理过程的符号，它是建立的第一个处理过程。单击鼠标右键释放 Process 工具，再双击新建立的处理过程符号，出现特性对话框，如图 1.2 所示，对该过程命名，单击 OK 按钮完成。

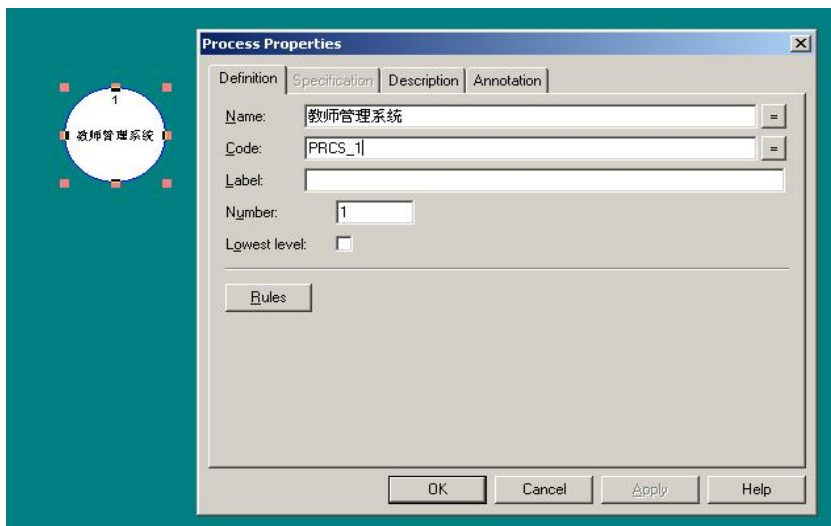


图 1.2 Process Properties 对话框

S2: 建立并定义外部实体。单击工具栏的 External Entity 工具，在模型工作区内单击，出现一个外部实体的符号。单击右键释放工具，同上，双击建立该外部实体的名字。单击 OK 按钮完成。

重复 S2 建立其它外部实体。

S3: 在对象间建立数据流。单击工具栏中的 Flow 工具，单击实体，并一直按住鼠标左键不放，拖到处理过程 P1 上，再释放鼠标左键，在两个对象之间出现了一个数据流。单击鼠标右键释放工具，双击该数据流符号对其命名。

重复 S3 建立其它数据流。

结果如图 1.3 所示。

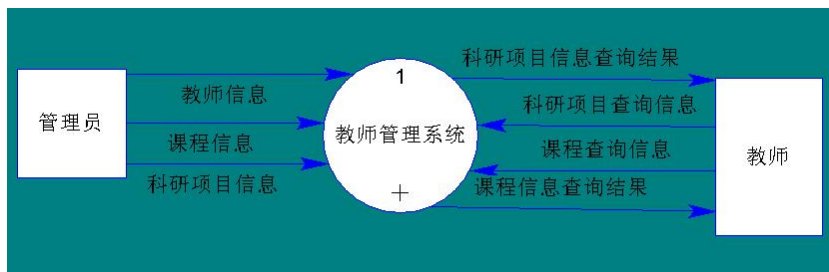


图 1.3 顶层数据流图示例

S1: 建立子处理窗口。单击工具栏的 Decomposition 工具，单击模型中的待分解的处理过程，出现该处理过程的子处理窗口 Subprocess。在子处理窗口中，外部实体和数据流以一种特殊形式出现，它们的名字两边有方括号，表明它们是从上一层引入的数据流。

S2: 建立和连接处理过程。在该子处理窗口新建立一个过程，双击该过程出现特性对话框，如果该层是最后一层，不需要对该过程分解了就选中 Lowest level 检查框。

S3: 在子处理窗口建立其它需要的外部实体、处理过程、数据流及数据存储，并把上一层引入的对象连接起来。

S4: 建立数据存储。单击工具栏的 Data Store 工具，在模型工作区中单击，出现一个数据存储符号。单击鼠标右键释放。双击对其命名，单击 OK 按钮完成。如果要对处理过程的有关数据进行存储，可按照 S4 在处理过程和数据存储间建立数据流。

如果还需要分解的处理过程，重复第②步即可。

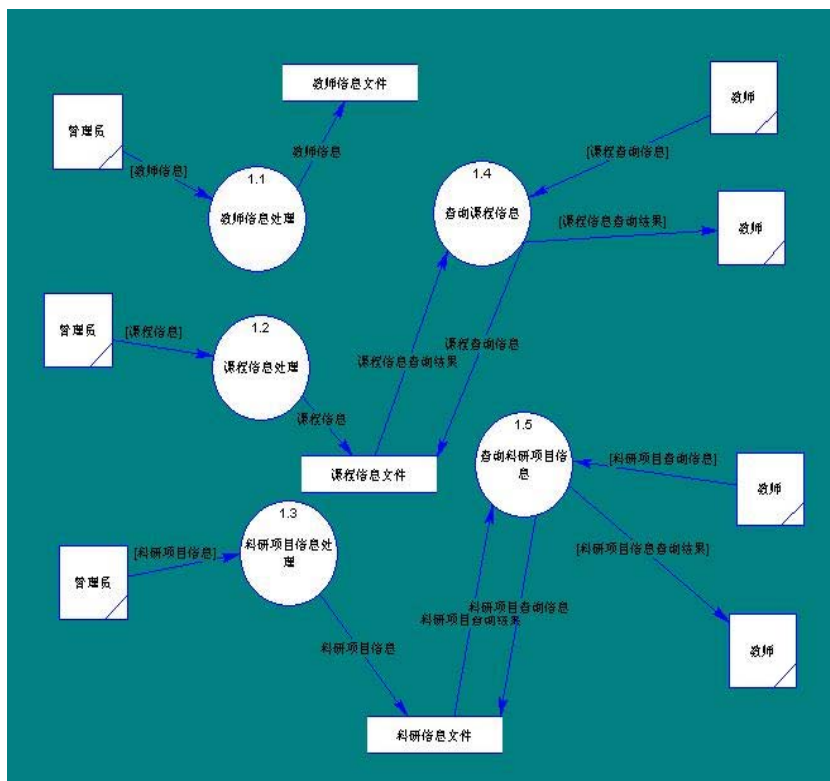


图 1.4 1 层数据流图示例

## 实验 2 用 PowerDesigner 建模工具设计数据库

### 2.1 实验环境

Windows XP、Sybase 公司的 PowerDesigner 工具软件。

### 2.2 实验目的

- (1) 掌握 PowerDesigner 中的 DataArchitect (DA) 模块的建模方法及其工具的使用方法；
- (2) 学会利用建模工具设计数据库。

### 2.3 实验内容

利用建模工具设计数据库。

### 2.4 实验要求

利用 PowerDesigner 中的 DataArchitect 模块，完成教室预订系统的数据库建模。  
项目的具体情况同实验 1。

### 2.5 实验步骤



#### (1) DataArchitect 简介

DataArchitect 是一种数据库设计工具，主要用于进行概念数据模型 (CDM) 和物理数据模型 (PDM) 的设计及其相互转换与维护。其中，概念数据模型就是 E-R 图，将现实的应用抽象为实体与实体之间的联系。CDM 的具体对象包括：域 (Domain)、数据项 (Data Item)、实体 (Entity)、实体属性、和继承链 (inheritance link) 等。而物理数据模型则针对某中 DBMS 定义物理层次上的各类数据对象 (包括表、域、列、参

照、码、索引、视图、扩展属性和检查参数等)。

## (2) DataArchitect 使用

下面以一个简单的教师管理系统为例，简介 PowerDesigner 中的 DataArchitect (DA) 模块的使用方法。

DataArchitect (DA) 是 Sybase 公司开发的用于数据库建模工具，为“Table”按钮，在模型工作区可以用来绘制数据库；为“Reference”按钮，可以用来连接两个或两个以上数据库表，并且定义主外键；

选择“开始”，再依次选择“程序”→“Sybase”→“PowerDesigner”→“DataArchitect”，启动 DataArchitect 程序。

## (3) 数据库设计

表结构设计如下：

职称表如下，主键为职称编号：

职称编号	职称名称

教师信息表，主键为教师编号，外键为职称编号：

教师编号	教师姓名	职称编号	年龄	性别	家庭住址	联系方式

## (3) 利用 DA 建立数据库模型

### ① PDM 文件的建立

进入“DataArchitect”，依次选择“File”→“Physic Model”→“New”，新建一个 PDM 文件，如图 1.5 所示。

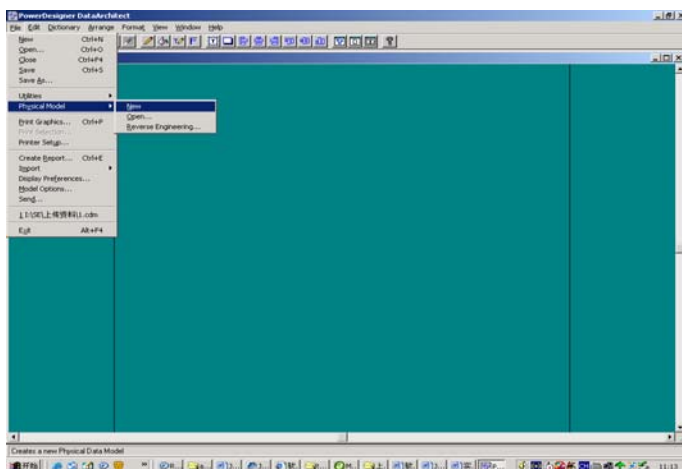


图 1.5 “DataArchitect”的主界面



② 在 Database name 处选择数据库类型，点击“OK”按钮确定。

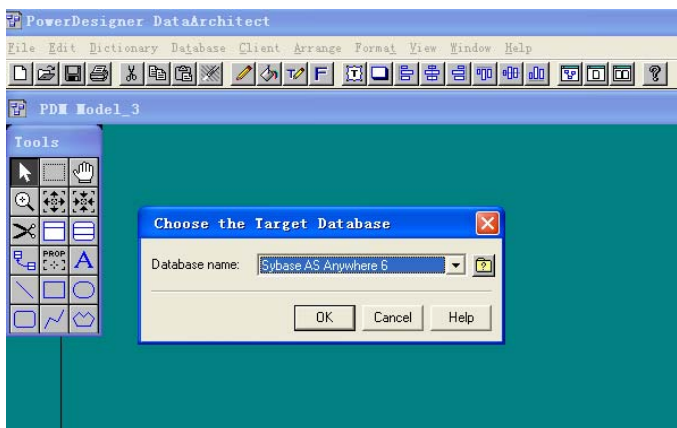


图 1.6 新建 PDM 文件界面

在工具栏 (Tools) 中，单击“Table”按钮，然后在模型工作区内单击会出现一个空表的符号，如图 1.7 所示，需要几个空表就重复上述操作几次。

③ 表的建立及各字段的编辑

双击空表，在弹出的“Table Properties”对话框中，点击“Columns”按钮，根据事先设计的表，填写各字段和属性，如图 1.8 所示。

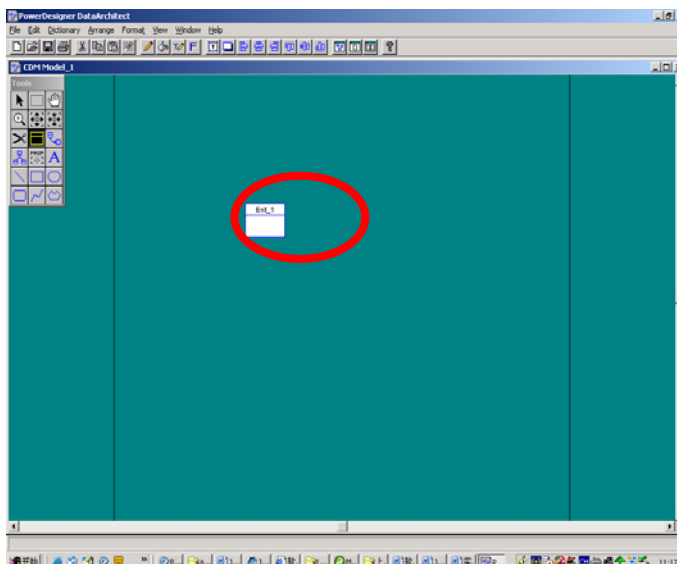


图 1.7 新建一张空表界面

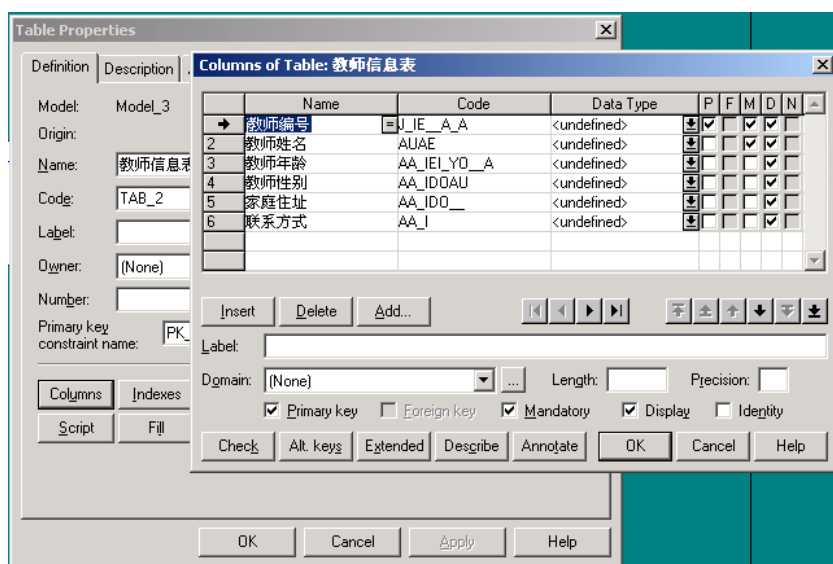


图 1.8 编辑字段信息界面

## ④ 主外键关系的建立

职称表的主键为职称编号，教师信息表主键为教师编号，外键为职称编号，如图 1.9 所示。

单击 Tools 工具栏的“Reference”按钮，然后将把鼠标移到模型工作区内，选择一张数据表，按住鼠标不放开并移至另一张数据表上，如图 1.10 所示。

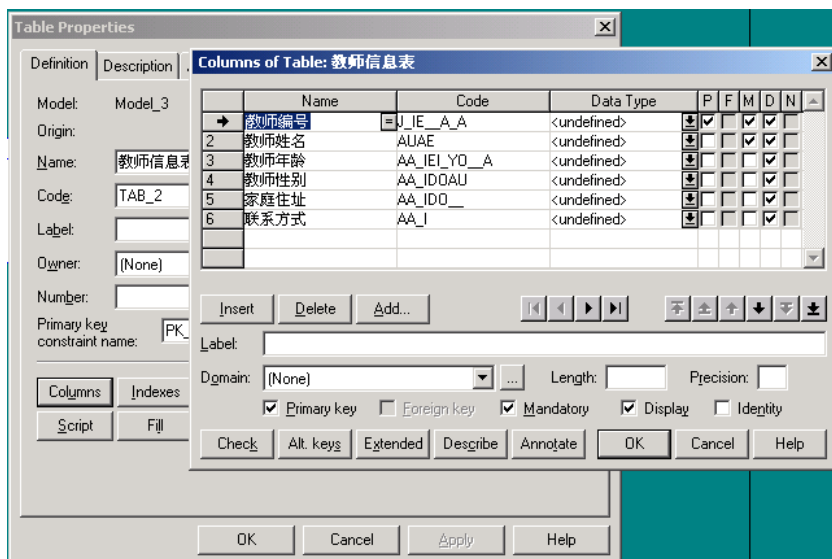


图 1.9 设置主、外键界面

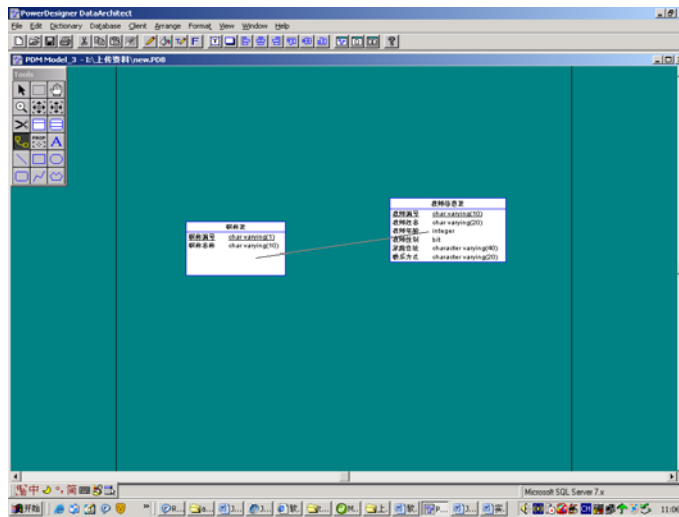


图 1.10 建立表间关系界面

结果如图 1.11 所示。

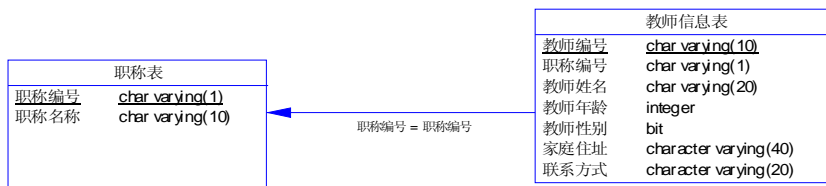


图 1.11 建立的数据表

## 实验 3 用 Rational Rose 绘制用例图

### 3.1 实验环境

Windows XP、Rational Software 公司的 Rational Rose 工具软件。

### 3.2 实验目的

- (1) 了解 Rational Rose 工具软件的组成及功能；
- (2) 掌握用 Rational Rose 进行用例建模的方法。

### 3.3 实验内容

- (1) 设计用例图（Use Case 框图）；
- (2) 用 Rational Rose 在 Use Case 视图中创建 Use Case 框图。

### 3.4 实验要求

建立一个 Use Case 框图，并给出“预订教室”的用例描述。  
项目的具体情况同实验 1。

### 3.5 实验步骤

#### (1) Rational Rose 简介

Rational Rose 是由美国的 Rational 公司开发的、面向对象的可视化建模工具。利用 Rose 工具，可以建立用 UML 描述的软件系统模型，它支持 UML 中的用例框图、活动框图、序列框图、协作框图、状态框图、组建框图和布局框图等等；而且，通过其正向和逆向转出工程代码的特性，可以支持 C++、Java、Visual Basic 和 XML DTD 的代码生成和逆向转出工程代码。

Rose 提供了一套十分友好的界面让用户对系统进行建模。安装完 Rose 之后，依次选择“开始”→“程序”→“Rational Software”→“Rational Rose Enterprise Edition”命令，弹出如图 1.12 所示的界面，选择 Rose 新模型的应用架构（Framework）界面。开发人员可以选择 J2EE、J2SE1.2、J2SE1.3、jfc-11 等应用框架进行系统分析和设计。

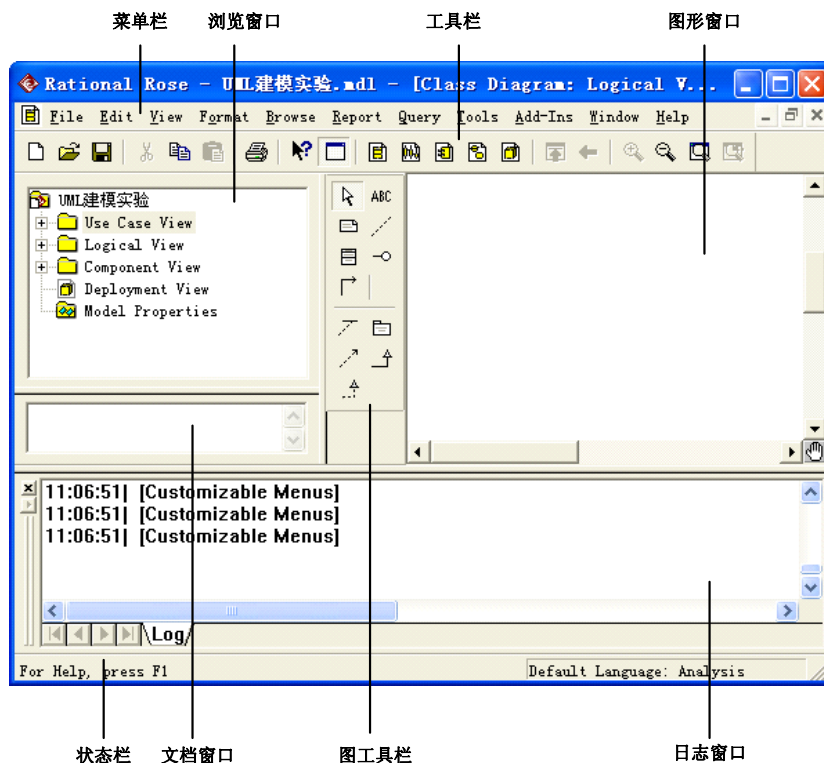


图 1.12 Rose 建模界面

Rational Rose 模型提供了 4 种视图：用例视图（Use Case View）、逻辑视图（Logical View）、组件视图（Component View）和部署视图（Deployment View）。每当创建一个新的 Rose 模型时（扩展名为.mdl），Rose 将自动生成上述视图，Rose 把视图看作模型结构的第一层次。每种视图针对不同的对象，具有不同的用途。

在本课程的实验部分，需要掌握用例图和类图的绘制，鼓励同学在完成大作业的过程中了解其他图的绘制。

## (2) Use Case 框图

Use Case 框图表示整个机构提供的功能，可以用来回答下列一些问题：公司是干什么的？为什么要建立这个系统？还有那些人使用这些系统。Use Case 框图在业务建模活动期间大量用于设置系统情景和形成创建使用案例的基础。

## ① 使用 Rational Rose 画 Use Case 框图

S1: 如图 1.13 所示, 依次选择“开始”→“程序”→“Rational rose”→“Rational Rose Enterprise Edition”, 进入该软件, 如图 1.12 所示。

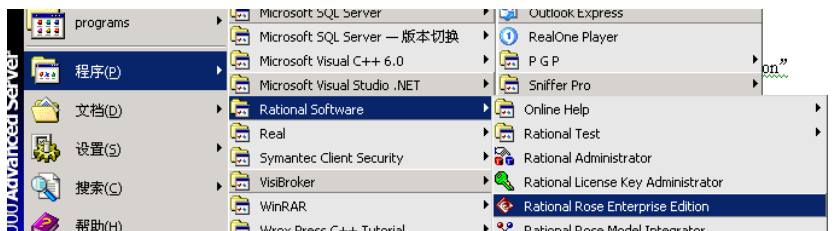


图 1.13 Rational Rose 的启动

S2: 如图 1.14 所示, 在左边“浏览窗口”双击“Use Case View”之后, 出现一个“main”, 双击“main”弹出一个界面, 此时, 可以开始用例图的绘制。

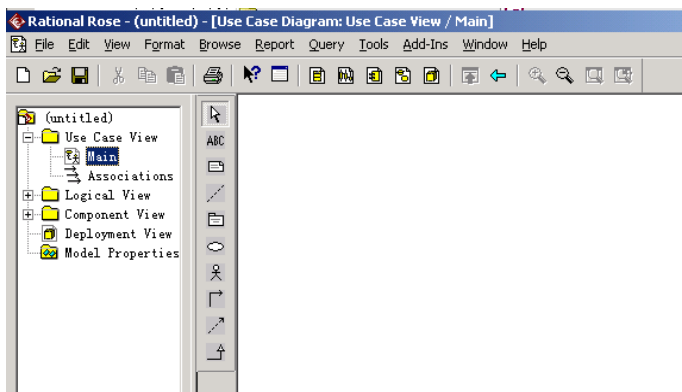


图 1.14 “Use Case View”的“main”界面

S3: 图 1.14 中, 稍微靠左的位置, 可见 Use Case 框图工具栏图标, 各图标代表的含义如图 1.15 所示。

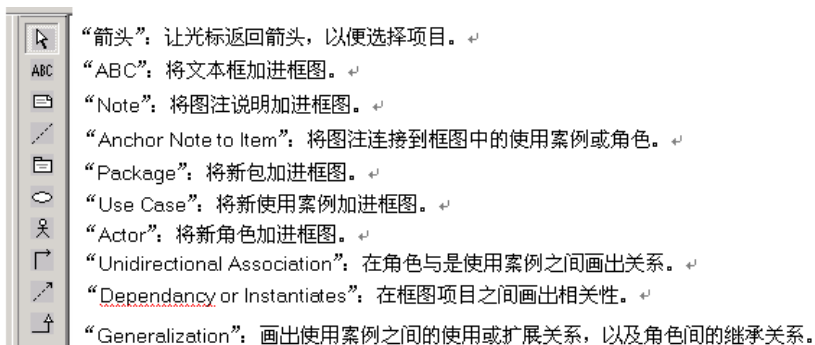


图 1.15 “Use Case”的“main”界面框图工具栏

## ② 绘制与保存

S1: 根据预习实验所画的用例图草稿和 Use Case 工具栏图标绘制用例图;

S2: 在绘制完成之后点击保存, 弹出选择保存位置的对话框, 选择或输入地址进行保存即可。

### (3) “Use Case View”框图应用实例

以“教师管理系统”为例, “Use Case”框图显示“教师管理系统”使用案例与角色间的交互。本例中, 管理员启动几个使用案例: 基本信息管理、课程管理、教师职称管理、科研项目管理、基本课程查询、科研项目查询; 教师可以启动基本课程查询、科研项目查询使用案例。箭头从使用案例到角色表示使用案例产生一些角色要使用的信息。

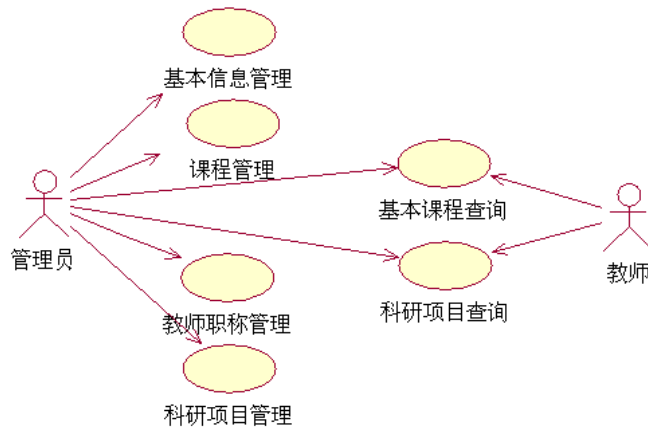


图 1.16 “教师管理系统”的用例图

## 实验 4 用 Rational Rose 工具绘制类图

### 4.1 实验环境

Windows XP、IBM 公司的 Rational Rose 工具软件。

### 4.2 实验目的

- (1) 理解类及类间关系的基本概念；
- (2) 掌握如何从需求分析中抽象出类的方法；
- (3) 掌握描绘类间关系的方法；
- (4) 了解 Rational Rose 工具软件的组成及功能；
- (5) 掌握在 Rational Rose 中绘制类及类关系的操作方法。

### 4.3 实验内容

- (1) 寻找和抽象出指定教室预订系统中的实体类；
- (2) 对实体类的关系建模。

### 4.4 实验要求

利用 Rational Rose 工具绘制教室预订系统的类图。

项目的具体情况同实验 1。

### 4.5 实验步骤

#### (1) 类图简介

类图表示不同的实体（人、事物和数据）如何彼此相关；换句话说，它显示了系统的静态结构。类图可用于表示逻辑类，逻辑类通常就是业务人员所谈及的事物种类——



摇滚乐队、CD、广播剧；或者贷款、住房抵押、汽车信贷以及利率。类图还可用于表示实现类,实现类就是程序员处理的实体。实现类图或许会与逻辑类图显示一些相同的类。然而,实现类图不会使用相同的属性来描述,因为它很可能具有对诸如 Vector 和 HashMap 这种事物的引用。

## (2) 新建类

依次选择“开始”→“程序”→“Rational Software”→“Rational Rose Enterprise Edition”命令,弹出如图 1.12 所示的界面,系统自动建立一个新的.mdl 文件,在此文件中,可以绘制类图。

界面右边的空白“图形窗口”区域为工作区,在工作区可进行绘图操作。

界面中间的动作条为绘图元素,如图 1.17 所示,可以使用这些元素进行绘图。

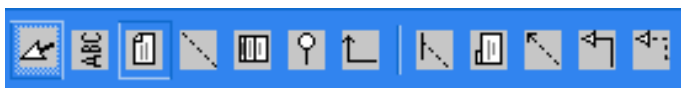


图 1.17 “Rational Rose”的图工具栏

在工具栏中,选择 Class 元素,在绘图区中单击左键,建立一个空白类,如图 1.18 所示。

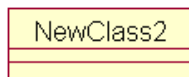


图 1.18 建立的空白类

如图 1.19 所示,类的 UML 表示是一个长方形,垂直地分为三个区。顶部区域显示类的名字。中间的区域列出类的属性,底部的区域列出类的操作。

当在一个类图上画一个类元素时,必须要有顶端的区域,下面的二个区域是可选择的(当图描述仅仅用于显示分类器间关系的高层细节时,下面的两个区域不是必要的)。

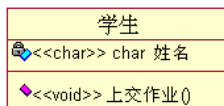


图 1.19 建立的学生类

## (3) 类的编辑

如图 1.19,类名是“学生”,在中间区域看到学生类的 1 个属性:姓名,在底部区域可以看到学生类的操作:上交作业。

左键单击空白类的 ClassName,可更改类的名字。

右键点击新类,选择 New Attribute,这时新类中将会多一个属性。

左键单击新的属性，更改成需要的名字和类型。

右键单击新类，选择 New Operation，为类添加新的函数或操作。

以同样的方法再建立一个新类“大学生”，结果如图 1.20 所示。

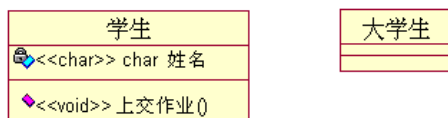


图 1.20 两个类

在面向对象的设计中，继承是一个非常重要的概念，指的是一个类（子类）继承另外的一个类（超类）的同一功能，并增加它自己的新功能（一个非技术性的比喻，想象张三继承了张三母亲的一般的音乐能力，但是在张三的家里，张三是唯一一个玩电吉他的人）的能力。为了在一个类图上建模继承，从子类（要继承行为的类）拉出一条闭合的，单键头（或三角形）的实线指向超类。示例如图 1.21 所示。

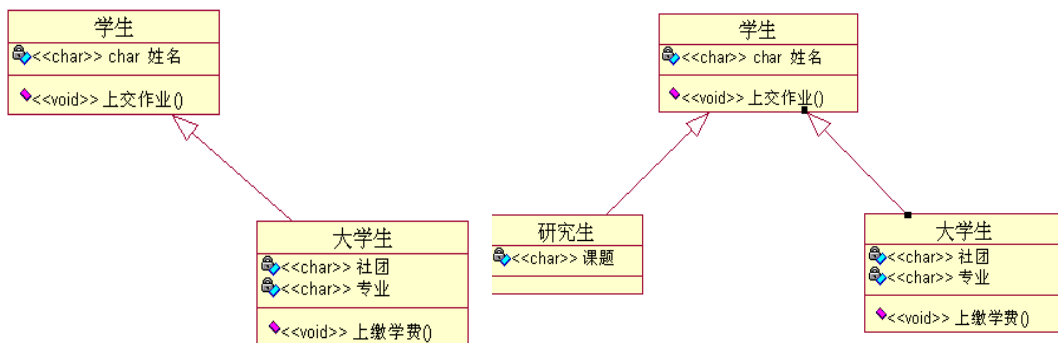


图 1.21 两个类

进行系统建模时，特定的对象间将会彼此关联，而且这些关联本身需要被清晰地建模。关联是两个类间的联接。关联总是被假定是双向的；这意味着，两个类彼此知道它们间的联系，除非限定一些其它类型的关联。

一个双向关联用两个类间的实线表示。在线的任一端，放置一个角色名和多重值。



图 1.22 两个类的关联

如图 1.22 所示，学生类与教师类相关联。

# 软件工程大作业

## 5.1 目的及要求

《软件工程》是一门实践性极强的课程。进行《软件工程大作业》的目的就是通过一个实际的项目，促进学生有针对性地主动学习和查阅软件工程的基本教学内容和相关资料，从而培养学生完整、严格的软件工程概念、意识和能力。一般来讲，大作业比课程实验更复杂一些，涉及的尝试更广一些，并更加注重实际应用，重点培养学生实际分析问题和解决问题的能力，最终目标是通过大作业的形式，帮助学生系统掌握软件工程课程的主要内容，更好地完成软件工程课程的教学任务。

大作业在教师的指导下进行，以实际应用中自选的中、小型题目为主，根据不同的题目类型选择一种开发模式，完成从系统定义、软件项目计划、软件需求分析、软件设计、编程、测试，直到运行的软件生存周期的全过程。

要求学生能把软件工程的基本原理和方法应用到软件的实际开发中。强化软件开发的团队意识，提高合作能力，让开发人员控制一个开发项目并按时提交一个完整的、正确的、结构良好的而且健壮的项目。

## 5.2 总体安排

大作业一般完成较大的综合设计，每个班依据实际情况分成几个项目小组进行自主软件开发（分析、设计、编程、测试）、教师辅导答疑为辅的教学方式。

要求每个小组在本学期内完成一个项目，各个项目成品的交付日期必须严格遵守。

## 5.3 任务

(1) 选题：根据课程设计的要求，由学生自行选定题目。如：学生学籍管理系统，教务管理系统，进、销、存管理系统，图书销售和管理系统，物业管理系统，酒店管理系统，网上商城管理系统，饭卡管理系统等等。

(2) 按时、独立完成、排版规范，鼓励参考网上的系统(文档)或某些专业公司的文档规范，按自己的选题进行改写。

(3) 提交的设计报告必须包含如下的 8 个文档：

- ◆ 系统规格说明书(含可行性分析报告)
- ◆ 软件项目计划
- ◆ 软件需求规格说明书(含初步用户手册、确认测试计划)
- ◆ 软件设计说明书
- ◆ 模块开发卷宗(源代码清单：电子版)和单元测试
- ◆ 软件测试计划和测试分析报告
- ◆ 用户（安装、操作）手册
- ◆ 项目开发总结报告

(4) 课程设计报告正文统一用小四号宋体字，使用 A4 纸打印，要求每个小组的设计报告不少于 100 页。

(5) 文档基本符合国家标准。

(6) 工作量：每小组因选题不同，可以略有不同，不含空行和注释的(自己编写的)程序量要求在 1.0 至 2.0 万行程序之间。

(7) 开发工具：Java 或 C++。

## 5.4 设计结果提交方式

(1) 每个小组提交整个小组 1 份完整的文档(含上述规定的 8 个文档)和所设计系统的源程序和可执行程序。

(2) 每人需提交 1 份自己负责部分的相关说明。

(3) 除提交打印的文档外，每个小组还必须提交电子版的文档和程序。

## 5.5 教材和主要参考资料

中国标准出版社. 计算机软件工程规范国家标准汇编（第二版）[M]. 中国标准出版社，2011.08

---

## 第二篇 习题



## 第1章 概论

### 1.1 填空题

- (1) 构成一个完整的计算机系统的两部分是硬件与\_\_\_\_\_。
- (2) 程序设计时代的生产方式是\_\_\_\_\_。
- (3) 软件与物质产品有很大的区别，软件是一种\_\_\_\_\_产品。
- (4) 软件危机指的是：\_\_\_\_\_。
- (5) 软件工程学的内容可包括：\_\_\_\_\_, \_\_\_\_\_, \_\_\_\_\_, \_\_\_\_\_。
- (6) 软件工程的原则是：\_\_\_\_\_。
- (7) 软件工程研究的主要内容是软件开发技术和\_\_\_\_\_两个方面。
- (8) 软件工具是支持软件开发人员的开发和维护活动而使用的\_\_\_\_\_。
- (9) 软件工程涉及到几门学科，它是一门\_\_\_\_\_学科。
- (10) 软件是计算机系统中的\_\_\_\_\_, \_\_\_\_\_、数据及\_\_\_\_\_的总称。

### 1.2 选择题

- (1) 软件是一种( )产品。  
A. 物质  
B. 逻辑  
C. 有形  
D. 消耗
- (2) 软件产品的开发主要是( )。  
A. 复制  
B. 再生产  
C. 剪贴  
D. 研制
- (3) 个体手工劳动生产方式的时代是( )时代。  
A. 程序设计  
B. 软件生产自动化  
C. 程序系统  
D. 软件生产
- (4) 同一软件的大量软件产品的生产主要是通过( )而得到的。

- A. 研究
  - B. 复制
  - C. 开发
  - D. 研制
- (5) 作坊式小团体合作生产方式的时代是( )时代。
- A. 程序设计
  - B. 软件生产自动化
  - C. 程序系统
  - D. 软件生产
- (6) 软件开发费用只占整个软件系统费用的( )。
- A. 1/2
  - B. 1/3
  - C. 1/4
  - D. 2/3
- (7) 随着开发小组人数的( )，因交流开发进展情况和讨论遇到的问题而造成的通信开销也急剧增加。
- A. 增加
  - B. 降低
  - C. 稳定
  - D. 不稳定
- (8) 下面( )不是系统软件。
- A. BIOS
  - B. Windows
  - C. 设备驱动程序
  - D. 办公软件
- (9) ( )是将系统化的、规范的、可定量的方法应用于软件的开发、运行和维护的过程，它包括方法、工具和过程三要素。
- A. 软件过程
  - B. 软件测试
  - C. 软件生存周期
  - D. 软件工程
- (10) 在软件开发的各种资源中，( )是最重要的资源。
- A. 开发工具
  - B. 方法
  - C. 硬件环境
  - D. 人员
- (11) 软件工程的基本要素包括方法、工具和( )。
- A. 过程
  - B. 软件系统
  - C. 硬件环境
  - D. 人员
- (12) 软件产品的生产主要是( )。
- A. 制造
  - B. 复制
  - C. 开发
  - D. 研制
- (13) 软件工程是一门( )学科。
- A. 理论性
  - B. 原理性



- 23

- (21) 软件工程的目标有( )。
- A. 易于维护
  - B. 低的开发成本
  - C. 高性能
  - D. 短的开发期
- (22) 软件工程学的目的和意义是( )。
- A. 应用科学的方法和工程化的规范管理来指导软件开发
  - B. 克服软件危机
  - C. 作好软件开发的培训工作
  - D. 以较低的成本开发出高质量的软件

### 1.3 判断题

- (1) ( ) 一个成功的项目唯一提交的就是运行程序。
- (2) ( ) 软件工作的考虑范围主要是程序设计和实现。
- (3) ( ) 在软件开发的过程中，若能推迟暴露其中的错误，则为修复和改正错误所花费的代价就会降低。
- (4) ( ) 软件错误可能出现在开发过程的早期，越早越好修改。
- (5) ( ) 软件工具的作用是为了延长软件产品的寿命。
- (6) ( ) 软件开发小组的组成成员的素质应该好，而人数则不宜过多
- (7) ( ) 软件就是程序，编写软件就是编写程序。
- (8) ( ) 软件危机的主要表现是软件需求增加，软件价格上升。
- (9) ( ) 与计算机科学的理论研究不同，软件工程是一门原理性的科学。
- (10) ( ) 软件工程学科出现的主要原因是软件危机的出现。

### 1.4 简答题

- (1) 什么是软件？软件产品的特性是什么？
- (2) 软件开发过程的大致分为几个时期？都是什么？
- (3) 软件危机表现在几个方面？
- (4) 有人说：软件开发时，一个错误发现的越晚，为改正它所付出的代价就越大。请问对否？请解释你的回答？

- (5) 软件工程是开发、运行、维护和修复软件的系统化方法，它包括哪些要素？试说明之。
- (6) 软件工程学的基本原理是哪些？试说明之。
- (7) 软件工程学讲的是什？
- (8) 什么是软件工程？
- (9) 软件工程的目标是什么？
- (10) 软件危机产生的原因是什么？某软件公司抢时间为某单位开发了一个人事管理软件，但软件交付用户使用一段时间之后，用户有了抱怨，原因是单位里某个职工改了名字，但人事管理软件却不允许修改姓名，而只能删除整条记录以后重新输入。试从软件危机角度对这个问题做一些评论。
- (11) 软件工程的研究内容是什么？
- (12) 软件工程需要解决的问题是什么？
- (13) 你认为一个好的软件开发人员应具备哪些基本素质？
- (14) 试比较软件发展的不同时期的特点，从软件工作的范围、软件开发组织、决定质量的因素、开发技术和手段等几个方面说明它们的差别。
- (15) 简述软件的种类。
- (16) 什么是 **SWEBOK**？由哪些知识域组成？
- (17) 阅读《人月神话》，谈谈你对软件危机和软件工程发展历程的理解。

## 第 2 章 软件工程过程模型

### 2.1 填空题

- (1) 软件生存周期分三个阶段，分别是：\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_。
- (2) Jackson 图中，各数据元素之间的逻辑关系有\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_三种。
- (3) 使用原型法开发软件必须依赖\_\_\_\_\_工具。
- (4) 原型法适用于开发较复杂的系统，原型可分为\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_。
- (5) 国家标准“计算机软件开发规范”中，把软件生存周期划分为 8 个阶段，即\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_。
- (6) 螺旋模型将开发过程分为几个螺旋周期，在每个螺旋周期内分为\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_四个步骤。
- (7) 软件工程过程包含的 7 个过程是\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_。

### 2.2 选择题

- (1) JSP 方法是一种面向( )的设计方法。  
A. 对象  
B. 数据流  
C. 控制结构  
D. 数据结构
- (2) 软件过程是( )。  
A. 特定的开发模型  
B. 一种软件求解的计算逻辑  
C. 软件开发活动集合  
D. 软件生存周期模型
- (3) 包含风险分析的软件工程模型是( )。  
A. 螺旋模型  
B. 瀑布模型

- 27

## 2.3 判断题

- (1) (        ) 软件工程过程应该以软件设计为中心，关键是编写程序。
- (2) (        ) 瀑布模型的最大优点是将软件开发的各个阶段分得十分清晰。
- (3) (        ) 原型化开发方法包括生成原型和实现原型两个步骤。
- (4) (        ) 文档只起备忘录的作用，可以在软件开发完成后再整理生成。
- (5) (        ) 软件工作直到软件交付使用为止。
- (6) (        ) 用面向对象方法分析、设计、实现软件，仍属线性的瀑布开发模型。
- (7) (        ) 在软件开发中采用原型系统策略的主要困难是成本问题。
- (8) (        ) Jackson 方法最适合在详细设计阶段，面向数据结构来设计每个模块的处理过程。
- (9) (        ) 面向对象的开发方法将过程作为最基本元素，是分析问题、解决问题的核心。

## 2.4 简答题

- (1) 软件生命期的阶段任务是什么？
- (2) 什么是软件生存周期模型？它有哪些主要模型？
- (3) 软件工程过程有哪几个基本过程活动？
- (4) 什么是软件过程？它与软件工程方法学有何关系？
- (5) 试说明“软件生存周期”的概念？
- (6) 试论述用瀑布模型进行软件开发的基本过程？
- (7) 软件生存周期各阶段的任务和目的各是什么？
- (8) 结构化分析、设计方法的实质是什么？有何缺点？
- (9) 什么叫面向对象？面向对象方法的特点是什么的？为什么要用面向对象方法开发软件？
- (10) 面向对象开发方法与面向数据流的结构化开发方法有什么不同？使用面向对象开发方法的优点是什么？
- (11) 简述 Jackson 方法的设计步骤。
- (12) 什么是软件开发方法？有哪些主要方法？

- (13) 软件生存周期模型的种类及各种模型的特点是什么？
- (14) 试比较各种模型在步骤和步骤顺序之间的差异。
- (15) 用例驱动是如何体现的？
- (16) 用例驱动和测试驱动有何差异？各适用于什么样的软件开发环境？

## 2.5 应用题

- (1) 举例说明哪些项目的开发适合用原模型和螺旋模型，哪些不适合采用这两种模型？
- (2) 原型进化模型是一种与瀑布模型有着显著差别的软件过程模型。与瀑布模型相比，其优点是什么？一般认为，原型进化模型不能适应较大型软件项目的开发，其原因是什么？
- (3) 极限编程在编码前是否要形成完整的设计文档？应用该模型应该如何保证软件的质量？阅读相关的参考书籍和其它资料，找出实例来加以证实。
- (4) 某大型企业计划开发一个“综合信息管理系统”，涉及销售、供应、财务、生产、人力资源等多个部门的信息管理。该企业的想法是按部门优先级别逐个实现，边应用边开发。对此，需要一种比较合适的过程模型。请对这个过程模型作出符合应用需要的选择，并说明选择理由
- (5) 某公司的软件产品以开发实验型的新软件为主。用瀑布模型进行软件开发已经有近十年了，并取得了一些成功。若作为一名管理员刚加入该公司，你认为快速原型法对公司的软件开发更加优越，请向公司副总裁写一份报告阐明你的理由，切忌：副总裁不喜欢报告长度超过一页(A4)。

## 第3章 传统软件工程

### 3.1 填空题

- (1) 需求分析阶段产生的最重要的文档是\_\_\_\_\_。
- (2) 需求分析阶段，分析人员要确定对问题的综合需求，其中最重要的是\_\_\_\_\_。
- (3) 需求分析是指开发人员要准确理解\_\_\_\_\_，进行细致的\_\_\_\_\_，将用户非形式的需求陈述转化为完整的\_\_\_\_\_，再由\_\_\_\_\_转换到相应的形式功能规约(需求规格说明)的过程。
- (4) 需求分析阶段必须明确系统中应具备的每一个加工、加工的处理对象和加工所引起的\_\_\_\_\_。
- (5) 系统需求的逻辑表示用于指明系统所要达到的\_\_\_\_\_和需求处理的数据，不涉及实现的细节。系统需求的物理表示用于指明\_\_\_\_\_和数据结构的实际表现形式，通常由系统中的设备决定。
- (6) 需求分析的基本任务是准确地回答“\_\_\_\_\_”这个问题。
- (7) 分析系统的数据要求通常采用建立\_\_\_\_\_的方法。
- (8) 编写文档包括编写需求说明书、初步用户使用手册、\_\_\_\_\_和修改完善项目开发计划。
- (9) 需求分析方法有功能分解方法、结构化分析方法、信息建模方法和\_\_\_\_\_等。
- (10) 软件需求开发的基本过程包括了需求获取、需求分析、编写需求规格说明书以及\_\_\_\_\_ 4个步骤。
- (11) 需求获取的基本技术有用户面谈、专题讨论会、问卷调查和原型化方法等，随着面向对象技术的发展，\_\_\_\_\_在需求获取和建模方面应用越来越普遍。
- (12) 需求分析阶段最后一步工作是将对系统分析的结果用\_\_\_\_\_的形式表示出来，以此作为审查需求分析阶段工作完成情况的依据和设计阶段开展工作的基础。
- (13) 编写需求规格说明的原则是：只描述\_\_\_\_\_而无需描述\_\_\_\_\_。



- (14) 需求的完整性是指目标系统的需求必须是全面的，需求规格说明书中应包括\_\_\_\_\_。
- (15) 软件需求可以分为不同的层次：业务需求、用户需求以及\_\_\_\_\_和\_\_\_\_\_。
- (16) 非功能需求由三个方面构成：\_\_\_\_\_、\_\_\_\_\_和外部需求。
- (17) 对于非功能需求的表达要求尽可能的量化且\_\_\_\_\_。
- (18) 需求管理包括在工程进展过程中维护需求约定集成性和精确性的所有活动，主要包括变更控制、\_\_\_\_\_和\_\_\_\_\_等活动。
- (19) 需求分析的困难主要体现在问题的复杂性、\_\_\_\_\_、\_\_\_\_\_和\_\_\_\_\_。
- (20) 在验证需求时，每一项需求都可以在早期的文档中追溯到其来源，这是需求的\_\_\_\_\_性。
- (21) 软件设计阶段产生的最重要的文档是\_\_\_\_\_。
- (22) 软件结构是以\_\_\_\_\_为基础而组成的一种控制层次结构。
- (23) 反映软件结构的基本形态特征是\_\_\_\_\_。
- (24) 一个模块把数值作为参数送给另一个模块，这种耦合方式称为\_\_\_\_\_。
- (25) 两个模块通过全程变量相互作用，这种耦合方式称为\_\_\_\_\_。
- (26) 将同一张报表有关的所有程序段组成一个模块，该模块的内聚性为\_\_\_\_\_。
- (27) 一个模块的作用范围指\_\_\_\_\_的集合。
- (28) 一个模块的控制范围指\_\_\_\_\_的集合。
- (29) 结构化设计以\_\_\_\_\_为基础映射成软件结构。
- (30) 为了防止软件概要设计的错误传播到开发的后续阶段，在概要设计文档完成以后，进行\_\_\_\_\_。
- (31) 在结构化分析中，用于描述加工逻辑的主要工具有三种，即结构化语言、判定表和\_\_\_\_\_。
- (32) 数据流图仅反映系统必须完成的逻辑功能，所以它是一种\_\_\_\_\_模型。
- (33) 数据流图(DFD)中的箭头代表\_\_\_\_\_。
- (34) 在结构化分析方法中，用于描述系统中所用到的全部数据和文件的文档称为\_\_\_\_\_。
- (35) 软件设计中划分程序模块通常遵循的原则是使各模块间的耦合尽可能\_\_\_\_\_。

- (36) 为高质量地开发软件项目，在软件结构设计时必须遵循\_\_\_\_\_的原则，  
\_\_\_\_\_建立软件系统的模块结构。并且应根据\_\_\_\_\_评价系统模块划分的质量。此外在模块设计时，应从3种基本的\_\_\_\_\_出发，利用它们组合成一个模块的程序结构。要求每个\_\_\_\_\_的结构应是单入口和单出口。
- (37) 在完成软件概要设计，并编写出相关文档之后，应当组织对概要设计工作的评审。评审的内容包括：分析该软件的系统结构、子系统结构，确认该软件设计是否覆盖了所有已确定的软件需求，软件每一成分是否可\_\_\_\_\_到某一项需求。分析软件各部分之间的联系，确认该软件的内部接口与外部接口是否已经明确定义。模块是否满足\_\_\_\_\_和\_\_\_\_\_的要求。模块\_\_\_\_\_是否在其\_\_\_\_\_之内。
- (38) 结构化设计方法中，要把数据流图转换成软件结构，若某个加工将它的输入流分离成许多发散的数据流，形成许多加工路径，并根据输入的值选择其中一个路径在执行，这种特征的DFD称为\_\_\_\_\_的数据流图。
- (39) 模块的独立程度可以由两个定性标准度量，这两个标准分别称为\_\_\_\_\_和\_\_\_\_\_。
- (40) 结构化设计中以数据流图为基础的两种具体设计方法是\_\_\_\_\_设计和\_\_\_\_\_设计。
- (41) 结构化程序设计方法的要点是使用\_\_\_\_\_结构，自顶向下、逐步求精地构造算法或程序。
- (42) 为了产生结构化的流程图，应由三种基本控制结构顺序组合或\_\_\_\_\_而成。
- (43) 结构化程序设计方法提倡开发人员的组织形式是\_\_\_\_\_。
- (44) PAD图是一种由左向右展开的二维树型结构，图中的竖线为程序的\_\_\_\_\_。
- (45) 在详细设计阶段，一种历史最悠久、使用最广泛的描述程序逻辑结构的工具是\_\_\_\_\_。
- (46) 结构化程序设计技术指导人们用良好的思想方法开发\_\_\_\_\_的程序。
- (47) 详细描述处理过程常用的三种工具是图形、语言和\_\_\_\_\_。
- (48) PDL具有严格的关键字外层语法、用于定义控制结构、数据结构和\_\_\_\_\_。
- (49) 详细设计的目标不仅是逻辑上正确地实现每个模块的功能，还应使设计出的处理过程\_\_\_\_\_。
- (50) 在详细设计阶段，除了对模块内的算法进行设计，还应对模块内的\_\_\_\_\_进

- (51) 在详细设计阶段, 为了提高数据的输入、储存、检索等操作的效率并节约存储空间, 对某些数据项的值要进行\_\_\_\_\_设计。
- (52) 详细设计也称\_\_\_\_\_, 它是软件设计的第二阶段, 主要确定每个\_\_\_\_\_的具体执行过程。
- (53) 过程设计中最典型的方法是\_\_\_\_\_, 其基本思想是\_\_\_\_\_、\_\_\_\_\_。
- (54) PAD 图的控制流程为\_\_\_\_\_, \_\_\_\_\_地执行。
- (55) 进行程序设计语言选择时, 优先考虑问题的\_\_\_\_\_。
- (56) 结构化程序设计方法简称\_\_\_\_\_。PAD 图又称\_\_\_\_\_。
- (57) 过程涉及语言简称\_\_\_\_\_。它是伪码的基础上, 扩充了模块的定义与调用、数据定义和输入输出而形成的。它是一种用于描述\_\_\_\_\_和\_\_\_\_\_的语言。
- (58) N-S 图的优点是所有的程序结构均用方框来表示, 无论并列或嵌套, 程序的结构清晰可见, 而且它只能表达\_\_\_\_\_的程序逻辑。
- (59) 程序的注释有两种, 它们是\_\_\_\_\_和\_\_\_\_\_。
- (60) 语言是人机通信的工具, 了解程序设计语言的分类可以帮助我们\_\_\_\_\_合适的程序设计语言。

- (1) 在下面的叙述中, ( ) 不是软件需求分析的任务。
  - A. 问题分解
  - B. 可靠性与安全性要求
  - C. 结构化程序设计
  - D. 确定逻辑模型
- (2) 需求分析的结果是产生定义下面 ( ) 问题域的分析模型。
  - A. 信息
  - B. 功能
  - C. 性能
  - D. 以上所有选项
- (3) 需求规格说明书的作用作用不应包括 ( )。
  - A. 软件设计的依据
  - B. 用户与开发人员对软件要做什么的共同理解

- 34

- (12) 软件开发的需求活动，其主要任务是( )。
- A. 给出软件解决方案                      B. 给出系统模块结构
- C. 定义模块算法                          D. 定义需求并建立系统模型
- (13) 在软件开发和维护过程中需要变更需求时，为了保持软件各个配置成分的一致性，必须实施严格的( )。
- A. 产品检验                                  B. 产品控制
- C. 产品标准化                              D. 开发规范
- (14) 软件需求分析阶段的测试手段一般采用( )。
- A. 总结                                          B. 阶段性报告
- C. 定义模块算法                          D. 定义需求并建立系统模型
- (15) 原型化方法是用户和软件开发人员之间进行的一种交互过程，适用于( )系统。
- A. 需求不确定的                          B. 需求确定的
- C. 管理信息                                  D. 决策支持
- (16) 原型化方法是一类动态定义需求的方法，下列描述中，( )不具有原型化方法的特征。
- A. 提高严格定义的文档                  B. 加强用户参与和决策
- C. 简化项目管理                           D. 加速需求的确定
- (17) 准确地解决“软件系统必须做什么”是什么\_\_\_\_\_阶段的任务。
- A. 可行性研究                              B. 需求分析
- C. 软件设计                                  D. 程序编码
- (18) 需求分析的主要过程不包含( )。
- A. 定义系统的边界                      B. 确定需求优先级
- C. 模块设计                                  D. 分析需求可行性
- (19) 原型化方法是软件开发中常用的方法，它与结构化方法相比，更需要( )。
- A. 明确的需求定义                      B. 完整的生命周期
- C. 较长的开发时间                      D. 熟练的开发人员
- (20) 需求分析是( )。
- A. 软件开发工作的开始                  B. 软件生命周期的开始
- C. 由系统分析员单独完成的           D. 由用户自己单独完成的

- (21) 初步用户手册在( )阶段编写。
  - A. 可行性研究
  - B. 需求分析
  - C. 软件概要设计
  - D. 软件详细设计
- (22) 进行需求分析可使用多种工具, 但( )是不适用的。
  - A. 数据流图
  - B. 判定表
  - C. PAD 图
  - D. 数据词典
- (23) 结构化设计方法在软件开发中, 用于( )。
  - A. 测试用例设计
  - B. 软件概要设计
  - C. 程序设计
  - D. 软件详细设计
- (24) 软件结构图中, 模块框之间若有直线连接, 表示他们之间存在着( )关系。
  - A. 调用
  - B. 组成
  - C. 链接
  - D. 顺序执行
- (25) 划分模块时, 一个模块的( )
  - A. 作用范围应在其控制范围之内
  - B. 控制范围应在其作用范围之内
  - C. 作用范围与控制范围互不包含
  - D. 作用范围与控制范围不受任何限制
- (26) 属于软件设计的基本原理是( )
  - A. 数据流分析设计
  - B. 变化流分析设计
  - C. 事物流分析设计
  - D. 模块化
- (27) 软件设计阶段一般又可分为( )
  - A. 逻辑设计与功能设计
  - B. 概要设计与详细设计
  - C. 概念设计与物理设计
  - D. 模型设计与程序设计
- (28) 好的软件结构应该是( )
  - A. 高聚合、高内聚
  - B. 低耦合、高内聚
  - C. 高耦合、低内聚
  - D. 低耦合、低内聚
- (29) 结构图中, 不是其主要成分的是( )
  - A. 模块
  - B. 模块间传递的数据
  - C. 模块内部数据
  - D. 模块的控制关系
- (30) 软件设计一般分为总体设计和详细设计, 它们之间的关系是( )
  - A. 全局和局部
  - B. 抽象和具体
  - C. 总体和层次
  - D. 功能和结构

- (31) 在面向数据流的软件设计方法中，一般将信息流分为( )
- A. 变换流和事物流                      B. 变换流和控制流  
C. 事物流和控制流                      D. 数据流和控制流
- (32) 软件设计中一般将用到图形工具，下列选项中可用作设计的图形工具有( )
- a. 结构图      b. 实体联系图      c. IPO 图      d. 层次图
- A. a 和 b                                      B. c 和 d  
C. a、c 和 d                                  D. 全部
- (33) 软件设计将涉及软件的构造、过程和模块的设计，其中软件过程是指( )
- A. 模块间的关系                          B. 模块的操作细节  
C. 软件层次结构                          D. 软件开发过程
- (34) 模块独立性是软件模块化所提出的要求，衡量模块独立性的度量标准则是模块的( )。
- A. 抽象和信息隐蔽                      B. 局部化和封装化  
C. 内聚性和耦合性                      D. 激活机制和控制方法
- (35) 模块的独立性是由内聚性和耦合性来度量的，其中内聚性是( )
- A. 模块间的联系程度                      B. 模块的功能强度  
C. 信息的隐蔽程度                      D. 接口的复杂程度
- (36) 软件结构是软件模块间关系的表示，下列术语中，( )不属于对模块间关系的描述。
- A. 调用关系                                  B. 从属关系  
C. 嵌套关系                                  D. 主次关系
- (37) 软件设计中划分模块的一个准则是( ① )。两个模块之间的耦合方式中，( ② )耦合的耦合度最高，( ③ )耦合的耦合度最低。一个模块内部的内聚种类中( ④ )内聚的内聚度最高，( ⑤ )内聚的内聚度最低。
- ① A. 低内聚低耦合                      B. 低内聚高耦合  
    C. 高内聚低耦合                      D. 高内聚高耦合
- ② A. 数据                                      B. 非直接  
    C. 控制                                      D. 内容
- ③ A. 数据                                      B. 非直接

- C. 控制  
D. 内容
- ④ A. 偶然  
B. 逻辑  
C. 功能  
D. 过程
- ⑤ A. 偶然  
B. 逻辑  
C. 功能  
D. 过程
- (38) 结构化程序设计主要强调的是( )  
A. 程序的规模  
B. 程序的效率  
C. 程序设计语言的先进性  
D. 程序易读性
- (39) 结构化方法把软件结构划分为( ① )和( ② )两大类，并提出了对应于以上结构的分析方法。  
① A. 控制型  
B. 变换型  
C. 模拟型  
D. 处理型  
② A. 事务型  
B. 分析型  
C. 通信型  
D. 功能型
- (40) 软件结构使用的图形工具，一般采用( )图。  
A. DFD  
B. PAD  
C. SC  
D. ER
- (41) 在软件概要设计中，不使用的图形工具是( )图。  
A. SC  
B. IPO  
C. IDEF0  
D. PAD
- (42) 下列不属于结构化分析的常用工具是( )。  
A. 数据流图  
B. 数据字典  
C. 判定树  
D. PAD 图
- (43) 20 世纪 60 年代后期，由 Dijkstra 提出的、用来增加程序设计的效率和质的方法是( )。  
A. 模块化程序设计  
B. 并行程序设计  
C. 标准化程序设计  
D. 结构化程序设计
- (44) PAD 图的控制执行流程为( )。  
A. 自下而上、从左到右  
B. 自上而下、循环执行  
C. 自上而下、从左到右  
D. 都不对





A. PAD                      B. PDL  
C. IPO                        D. 判定树

A. 需求分析                      B. 概要设计  
C. 详细设计                      D. 编程

A. 遍历  
B. 排序  
C. 循环  
D. 分支

A. 过程、子程序和分程序                      B. 顺序、选择和重复  
C. 递归、堆栈和队列                          D. 调用、返回和转移

A. 不能嵌套使用                      B. 只能用来写简单程序  
C. 已经用硬件实现                    D. 只有一个入口和一个出口

A. FORTRAN  
B. C 语言  
C. Pascal  
D. COBOL

A. 只能使用面向对象语言

B. 可以使用非面向对象语言

C. 可以使用第四代语言

D. 既可以使用面向对象语言，也可以使用非面向对象语言

A. Gantt                      B. C++  
C. Eiffel                     D. Lisp

A. 对象、条件和操作                      B. 具体操作  
C. 对象                                          D. 组合

40

- A. 效率
  - B. 规模
  - C. 完整性
  - D. 容错性
- (64) 在详细设计阶段，经常采用的工具有( )。
- A. PAD
  - B. SA
  - C. SC
  - D. DFD
- (65) 软件详细设计的主要任务是确定每个模块的( )。
- A. 算法和使用的数据结构
  - B. 外部接口
  - C. 功能
  - D. 编程
- (66) PAD 图为( )提供了有力的工具。
- A. 系统分析
  - B. 软件的自动化生成
  - C. 自动分析数据
  - D. 测试软件
- (67) 结构化程序设计的一种基本方法是( )。
- A. 筛选法
  - B. 递归法
  - C. 迭代法
  - D. 逐步求精法
- (68) 结构化程序设计主要强调的是( )。
- A. 程序的效率
  - B. 程序的执行速度
  - C. 程序的易读性
  - D. 程序的规模
- (69) 对于详细设计，以下说法错误的是( )。
- A. 详细设计是具体地编写程序
  - B. 详细设计是细化成容易从中产生程序的结果
  - C. 详细设计的结果基本决定了最终程序的质量
  - D. 详细设计中采用的典型方法是结构化程序设计方法
- (70) PDL 是软件开发过程中用于( )阶段的描述工具。
- A. 需求分析
  - B. 概要设计
  - C. 详细设计
  - D. 编程
- (71) Jackson 图上下层之间的关系是( )。
- A. 调用关系
  - B. 组成关系
  - C. 继承关系
  - D. 嵌套关系
- (72) Jackson 方法是根据( )来导出程序结构的。
- A. 数据流图
  - B. IPO 图

C. 数据间的控制结构

D. 数据结构

### 3.3 判断题

- (1) (            )在需求分析中，分析员要从用户那里解决的最重要的问题是明确软件做什么。
- (2) (            )用例参与者总是人员而不是系统设备。
- (3) (            )在需求确认过程中需求模型被评审以保证其技术可行性。
- (4) (            )需求规格说明书在软件开发中具有重要的作用，它也可以作为软件可行性分析的依据。
- (5) (            )需求跟踪包括编制每个需求与系统元素之间的联系文档。
- (6) (            )需求分析的主要目的是解决系统开发的具体方案。
- (7) (            )在需求分析阶段主要采用图形工具来描述的原因是图形的信息量大，便于描述规模大的软件系统。
- (8) (            )目前存在一个很普遍的现象，即不同的客户提出的需求是相互矛盾的，但每个人都争辩自己是正确的。
- (9) (            )利益相关者是将来购买所开发软件系统的人。
- (10) (            )需求规格说明描述了每个指定系统的实现。
- (11) (            )使用跟踪表有助于识别、控制和跟踪需求的变化。
- (12) (            )开发人员与客户创建用例以帮助软件团队理解有多少类型的最终用户将使用这些功能。
- (13) (            )需求阶段编写文档也包含编写确认测试计划，作为今后确认和验收的依据。
- (14) (            )非功能需求是从各个角度对系统的约束和限制，反映了应用对软件系统质量和特性的额外要求。
- (15) (            )需求分析阶段研究的对象是软件项目的用户需求。
- (16) (            )需求评审人员主要由开发人员组成，一般不包括用户。
- (17) (            )用户需求分析的软件工具，应该能够保证需求的正确性，即验证需求的一致性、完整性、现实性和有效性。
- (18) (            )信息建模方法是从数据的角度来建立信息模型的，最常用的描述信息

模型的方法是 E-R 图。

- (19) ( ) 数据流图建立系统的功能模型，由数据流、数据加工和数据存储组成。
- (20) ( ) HIPO 法既是需求分析方法，又是软件设计方法。
- (21) ( ) 划分模块可以降低软件的复杂度和工作量，所以应该将模块分得越小越好。
- (22) ( ) 在网状结构中任何两个模块都是平等的，没有从属关系，所以在软件开发过程中常常被使用。
- (23) ( ) 信息屏蔽原则有利于提高模块的内聚性。
- (24) ( ) 中心变换型的 DFD 图可看成是对输入数据进行转换而得到输出数据的处理，因此可以使用事务分析技术得到初始的模块结构图。
- (25) ( ) SD 法是一种面向数据结构的设计方法，强调程序结构与问题结构相对应。
- (26) ( ) 所谓结构冲突，是指输入数据与输出数据之间很少或没有结构上的对应关系。通常解决的办法是：构造一个或者多个中间结构，在输入和输出结构之间进行转换。
- (27) ( ) 当模块的控制范围是其作用范围的子集时，模块之间的耦合度较低。
- (28) ( ) SA 法是面向数据流，建立在数据封闭原则上的需求分析方法。
- (29) ( ) 加工小说明是对系统流程图中的加工进行说明。
- (30) ( ) 判定表的优点是容易转换为计算机实现，缺点是不能够描述组合条件。
- (31) ( ) 分层的 DFD 图可以用于可行性分析阶段，描述系统的物理结构。
- (32) ( ) 总体设计的基本目的就是回答：“概括地说，系统应该如何实现？”这个问题。
- (33) ( ) 模块独立要求高耦合低内聚。
- (34) ( ) 数据流图表示了程序系统对数据的算法处理过程，即系统的物理模型。
- (35) ( ) 结构化分析是面向数据流进行需求分析的方法。
- (36) ( ) 信息建模方法是从数据的角度来建立信息模型的，最常用的描述信息模型的方法是 E-R 图。
- (37) ( ) 结构化方法的工作模型是使用螺旋模型进行开发。
- (38) ( ) 结构化方法和 JSP 方法都不适合大型软件的开发。
- (39) ( ) C 语言是一种纯面向对象的语言。

- (40) ( )进行程序设计语言的选择时，首先考虑的是应用领域。
- (41) ( )二义性、简洁性、局部性、顺序性和传统性属于程序设计语言的工程特性。
- (42) ( )提高程序效率的根本途径是选择良好的设计方法、数据结构与算法。
- (43) ( )良好的设计风格就是高的编程技巧。
- (44) ( )项目的应用领域是选择程序设计语言的关键因素。
- (45) ( )FORTRAN、Pascal、C 语言和汇编语言都是科学工程计算可选用的语言。
- (46) ( )SP 方法中的基本控制结构有 GOTO 语句。

### 3.4 简答题

- (1) 什么是需求分析？该阶段的基本任务是什么？如何理解需求分析的重要性和困难性？
- (2) 需求分析方法有哪一种？各自的特点是什么？
- (3) 需求分析应遵循的原则是什么？
- (4) 原型化方法主要用于解决什么问题？
- (5) 需求工程包括哪些基本活动？每一项活动的主要任务是什么？
- (6) 结构化分析方法的步骤是什么？试举例说明“自顶向下、逐步分解”的原理在软件开发中的应用。
- (7) 变换分析设计与事务分析设计有什么区别？简述其设计步骤。
- (8) 什么是软件概要设计？该阶段的基本任务是什么？
- (9) 软件设计的基本原理包括哪些内容？
- (10) 衡量模块独立性的两个标准是什么？各表示什么含义？
- (11) 模块的耦合性有哪一种？各表示什么含义？
- (12) 模块的内聚性有哪一种？各表示什么含义？
- (13) 简述软件结构设计的优化准则。
- (14) 什么是模块的作用范围？什么是软件的控制范围？它们之间应建立什么关系？
- (15) 什么样是“事务流”？什么是“变化流”？试将相应形式的数据流图转换为软件结构图。
- (16) 按照“降低块间联系，提高块内联系”的设计总则对模块进行修改，具体从哪些

方面进行改进？

- (17) 在图 2.1 中，模块 C 中有一个判定对模块 F 产生影响。根据模块的作用范围与控制域的原则，判定图 2.1 (a) 和 (b) 两图的正确性。

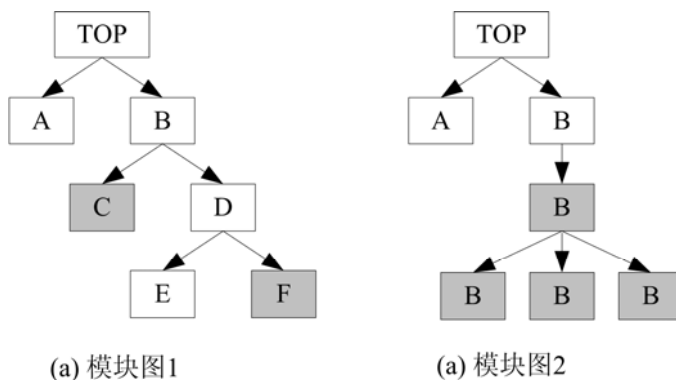


图 2.1 模块结构图

- (18) 图 2.2 中，模块 G 为判定，判断涉及模块 B、F、G，请指出设计中的错误，再根据改进模块图的基本原则，画出一两个改进方案(不改变模块 G 的判断关系)，并说明是按照哪条基本原则进行改进的。

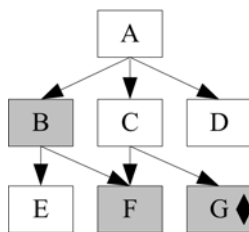


图 2.2 模块结构图

- (19) 为什么要进行程序设计语言的选择？
- (20) 如何运用原型法方法来获取需求？应当注意哪些可能出现的问题？
- (21) 试举例说明“软件复用”在软件开发中的应用。
- (22) 试举例说明“抽象”在软件开发中的应用。
- (23) 试举例说明“信息隐藏”在软件开发中的应用。
- (24) 试举例说明每一种模块间的耦合的例子。

### 3.5 应用题

- (1) 画出“图书馆借阅系统”中图书借阅的数据流图。
- (2) 写出“图书馆借阅系统”中图书、读者、借阅记录的数据字典。
- (3) 写出“图书馆借阅系统”中超期罚款的结构化语言描述、判定表和相应的判定树。
- (4) 画出“图书馆借阅系统”中反映借阅关系的 E-R 图。
- (5) 画出“图书馆借阅系统”中超期罚款的程序流程图、N-S 图和 PAD 图。
- (6) 某高校学生选课系统有如下功能。学生根据开课情况和培养方案填写选课单，选课系统对每个学生的选课单进行处理。选课系统根据教学计划检查学生学分情况，计算上课时间冲突，如果不发生冲突，或冲突小于 20%，则可以选修。根据选课优先级，最后产生每个学生的个人课表和每门课程的选课名单。请画出该系统的顶层和 1 层的数据流图。
- (7) 某单位住房分配方案如下：所有住户 50 平方米以内每平方米 1000 元，超过 20 平方米后，在本人住房标准面积以内每平方米 1500 元，其中住房标准为：教授 105 平方米，副教授 90 平方米，讲师 75 平方米；标准面积以外每平方米 4000 元。请用判定表和判定树表示各条件组合与费用之间的关系。
- (8) 某商场在“五一”期间，针对顾客购物时的收费有 4 种情况：普通顾客一次购物累计少于 100 元，按 A 类标准收费(不打折)，一次购物累计多于或等于 100 元，按 B 类标准收费(打 9 折)，会员顾客一次购物累计少于 1000 元，按 C 类标准收费(打 8 折)，一次购物累计等于或多于 1000 元，按 D 类标准收费(打 7 折)。请用流程图、PAD 图、PDL 来设计收费算法。
- (9) 某校的课酬计算方案如下：基本课酬为每节课 20 元；班级人数超过 60 人，增加基本课酬的 10%；班级人数超过 80 人，增加基本课酬的 20%；如果教师为副教授，增加基本课酬的 15%；如果教师为教授，增加基本课酬的 25%；如果教师为助教，不增加课酬。请用判定表表述上述方案。
- (10) 某程序的描述如下：

```
if ((a>=b&&|>10) || (a<b&&|<=5) )
    k=a;
else
    k=b;
```



- ① 画出单个条件的嵌套的分支结构；
  - ② 画出 N-S 图；
  - ③ 计算该结构的 McCabe 环路复杂性度量。
- (11) 自动柜员机验证顾客身份的工作流程图如下所述：顾客将金融卡插入自动柜员机，自动柜员机读取账户号码，并通过“确认账号”程序，启动账户资料库取得账号资料，进行核对账号的工作，接着，自动柜员机要求顾客输入密码，进入“读取密码”的程序。然后密码资料通过“确认密码”程序，此程序会开启“账户”资料库取得密码资料，进行核对密码的工作，然后传出“正确的密码”资料，再根据顾客要求进行相应操作。画出自动柜员机验证顾客身份部分的工作流程图。

## 第 4 章 面向对象基础

### 4.1 填空题

- (1) 面向对象的基本思想是\_\_\_\_\_。
- (2) 面向对象方法认为系统是由应用域的\_\_\_\_\_组成的。
- (3) 面向对象的方法起源于面向对象程序设计语言，其发展过程大体经历了\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_三个阶段
- (4) 不同应用中信息共享的这种机制和构造是通过\_\_\_\_\_来实现的。
- (5) UML 有三种基本构造块，分别是\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_。
- (6) UML 中的实物分为\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_。
- (7) UML 中有 4 种关系，分别是\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_关系。
- (8) 受限关联由两个类和一个\_\_\_\_\_组成。
- (9) 顺序图和协作图建立了 UML 面向对象开发过程中的对象动态\_\_\_\_\_模型。
- (10) 顺序图的模型元素有\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_等，这些模型元素表示某个用例的若干个对象和对象之间所传递的消息，来对系统的行为建模。
- (11) 配置图由\_\_\_\_\_和\_\_\_\_\_之间的联系组成，描述了处理器、设备和软件构建运行时的体系结构。
- (12) 构件图中的构件没有实例，只有在\_\_\_\_\_中才能标识构件的实例。
- (13) 软件构件分为\_\_\_\_\_构件、\_\_\_\_\_构件和\_\_\_\_\_构件。
- (14) \_\_\_\_\_体现了良好的模块性，它将定义模块和实现模块分开。
- (15) 在用户需求阶段，可以通过\_\_\_\_\_，描述对系统感兴趣的外部角色和它们对系统的功能要求。
- (16) UML 建立的模型也是测试阶段的依据。可使用\_\_\_\_\_进行单元测试，可使用\_\_\_\_\_、\_\_\_\_\_进行集成测试，可使用\_\_\_\_\_进行确认测试，以验证测试结构是否满足用户的需求。
- (17) 如果两个包中的任意两个类之间存在依赖关系，则这两个包之间存在\_\_\_\_\_

关系。

- (18) 活动图描述的要执行的活动以及执行的顺序，但无法说明这些活动由谁来完成，也无法描述每个活动是由哪个类来完成，采用\_\_\_\_\_的方法解决这个问题。
- (19) 在面向对象程序设计语言中，有实现\_\_\_\_\_的机制，可允许每个对象以自己的解释方式来响应共同的消息。

## 4.2 选择题

- (1) 面向对象的开发方法中, ( ) 将是面向对象技术领域内占主导地位的标准建模语言。
  - A. Booch 方法
  - B. UML 语言
  - C. OMT 方法
  - D. Coad 方法
- (2) 一个面向对象系统的体系结构通过它的成分对象和对象间的关系确定, 与传统的面向数据流的结构化开发方法相比, 它具有的有点是( )。
  - A. 设计稳定
  - B. 变换分析
  - C. 事务分析
  - D. 模块独立性
- (3) ( ) 是把对象的属性和操作结合在一起, 构成一个独立的对象, 其内部信息对外界是隐蔽的, 外界只能通过有限的接口与对象发生关系。
  - A. 多态性
  - B. 继承
  - C. 封装
  - D. 消息
- (4) ( ) 意味着一个操作在不同的类中可以有不同的实现方式。
  - A. 多态性
  - B. 多继承
  - C. 消息
  - D. 封装
- (5) UML 是( ) 的缩写。
  - A. Unified Module Language
  - B. Unified Modeling Language
  - C. Universal Module Leveling
  - D. Universal Module Language
- (6) 顺序图反映对象之间发送消息的时间顺序, 它与( ) 是同构的。
  - A. 用例图
  - B. 类图
  - C. 合作图
  - D. 状态图
- (7) ( ) 定义了系统的功能需求, 它是从系统的外部看系统功能, 并不描述系

统内部对功能的具体实现。

- A. 用例图
  - B. 类图
  - C. 活动图
  - D. 状态图
- (8) 状态图包括( )。
- A. 类的状态和状态之间的转换
  - B. 触发类的动作的事件
  - C. 类执行的动作
  - D. 所有以上选项
- (9) 每个对象可用它自己的一组属性和它可以执行的一组( )来表征。
- A. 行为
  - B. 功能
  - C. 操作
  - D. 数据
- (10) 关联是建立在( )之间关系的一种手段。
- A. 类
  - B. 对象
  - C. 角色
  - D. 属性
- (11) 面向对象软件技术的许多强有力的功能和突出的优点，都来源于把类组织成一个层次结构的系统，一个类的上层可以有父亲，下层可以有子类，这种层次结构系统的一个重要性质是( )，一个类获得其父类的全部描述(数据和操作)。
- A. 传递性
  - B. 继承性
  - C. 复用性
  - D. 多态性
- (12) 类库这种机制是( )级别的信息共享。
- A. 同一类
  - B. 不同类
  - C. 同一应用
  - D. 不同应用
- (13) 在软件工程学中，把一组具有相同数据结构和相同操作的对象的集合定义为( )，此定义包括一组数据属性和对数据的一组合法操作。
- A. 类
  - B. 属性
  - C. 对象
  - D. 消息
- (14) 下列选项中不属于面向对象开发方法的有( )。
- A. Booch
  - B. CAD
  - C. Coad
  - D. OMT
- (15) ( )面向客户、建模人员、开发人员和测试人员，是系统模型图的核心。
- A. 状态图
  - B. 类图

- ### 4.3 判断题

- 51

去建立通用软件，多态性和继承性相结合使软件有更广泛的重用性和可扩充性。

- (7) ( ) 继承是父类和子类之间共享数据结构和消息的机制，这是类之间的一种关系。
- (8) ( ) 类图用来表示系统中类和类之间的关系，它是对系统动态结构的描述。
- (9) ( ) 类图中的角色是用于描述该类在关联中所扮演的角色和职责的。
- (10) ( ) 类是指具有相同或相似性质对象的抽象，对象是抽象的类，类的具体化就是对象。
- (11) ( ) UML 建模语言是由视图、图、模型元素和通用机制构成的层次关系来描述的。
- (12) ( ) UML 是一种直观化、明确化、构建和文档化软件系统的通用可视化建模语言。
- (13) ( ) UML 语言支持面向对象的主要概念，并与具体的开发过程相关。
- (14) ( ) 模型是对现实的简化，建模是为了更好地理解所开发的系统。
- (15) ( ) UML 是一种建模语言。
- (16) ( ) 模型是对现实的简化，建模是为了更好理解所开发的系统。
- (17) ( ) 配置图描述系统硬件的物理拓扑结构以及在此结构上执行的软件。
- (18) ( ) 多态性防止了程序相互依赖而带来的变动影响。
- (19) ( ) 类封装比对象封装更具体、更细致。
- (20) ( ) 面向对象的继承性是子类自动共享父类数据结构和方法的机制。

## 4.4 简答题

- (1) 简述活动图和状态图的区别。
- (2) 设计方应和系统的用户进行广泛而深入的交流，明确系统必须具备的主要功能，然后根据这些功能，为参与系统工作的用户分配主要责任等。为了识别出一个系统所设计的执行者，可以向用户提出哪些问题？
- (3) 用例分析技术和结构化分析相比，最大的区别是什么？
- (4) UML 中的使用、扩展和细化三种关系有什么异同，请简述之。
- (5) 简述 UML 实际建模过程。

## 4.5 应用题

- (1) 试用顺序图描述生活中的打电话事件。
- (2) 对于一个电子商务网站而言，以下哪些不是合适的用例？指出并说明理由。  
输入支付信息      将商品放入购物车      结账      预订商品  
用户登录      邮寄商品      查看商品详情
- (3) 面向对象方法相对于传统的结构化方法有什么优势？
- (4) 用你所熟悉的语言来描述教材中图 4.3 中的类。
- (5) 用你所熟悉的语言来编写一段运用了面向对象方法中多态特性的代码。
- (6) 针对聚合与组合，分别举 1 个生活中遇到的例子。
- (7) 常见的 UML 图形有哪些？它们分别有什么作用？
- (8) 什么是模式？
- (9) 在软件开发过程中使用模式有何意义？

## 第 5 章 可行性分析及制定项目计划

### 5.1 填空题

- (1) 可行性研究的目的是：\_\_\_\_\_。
- (2) 可行性分析实质上是进行一次简化、压缩了的\_\_\_\_\_和\_\_\_\_\_过程，要在较高的层次上以比较抽象的方式进行需求分析和设计的过程。
- (3) 可行性研究的三个方面是技术可行性、社会可行性和\_\_\_\_\_。
- (4) 可行性研究的第一个具体步骤是\_\_\_\_\_。
- (5) 若年利率为  $i$ ，不计复利， $P$  元在  $n$  年后的价值  $F$  是\_\_\_\_\_。
- (6) 可行性研究中描述系统高层物理模型的工具是\_\_\_\_\_。
- (7) 可行性研究的目的不是开发一个软件项目，而是研究这个项目是否\_\_\_\_\_。
- (8) 技术可行性一般要考虑的情况包括\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_。
- (9) 效益分析首先是估算将要开发的系统的\_\_\_\_\_，然后与可能取得的效益进行比较\_\_\_\_\_。
- (10) 经济效益可分为有形效益和无形效益两种。有形效益的主要度量指标是\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_。
- (11) 纯收入是指在整个生存周期之内的\_\_\_\_\_与\_\_\_\_\_之差。
- (12) 经济可行性一般要考虑的情况包括\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_。
- (13) 成本效益分析的目的是从\_\_\_\_\_评价一个软件项目是否可行。
- (14) 可行性分析具体步骤的最后一步是\_\_\_\_\_。
- (15) 软件系统可行性分析应考虑\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_和\_\_\_\_\_等方面因素。
- (16) 成本估算方法中，有自顶向下估算方法、自底向上估算方法和\_\_\_\_\_方法。
- (17) 参照以前完成的项目所耗费的总成本，来推算将要开发的软件的总成本，然后把他们按阶段、步骤和工作单元进行分配，这种方法称为\_\_\_\_\_方法。



- (1) 软件质量必须在( )加以保证。
  - A. 开发之前
  - B. 开发之后
  - C. 可行性研究之中
  - D. 分析、设计与实现过程中
- (2) 可行性是系统方案实现的( )。
  - A. 可能性
  - B. 必要性
  - C. 可能性和必要性
  - D. 操作性
- (3) 可行性研究从经济可行性、技术可行性、法律可行性、( )和开发方案可行性等方面进行。
  - A. 资源可行性
  - B. 效益可行性
  - C. 运行可行性
  - D. 政策可行性
- (4) 开发软件时对提高软件开发人员工作效率至关重要的是( )。
  - A. 程序开发环境
  - B. 操作系统的资源管理功能

- 56

- A. 技术性文档
- B. 管理性文档
- C. 需求分析文档
- D. 设计文档

### 5.3 简答题

- (1) 可行性研究主要研究哪些问题？试说明之。
- (2) 可行性研究的步骤是怎样的？
- (3) 可行性研究报告有哪些主要内容？
- (4) 软件开发成本估算方法有哪几种？
- (5) 试分析代码行估算与功能点估算技术的优缺点。
- (6) 什么是软件配置管理？
- (7) 成本-效益分析可用哪些指标进行度量？
- (8) 项目开发计划有哪些内容？

### 5.5 应用题

- (1) 试结合你所参与的某个项目，分析项目中存在的风险，并针对分析结果，制定风险管理措施。
- (2) 设计一个软件的开发成本为 5 万元，寿命为 3 年，未来 3 年的每年收益预计为 22000 元、24000 元、26620 元，银行年利率为 10%。试对此项目进行成本效益分析，以决定其经济性。
- (3) 某学校拟开发一个教学管理系统来减轻部分人工操作，每年可节省 5 万元。若软件可使用 8 年，而开发这个项目需花费 12 万元。试分析该项目的投资回收期。
- (4) 某超市有一个分类存放商品的仓库，仓库中现有的各种商品的数量以及每种商品的库存量临界值等数据记录在库存清单主文件中。当仓库中商品数量有变化时，应及时修改库存清单主文件，如果某种商品的库存量少于它的库存量的临界值，则应该报告给采购部门以便订货，规定每天向采购部门送一次订货报告。画出该系统的流程图并简要描绘其工作情况。
- (5) 某项目开发小组拟开发一软件系统，进度安排如下：可行性分析半个月，需求分析一个月；概要设计可在需求分析进展一半后开始进行，持续二个月；详细设计

在概要设计完成后开始，持续二个月；编码工作在详细设计进展 1 个半月后开始，持续一个月，测试工作与编码工作同时进行，持续一个半月。试绘制该系统进度安排的甘特图。

- (6) 请以无线校园为背景，分析无线校园的技术可行性、操作可行性与经济可行性。
- (7) 请以城市交通流拥堵与控制为背景，书写该系统的可行性分析报告。
- (8) 请以智能家庭为例，分析其经济可行性与技术可行性。
- (9) 请以“信息尘埃”在现代战争中的应用为背景，进行可行性分析。
- (10) 请以网络视频社区为背景，按照国标规范书写可行性分析报告。

## 第 6 章 面向对象分析

### 6.1 填空题

- (1) 面向对象方法描述现实世界建立的三种模型是\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_。
- (2) 面向对象的开发方法是以\_\_\_\_\_作为基本元素构建新系统的方法。它包括面向对象\_\_\_\_\_、面向对象\_\_\_\_\_、面向对象\_\_\_\_\_。
- (3) 系统分析通常是从一个\_\_\_\_\_和用户一系列的讨论开始的。
- (4) 面向对象模型可用 5 个层次表示,即\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_。
- (5) 面向对象分析过程可按照下列流程进行: 确定类-对象、识别结构、识别主题、定义属性、建立\_\_\_\_\_、建立\_\_\_\_\_、定义\_\_\_\_\_。
- (6) 对象模型描述了系统的数据结构,它是一种\_\_\_\_\_结构。动态模型描述了系统的\_\_\_\_\_。功能模型描述了系统的\_\_\_\_\_。
- (7) 在面向对象分析时,在需求文档(陈述)中的名词可作为潜在(候选)的\_\_\_\_\_,形容词可作为线索来确定\_\_\_\_\_,动词可作为潜在的\_\_\_\_\_。关联关系在需求陈述中,使用描述性动词或\_\_\_\_\_来表示。
- (8) 两个或多个对象之间的\_\_\_\_\_、\_\_\_\_\_的关系就是关联。
- (9) 按问题领域确定主题,应该将\_\_\_\_\_的对象确定为同一个主题。
- (10) 每个类的动态行为用一张状态图来描绘,各个类的状态图通过\_\_\_\_\_合并起来,从而构成系统的\_\_\_\_\_。
- (11) 功能模型由一组\_\_\_\_\_组成。它表明了系统中数据之间的依赖关系,以及有关的数据处理功能。
- (12) 在面向对象的编程语言中,\_\_\_\_\_和\_\_\_\_\_是两种不同的语法成分。前者是后者的实例,后者是前者的定义模板。

- (1) 面向对象分析阶段建立的 3 个模型中，核心的模型是( )模型。
- A. 功能 B. 动态
- C. 对象 D. 分析
- (2) 面向对象开发产生的分析文档应当( )考虑问题，在分析阶段识别的概念是高层的概念。
- A. 与问题直接相关 B. 与问题不相关

- C. 在更小的问题范围内                      D. 在更大的问题范围内
- (3) 面向对象技术特别强调的是(            )。
- A. 对象的数据结构                      B. 对象的使用方式
- C. 程序结构                      D. 结构化编程
- (4) 面向对象分析是对系统进行(            )的一种方法。
- A. 需求建模                      B. 程序设计
- C. 设计评审                      D. 测试验收
- (5) 面向对象的分析方法主要是建立 3 类模型，即(            )。
- A. 系统模型、ER 模型、应用模型    B. 对象模型、动态模型、应用模型
- C. E-R 模型、对象模型、功能模型    D. 对象模型、动态模型、功能模型
- (6) 应该在(            )，确定对象类中应有的服务。
- A. 建立对象模型之时                      B. 建立对象模型之后
- C. 建立功能模型之后                      D. 建立对象模型和功能模型之后
- (7) 面向对象的动态模型中，每张状态图表示(            )的动态行为。
- A. 有关联的若干个类                      B. 某一个类
- C. 一系列事件                      D. 一系列状态
- (8) 关联是建立(            )之间关系的一种手段。
- A. 类                      B. 对象
- C. 角色                      D. 属性
- (9) 分析模型一般采用(            )方式进行验证。
- A. 总结                      B. 阶段性报告
- C. 需求分析评审                      D. 转化成设计模型
- (10) 开发人员使用(            )可以将用例的行为分配到所识别的分析类中。
- A. 用例图                      B. 顺序图
- C. 类图                      D. 状态图
- (11) 下列的(            )不属于面向对象分析模型。
- A. 用例图                      B. 类图
- C. 实体关系图                      D. 顺序图
- (12) 下列的(            )不是分析建模的目的。

- A. 定义可验证的软件需求
  - B. 描述客户需求
  - C. 开发一个简单的问题解决方案
  - D. 建立软件设计的基础
- (13) 关于面向对象分析，下列的( )是正确的。
- A. 它是系统需求建模的方法
  - B. 它是分析系统设计的技术
  - C. 可以从分析直接编写代码
  - D. 在软件生命周期中，它出现在面向对象设计之后
- (14) 在考察系统的一些涉及时序和改变的状况时，要用动态模型来表示。动态模型着重系统的控制逻辑，它包括两个图：一个是事件追踪图，另一个是( )。
- A. 数据流图
  - B. 状态图
  - C. 系统结构图
  - D. 时序图
- (15) 动态模型的描述工具是( )。
- A. 对象图
  - B. 结构图
  - C. 状态图
  - D. 设计图
- (16) 有时间间隔的操作是( )。
- A. 动作
  - B. 活动
  - C. 加工
  - D. 处理
- (17) 常用动词或动词词组来表示( )。
- A. 对象
  - B. 类
  - C. 关联
  - D. 属性
- (18) 在确定类时，所有( )都是候选的类。
- A. 名词
  - B. 形容词
  - C. 动词
  - D. 代词
- (19) 表示对象的相互行为的模型是( )模型。
- A. 对象
  - B. 动态
  - C. 功能
  - D. 静态
- (20) 软件开发过程中，抽取和整理用户需求并建立问题域精确模型的过程叫( )。
- A. 生存期
  - B. 面向对象设计



- C. 面向对象程序设计                      D. 面向对象分析
- (21) 在确定属性时，所有(            )是候选的属性。
- A. 动词                                      B. 名词
- C. 修饰性名词词组                      D. 词组
- (22) 对象模型的描述工具是(            )。
- A. 状态图                                      B. 数据流图
- C. 对象图                                      D. 结构图
- (23) 火车是一种陆上交通工具，火车和陆上交通工具的关系是(            )关系。
- A. 组装                                      B. 整体成员
- C. has a                                      D. 一般具体
- (24) 与事件联系在一起的瞬时操作是(            )。
- A. 处理                                      B. 动作
- C. 活动                                      D. 加工
- (25) 一个面向对象的系统的体系结构通过它的(            )的关系确定。
- A. 类与对象                                      B. 成分对象和对象
- C. 过程与对象                                      D. 类与界面
- (26) 功能模型中所有的(            )往往形成一个层次结构，在这个层次结构中一个数据流图的过程可以由下一层数据流图做进一步说明。
- A. 数据流图                                      B. 概念模型图
- C. 状态迁移图                                      D. 事件追踪图

## 6.3 判断题

- (1) (            )面向对象分析的核心在于建立一个描述软件系统的模型。
- (2) (            )系统设计是程序设计的先导和前提条件。
- (3) (            )在数据流程图中，数据流必须通过加工。
- (4) (            )面向对象的分析过程主要包括三项内容：理解、表达和验证。
- (5) (            )面向对象分析，是抽取和整理用户需求并建立问题域精确模型的过程。
- (6) (            )面向对象的方法是以类作为最基本的元素，它是分析问题、解决问题的核心。

- (7) ( )在面向对象程序中，对象彼此间通过继承和多态性启动相应操作。
- (8) ( )在实际的软件开发过程中面向对象分析与面向对象的设计的界限是模糊的。
- (9) ( )面向对象的唯一性是指将具有一致性的数据结构(属性)和行为(操作)的对象抽象成类。
- (10) ( )有些数据流也是对象，把对象看成是单纯的数值和把对象看成是包含许多数值的数据存储这二者是相同的。
- (11) ( )数据流图不表示控制信息，控制信息在其他模型中表示。
- (12) ( )功能模型是类似编译器之类系统的主要模型。
- (13) ( )在面向对象方法中，一个对象请求另一个对象为其服务的方式是通过发送消息。
- (14) ( )从面向对象分析到面向对象设计，是一个逐渐扩充模型的过程。
- (15) ( )面向对象分析与面向对象设计的定义没有明显的区别。
- (16) ( )分类结构可用来描述现实世界中的类的组成与抽象关系。
- (17) ( )面向对象的继承性是子类自动共享父类数据结构和方法的机制。
- (18) ( )数据流图中数据存储用一条直线表示，线段之上标注存储名。
- (19) ( )数据流图中的数据流将对象的输出与处理、处理与对象的输入、处理与处理联系起来。
- (20) ( )描述类中某个对象的行为，反映了状态与事件关系的是对象图。

## 6.4 简答题

- (1) 简述三种面向对象模型的主要功能。
- (2) 面向对象分析模型的用途是什么？
- (3) 为什么将分析模型划分为3类模型？
- (4) 面向对象分析包括哪些活动？应该建立什么模型？
- (5) 常见的需求获取技术有哪些？
- (6) 标识参与者的主要途径有哪些？
- (7) 识别用例的主要途径有哪些？
- (8) 用于需求评审的主要特性有哪些？

## 6.5 应用题

(1) 试编写预订图书的用例描述。

(2) 分析如下案例，并绘制出用例图，撰写需求规格说明书。

ATM 自动柜员机为用户提供了查询、取款、转账等功能，用户在操作之前需插入自己的银行卡，输入正确的银行卡密码。银行工作人员能查询机器状态、形成日报表和月报表。

(3) 某书的组成包括封面、目录、前言、若干章、附录；每章由若干节，每节有若干段，每段有若干句子；每节有 0 个或多个插图，还有 0 个或多个表格。建立该书的对象模型。

(2) 一台微机有一个显示器、一个主机、一个键盘、一个鼠标，汉王笔可有可无。主机包括一个机箱、一个主板、一个电源、存储器等部件。存储器又分为固定存储器、活动存储器，固定存储器又分为内存和硬盘，活动存储器又分为软盘和光盘。建立微机的对象模型。

(3) 窗口分为对话框、图形窗、滚动窗三种；对话框中有若干对话项，由唯一的项名字来确定，对话项分为按钮、选择项、正文项三种，选择项中有若干对话项入口；图形窗中有若干形状元素，形状元素分为一维形状和二维形状，一维形状又分为直线、圆弧、折线；二维形状分为圆、椭圆、矩形、多边形，其中多边形和折线由若干有序项点组成，正文窗是滚动窗的一种，而图形滚动窗既是一种图形窗口又是一种滚动窗。建立窗口系统的对象模型。

(4) 在学校教学管理系统中，学生查询成绩就是系统中的一次交互，请用状态图来描述这种查询的交互行为。

## 第7章 面向对象设计

### 7.1 填空题

- (1) 所谓面向对象设计就是用面向对象观点建立\_\_\_\_\_模型的过程。
- (2) 面向对象方法用\_\_\_\_\_分解取代了传统方法的\_\_\_\_\_分解。
- (3) 面向对象设计中存在的三种内聚为服务内聚、\_\_\_\_\_和一般—特殊内聚。
- (4) 面向对象方法不仅支持过程抽象，而且支持\_\_\_\_\_。
- (5) 设计问题域部分的理由就是为了寻求\_\_\_\_\_。
- (6) 耦合指一个软件结构内\_\_\_\_\_依赖关系。
- (7) 内聚是\_\_\_\_\_的紧密程度。
- (8) 面向对象设计模型同样由主题、类-&-对象、结构、属性和服务等5个层次组成，并且又扩充了\_\_\_\_\_, \_\_\_\_\_、\_\_\_\_\_, \_\_\_\_\_4个部分。
- (9) 客户机和服务器构成两个界限分明的层次，这就是两层\_\_\_\_\_结构。
- (10) 任务管理子系统在设计时，应找出并分析系统中任务的\_\_\_\_\_。
- (11) 设计任务子系统中，常见的任务有\_\_\_\_\_, \_\_\_\_\_、优先任务、关键任务和协调任务等。
- (12) 设计数据管理子系统都包括设计\_\_\_\_\_和设计相应的服务两部分。
- (13) 生命周期方法学把设计进一步划分成总体设计和详细设计两个阶段，类似地，也可以把面向对象设计再细分为\_\_\_\_\_和\_\_\_\_\_。
- (14) 在面向对象开发方法中，将\_\_\_\_\_定义为模块。
- (15) 软件结构内不同模块之间的依赖关系越过，耦合度\_\_\_\_\_。
- (16) 在客户/服务器关系中，任何交互行为都是由\_\_\_\_\_驱动的，因此，作为客户的子系统必须了解作为服务器的子系统的接口，而服务器不必了解客户的接口。
- (17) 在任务管理子系统的设计中，分析并发性的主要依据是系统的\_\_\_\_\_模型。
- (18) 在设计服务时，可以从对象模型中引入服务，也可以从\_\_\_\_\_模型和

模型中确定服务。

- (19) 一个系统体系结构既可以是面向\_\_\_\_\_风格的, 又可以是\_\_\_\_\_风格的。
- (20) 体系结构把分布在不同节点上的系统组成部分之间的关系处理为请求服务和提供服务的关系, 提供服务的计算机称作\_\_\_\_\_, 请求服务的计算机称作\_\_\_\_\_。
- (21) 模型中主动对象的一个主动服务可以用程序中的一个\_\_\_\_\_描述或者用一个\_\_\_\_\_描述来实现。
- (22) 设计人机交互子系统时, 需要把\_\_\_\_\_作为主角。

## 7.2 选择题

- (1) 在面向对象的设计中，我们应遵循的设计准则除了模块化、抽象、低耦合、高内聚以外，还有( )。
- A. 隐藏复杂性                      B. 信息隐藏
- C. 即存类的重用                  D. 类的开发
- (2) ( )是选择合适的解决方案策略，并将系统划分成若干子系统，从而建立整个系统体系结构；对象设计细化原有的分析对象，确定一些新的对象、对每一个子系统接口和类进行准确详细的说明。
- A. 系统设计                      B. 对象设计
- C. 数据库设计                  D. 用户界面设计
- (3) 通过执行对象的操作改变该对象的属性，但它必须通过( )的传递。
- A. 接口                          B. 操作
- C. 信息                          D. 消息
- (4) 面向对象设计阶段的主要任务是系统设计和( )。
- A. 生存期设计                  B. 数据设计
- C. 面向对象程序设计          D. 对象设计
- (5) 一个模块内部的内聚种类中，( )内聚的内聚度最高。
- A. 偶然内聚                      B. 逻辑内聚
- C. 功能内聚                      D. 过程内聚

- (6) 良好设计的特征是( )。
- A. 模块之间呈现高耦合
  - B. 实现分析模型中的所有需求
  - C. 包括所有组件的测试用例
  - D. 模块内聚性弱
- (7) 耦合表示一个模块( )的程度。
- A. 可以被更加细化
  - B. 仅关注在一件事情上
  - C. 能够适时地完成其功能
  - D. 联接其他模块和外部世界
- (8) 两个模块之间的耦合方式中，( )耦合的耦合度最高。
- A. 数据耦合
  - B. 无直接耦合
  - C. 控制耦合
  - D. 内容耦合
- (9) 模块独立性是软件模块化所提出的要求，衡量模块独立性的度量标准是模块的( )。
- A. 抽象和信息隐藏
  - B. 局部化和封装化
  - C. 内聚性和耦合性
  - D. 激活机制和控制方法
- (10) 模块的独立性是由内聚性和耦合性来度量的，其中耦合性是( )。
- A. 模块间的联系程度
  - B. 模块的功能强度
  - C. 信息隐蔽程度
  - D. 接口的复杂程度
- (11) 模块的内聚性可以按照内聚程度的高低进行排序，一下排列中属于从低到高的正确次序是( )。
- A. 偶然内聚，时间内聚，逻辑内聚
  - B. 通信内聚，时间内聚，逻辑内聚
  - C. 逻辑内聚，通信内聚，顺序内聚
  - D. 功能内聚，通信内聚，时间内聚
- (12) 软件设计中划分模块的一个准则是( )。
- A. 低内聚低耦合
  - B. 低内聚高耦合
  - C. 高内聚低耦合
  - D. 高内聚高耦合
- (13) 顺序图和交互图的关系类似于下面( )类型的关系。
- A. 类和对象
  - B. 类和参与者
  - C. Java 和高级程序设计语言
  - D. UML 和 Java
- (14) 在一个课程注册系统中，定义了类 CourseSchedule 和类 Course，并且在类

- CourseSchedule 中定义了方法 `add(c:Course)` 和方法 `remove(c:Course)`，则类 CourseSchedule 和类 Course 之间的关系是( )。
- A. 泛化关系                      B. 组合关系  
C. 依赖关系                      D. 包含关系
- (15) 表达业务流程分析结果的图表是( )。
- A. 业务流程图                      B. 数据流图  
C. 局部 E-R 图                      D. 基本 E-R 图
- (16) 面向对象分析阶段建立的三个模型中，核心的模型是( )模型。
- A. 功能                      B. 动态  
C. 对象                      D. 分析
- (17) 所有的对象可以成为各种对象类，每个对象类都定义了一组( )。
- A. 说明                      B. 方法  
C. 过程                      D. 类型
- (18) 每个对象可用它自己的一组属性和它可以执行的一组( )来表征。
- A. 行为                      B. 功能  
C. 操作                      D. 数据
- (19) 用户界面设计最重要的目标是( )。
- A. 灵活性                      B. 复杂性  
C. 可使用性                      D. 可靠性
- (20) 界面方式可由用户动态制定和修改，这样便可以有较高的( )。
- A. 灵活性                      B. 复杂性  
C. 交互性                      D. 维护性
- (21) 在传统的人机系统中，( )被认为是操作者。
- A. 人                      B. 机器  
C. 人和机器                      D. 操作系统
- (22) 面向对象的模型概念中，类可以有嵌套结构。系统中所有的类组成一个有根的( )。
- A. 有向无环图                      B. 有向有环图  
C. 无向有环图                      D. 无相无环图

- (1) ( ) 用户界面设计对于一个系统的成功是至关重要的, 一个设计得很差的  
用户界面可能导致用户拒绝使用该系统。
- (2) ( ) 关系数据库可以完全支持面向对象的概念, 面向对象设计中的类可以  
直接对应到关系数据库中的表。
- (3) ( ) 面向对象设计是在分析模型的基础上, 运用面向对象技术生成软件实  
现环境下的设计模型。
- (4) ( ) 系统设计的主要任务是细化分析模型, 最终形成系统的设计模型。
- (5) ( ) 面向对象分析和设计活动是一个多次反复迭代的过程。
- (6) ( ) 软件部件的内部实现与外部访问性的分离, 是指软件的封装性。
- (7) ( ) 类库这种机制是统一应用级别的信息共享。
- (8) ( ) 软件设计一般分总体设计和详细设计, 它们之间的关系是全局和局部。
- (9) ( ) 算法的三种基本结构是顺序结构、分支结构、循环结构。
- (10) ( ) 一个设计得好的软件系统具有低内聚、高耦合的特征。
- (11) ( ) 对象模型的描述工具是对象图。
- (12) ( ) 在有多重继承的类层次结构中, 它的类层次结构是网状层次结构。
- (13) ( ) 通过执行对象的操作改变对象的属性, 但它必须通过消息传递。
- (14) ( ) 在面向对象的设计中, 应遵循的设计准则除了模块化、抽象、低耦合、  
高内聚以外, 还有信息隐藏。
- (15) ( ) 数据库系统中, 类是指具有相同的消息, 使用相同的方法, 具有相同  
的变量名和类型。
- (16) ( ) 一个数据库一般不是由独立的对象组成的, 对象的聚集形式的数学意  
义是笛卡尔积。
- (17) ( ) 提供数据库数据描述的集中管理的是数据库管理系统。



- (18) ( )对数据库模式进行规范化处理，是在数据库设计的概念设计阶段。
- (19) ( )对于分布式数据库，可以简单归纳为数据物理上分散，逻辑上统一。
- (20) ( )定义片段以及全局关系与片段之间映像的模式是分片模式。

## 7.4 简答题

- (1) 模块独立性的重要性是什么？
- (2) 面向对象程序设计有哪些优点？
- (3) 面向对象设计与面向对象分析的区别是什么？设计包括哪些活动？
- (4) 面向对象的特征和要素有哪些？
- (5) 类图划分的策略有哪两种？
- (6) 什么是职责分配，GRASP与职责分配之间有什么关系？
- (7) 什么是软件体系结构，常见的软件体系结构有哪些？
- (8) 什么是问题域设计？问题域设计有哪些关键内容？
- (9) 什么是持久化？有哪些常见的持久化策略？
- (10) 用户界面设计包含哪些内容？
- (11) 试述 GUI 界面设计的特点。
- (12) 为什么说 GUI 界面设计是一个迭代的过程？
- (13) GUI 界面设计中需要考虑哪些方面的用户因素？
- (14) 试说明 SDI 界面与 MDI 界面的区别？
- (15) 主要的用户交互形式有哪些？
- (16) 主要的用户信息表示形式有哪些？
- (17) 什么是任务管理设计？

## 7.5 应用题

- (1) 一个软件公司有许多部门，分为开发部门和管理部门两种。每个开发部门开发多个软件产品。每个部门由部门名字唯一确定。该公司有许多员工，员工分为经理、工作人员和开发人员。开发部门有经理和开发人员，管理部门有经理和工作人员。每个开发人员可参加多个开发项目，每个开发项目需要多个开发人员，开发人员

使用语言开发项目。每位经理可主持多个开发项目。建立该公司的对象模型。

- (2) 在温室管理系统中，有一个环境控制器类，当没有种植作物时处于空闲状态。一旦种上作物，就要进行温度控制，定义气候，即在什么时期应达到什么温度。当处于夜晚时，由于温度下降，要调用调节温度过程，以便保持温度；太阳出来时，进入白天状态，由于温度升高，要调用调节温度过程，保持要求的温度。当日落时，进入夜晚状态。当作物收获，终止气候的控制，则进入空闲状态。建立环境控制器类的状态图。
- (3) 请给出以下问题描述的用例模型。有一个学生选课管理系统，系统的执行者有教学管理员、教师和学生；教学管理员使用该系统管理课程信息、教师信息、学生信息以及新学期开设课程信息；教师使用该系统查询新学期开设的课程信息；学生使用该系统查询新学期开设的课程信息和教师开设信息，选课并登记注册课程。
- (4) 参照书中的例子完成一个实际的信息管理系统（如学生成绩管理系统、教务管理系统、办公系统等等）的面向对象设计过程。

## 第 8 章 软件编码与测试

### 8.1 填空题

- (1) 软件测试是为了\_\_\_\_\_而执行程序的过程。
- (2) 被测试程序不在机器上运行，而是采用人工检测和计算机辅助分析检测的手段称为\_\_\_\_\_测试。
- (3) 运行测试中，主要测试软件功能的方法称为\_\_\_\_\_法。
- (4) 选择测试用例，使得被测程序中每个判定的每个分支至少执行一次，这种逻辑覆盖标准称为\_\_\_\_\_。
- (5) 用等价类划分法设计一个测试用例时，使其覆盖\_\_\_\_\_尚未被覆盖的无效等价类。
- (6) 软件测试分 4 步骤：\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_。
- (7) 集成测试分为\_\_\_\_\_测试和\_\_\_\_\_测试。
- (8) 在单元测试中，测试一个模块时，需要设计\_\_\_\_\_和\_\_\_\_\_。
- (9) 软件的集成测试工作最好由\_\_\_\_\_人员承担，以提高集成测试的效果。
- (10) 集成测试也叫做组装测试或联合测试。通常，在\_\_\_\_\_的基础上，将所有模块按照设计要求组装成为系统。
- (11) 确定测试主要由\_\_\_\_\_参加测试，检验软件规格说明的技术标准的符合程度，是保证软件质量的最后关键环节。
- (12) 为了检验程序的正确性，在测试时不仅需要正常数据和异常数据，还需要使用\_\_\_\_\_数据。
- (13) 黑盒测试方法只检查\_\_\_\_\_是否符合要求。
- (14) 确定测试阶段的两项工作是\_\_\_\_\_和\_\_\_\_\_。
- (15) 为了便于对照检查，测试用例应由输入数据和预期的\_\_\_\_\_两部分组成。
- (16) 把软件、硬件和环境联系在一起进行全面的测试称为\_\_\_\_\_测试。
- (17) 面向对象分析的测试和面向对象设计的测试是对分析结果和设计结果的测试，主

要是对\_\_\_\_\_产生的文本进行的，是软件开发前期的关键性测试。

- (18) 对面向对象分析阶段的测试划分为以下五个方面：对认定的对象的测试；对认定\_\_\_\_\_的测试；对认定的主题测试；对定义的属性和实例关联的测试；对定义的\_\_\_\_\_的测试。
- (19) 面向对象编程的测试主要针对\_\_\_\_\_和\_\_\_\_\_进行测试，其主要的测试内容在面向对象单元测试和面向对象集成测试中体现。
- (20) 面向对象集成测试主要是对系统内部的\_\_\_\_\_进行测试，如成员函数间的相互作用，类间的消息传递等。面向对象集成测试不但要基于面向对象单元测试，更要参考面向对象设计或面向对象设计测试的结果。
- (21) 面向对象系统测试是基于面向对象集成测试的最后阶段的测试，主要以\_\_\_\_\_为测试标准，需要借鉴 OOA 或 OOA 测试结果。
- (22) 对面向对象设计的测试，从如下三方面考虑：\_\_\_\_\_的测试、对构造的类层次结构的测试和对类库支持的测试。
- (23) 面向对象的集成测试采用两种新的测试策略：基于线程的测试和\_\_\_\_\_的测试。
- (24) 面向对象的集成测试可以分成两步进行：先进行\_\_\_\_\_测试，再进行\_\_\_\_\_测试。
- (25) 软件测试的目标在于以最少的时间和人力找出软件潜在的各种错误和缺陷；能够确认软件实现的功能和性能与\_\_\_\_\_的一致性；能够收集到足够的测试结果，为软件可靠性评价提供依据。
- (26) 软件测试的任务就是根据软件开发各阶段的文档资料和程序的内部结构，精心设计一组“高产”的\_\_\_\_\_，利用这些用例执行程序，找出软件中潜在的错误和缺陷。

## 8.2 选择题

- (1) 测试的关键问题是( )。
- A. 如何组织软件评审                      B. 如何选择测试用例
- C. 如何验证程序的正确性                D. 如何采用综合策略
- (2) 软件测试的目的是( )。

- A. 证明软件的正确性
  - B. 找出软件中存在的所有错误
  - C. 证明软件系统中存在错误
  - D. 尽可能多地发现软件系统中的错误
- (3) 软件测试用例主要由输入数据和( )两部分组成。
- A. 测试计划
  - B. 测试规则
  - C. 预期输出结果
  - D. 以往测试记录分析
- (4) 下列几种逻辑标准中, 查错能力最强的是( )。
- A. 语句覆盖
  - B. 判定覆盖
  - C. 条件覆盖
  - D. 条件组合覆盖
- (5) 与设计测试用例无关的文档时( )。
- A. 项目开发计划
  - B. 需求规格说明书
  - C. 设计说明书
  - D. 源程序
- (6) 成功的测试是指运行测试用例后( )。
- A. 未发现程序错误
  - B. 发现了程序错误
  - C. 证明程序正确
  - D. 改正了程序错误
- (7) 在黑盒测试中, 着重检查输入条件组合的方法是( )。
- A. 等价类划分法
  - B. 边界值分析法
  - C. 错误推测法
  - D. 因果图法
- (8) 单元测试主要针对模块的几个基本特征进行测试, 该阶段不能完成的测试是( )。
- A. 系统功能
  - B. 局部数据结构
  - C. 重要的执行路径
  - D. 错误处理
- (9) 软件测试中, 白盒法是通过分析程序的( )来设计测试用例的。
- A. 应用范围
  - B. 内部逻辑
  - C. 功能
  - D. 输入数据
- (10) 为了提高软件测试的效率, 应该( )。
- A. 随机地选取测试数据
  - B. 取一切可能的输入数据作为测试数据
  - C. 在完成编码以后指定软件的测试计划
  - D. 选择发现错误可能性较大的数据作为测试用例

- (11) 软件测试过程中的集成测试主要是为了发现( )阶段的错误。
- A. 需求分析 B. 概要分析  
C. 详细分析 D. 编码
- (12) 不属于白盒测试的技术是( )。
- A. 路径覆盖 B. 判定覆盖  
C. 循环覆盖 D. 边界值分析
- (13) 黑盒法是根据程序的( )来设计测试用例的。
- A. 应用范围 B. 内部逻辑  
C. 功能 D. 输入数据
- (14) 集成测试时,能较早发现高层模块接口错误的测试方法为( )。
- A. 自顶向下渐增式测试 B. 自底向上渐增式测试  
C. 非渐增式测试 D. 系统测试
- (15) 确认测试以( )文档作为测试的基础。
- A. 需求规格说明书 B. 设计说明书  
C. 源程序 D. 开发计划
- (16) 使用独立测试团队的最好理由是( )。
- A. 软件开发人员不需要做任何测试  
B. 测试人员在测试开始之前不参与项目  
C. 测试团队将更彻底地测试软件  
D. 开发人员与测试人员之间的争论会减少
- (17) 黑盒测试是从\_\_\_\_\_观点的测试,白盒测试是从\_\_\_\_\_观点的测试。( )
- A. 开发人员、管理人员 B. 用户、管理人员  
C. 用户、开发人员 D. 开发人员、用户
- (18) 软件测试是软件质量保证的主要手段之一,测试的费用已超过( )的 30% 以上,因此提高测试的有效性非常重要。
- A. 软件开发费用 B. 软件维护费用  
C. 软件开发和维护费用 D. 软件研制费用
- (19) 使用白盒测试方法时确定测试数据应根据( )和指定的覆盖标准。
- A. 程序的内部逻辑 B. 程序的复杂程度



- (28) 软件测试的基本原则不包括( )。
- A. 关键是注重选择高效的测试用例 B. 选择尽可能多的测试用例  
C. 尽量不由程序设计者进行测试 D. 充分注意测试中的群集现象
- (29) 增量集成测试是将模块一个一个地连入系统，每连入一个模块( )。
- A. 只需要对新连入的模块进行测试 B. 不需要再进行测试  
C. 要对新子系统进行测试 D. 都要进行回归测试
- (30) 黑盒技术设计测试用例的方法之一为( )。
- A. 因果图 B. 逻辑覆盖  
C. 循环覆盖 D. 基本路径测试
- (31) 按测试数据执行程序逻辑程度判断，条件覆盖( )判定覆盖。
- A. 强于 B. 弱于  
C. 等价于 D. 不一定等价于
- (32) 白盒测试方法重视( )的度量。
- A. 测试覆盖率 B. 测试数据多少  
C. 测试费用 D. 测试周期
- (33) 系统测试是将软件系统与硬件、外设和其他系统元素结合，对整个软件系统进行测试，下述( )不是系统测试的内容。
- A. 强度测试 B. 路径测试  
C. 安装测试 D. 安全测试
- (34) 若有一个计算类型的程序，它的输入量只有一个X，其范围是 $[-1.0, 1.0]$ ，现从输入的角度考虑一组测试用例： $-1.001, -1.0, 1.0, 1.001$ 。设计这组测试用例的方法是( )。
- A. 条件覆盖法 B. 等价分类法  
C. 边界值分析法 D. 错误推测法
- (35) 黑盒测试在设计测试用例时，主要需要研究( )。
- A. 需求规格说明与概要设计说明 B. 详细设计说明  
C. 项目开发计划 D. 概要设计说明和详细设计说明
- (36) 软件测试方法中的静态测试方法之一为( )。
- A. 计算机辅助静态分析 B. 黑盒法



- C. 路径覆盖 D. 边界值分析
- (37) 等价分类法的关键是( )。
- A. 确定等价类的边界条件 B. 按照用例来确定等价类
- C. 划分等价类 D. 确定系统中相同和不同的部分
- (38) 程序测试是一项复杂的工作，一般测试者在进行测试中都需要设计( )。
- A. 数据结构 B. 测试实例
- C. 测试阶段 D. 测试方法

### 8.3 判断题

等价类。

- (14) ( ) 一个好的测试用例在于能发现至今未发现的错误。
- (15) ( ) 可以用穷举测试的方法对程序进行正确证明。
- (16) ( ) 测试只能证明程序有错误，不能证明程序没有错误。
- (17) ( ) 在程序调试时，找出错误的位置和性质比改正改错误更难。
- (18) ( ) 有效性测试是在模拟环境(或开发环境)下运用黑盒法，验证软件特性是否与需求符合。
- (19) ( ) 发现错误多的模块，残留在模块中的错误也多。
- (20) ( ) 边界值分析方法是取输入/输出等价类的边界值作为测试用例。
- (21) ( ) 安全测试是检测系统中的保护机制是否可以保护系统免受非正常的攻击。
- (22) ( ) 压力测试是检测在极限环境中使用系统时施加在用户上的压力。
- (23) ( ) 功能测试是根据软件需求规格说明和测试需求列表，验证产品的功能实现是否符合需求规格。

## 8.4 简答题

- (1) 软件编码规范的意义何在？
- (2) 软件测试的目的是什么？软件测试中，应注意哪些原则？
- (3) 软件测试要经过哪些步骤？这些测试与软件开发各阶段之间有什么关系？
- (4) 什么是白盒测试法？有哪些覆盖标准？试对它们的检错能力进行比较？
- (5) 什么是黑盒测试法？采用黑盒技术测试构造测试用例有那几种方法？这些方法各有什么特点？
- (6) 单元测试、集成测试和确认测试各自的主要目标是什么？它们之间有什么不同？相互间有什么关系？
- (7) 什么是集成测试？非增量测试和增量测试有什么区别？增量测试如何组装模块？
- (8) 简要说明面向对象的测试模型。
- (9) 什么是验证和确认？

## 8.5 应用题

- (1) 某程序的功能是输入代表三角形三条边长的三个整数,判断他们能否组成三角形,若能则输出等边、等腰或任意三角形的类型标记。请分别用黑盒法和白盒法对该程序设计测试用例。
- (2) 变量的命名规则一般规定如下:变量名的长度不多于 40 个字符,第一个字符必须为英文字母,其他字母可以是英文字母、数字以及下划线的任意组合。请用等价分类法设计测试用例。

## 第9章 软件维护

### 9.1 填空题

- (1) 所有软件维护申请要按规定方式提出，该报告也称\_\_\_\_\_报告。
- (2) 有两类维护技术：在开发阶段使用来减少错误，提高软件可维护性的面向维护技术；在维护阶段用来提高维护的效率和质量的\_\_\_\_\_技术。
- (3) 在软件维护中，因修改软件而导致出现的错误或其他情况称为\_\_\_\_\_。
- (4) 由于软件运行的外部环境(如软件、硬件)和数据环境等发生了变化而修改软件，使之适应这些变化的活动。这类维护称为\_\_\_\_\_。
- (5) \_\_\_\_\_维护是指在软件使用过程中，用户会对软件提出新的功能和性能要求，为了满足这些新的要求而对软件进行修改，使之在功能和性能上得到完善和增强的活动。
- (6) 预防性维护是为了改进未来的可维护性，或者为了给未来的改进奠定更好的基础，而修改软件的活动，这类活动包括\_\_\_\_\_和\_\_\_\_\_。
- (7) 软件维护活动包括一个标准化的事件序列。其中有建立维护组织、完成维护报告、维护过程、维护记录以及\_\_\_\_\_。
- (8) \_\_\_\_\_是指软件被理解、改正、调整和改进的容易程度。
- (9) 软件可维护性与软件质量和可靠性一样是难以量化的概念，但可以借助维护活动中可定量估算的属性\_\_\_\_\_度量可维护性。
- (10) \_\_\_\_\_指因修改软件的信息结构而带来的不良后果。
- (11) 维护应统一考虑整个软件配置，而不仅仅是源代码；否则，由于在设计文档和用户手册中未能反映修改情况而引起\_\_\_\_\_。
- (12) \_\_\_\_\_是目前预防性维护所采用的主要技术。
- (13) 软件维护的副作用有\_\_\_\_\_副作用、\_\_\_\_\_副作用、\_\_\_\_\_副作用三种。
- (14) 必须在软件交付之前对整个\_\_\_\_\_进行\_\_\_\_\_，以减少文档副作用。

- (15) 维护阶段是软件生存周期当中\_\_\_\_\_的阶段，也是花费精力和费用\_\_\_\_\_的阶段。
- (16) 软件的\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_是衡量软件质量的几个主要特性。
- (17) 软件维护的费用增加的主要原因是维护的\_\_\_\_\_非常低。
- (18) \_\_\_\_\_是一种由用户产生的文档，它是计划维护任务的基础、
- (19) 采用手工方法开发软件只有程序而无文档，维护困难，这是一种\_\_\_\_\_维护，采用\_\_\_\_\_方法开发软件，各阶段均有文档，容易维护，这是一种\_\_\_\_\_维护。
- (20) 软件交付使用后，由于软件开发过程中产生的\_\_\_\_\_没有完全彻底在\_\_\_\_\_阶段发现，必然有一部分隐含错误带到\_\_\_\_\_阶段。

## 9.2 选择题

- (1) 在软件维护的内容中，占维护活动工作量比例最高的是( )。
- A. 校正性维护                      B. 适应性维护
- C. 完善性维护                      D. 预防性维护
- (2) 当前时期软件维护的费用在总费用中的比例为( )。
- A. 35%~40%                      B. 40%~60%
- C. 60%~70%                      D. 70%~80%或更高
- (3) 使用软件时提出增加新功能就必须进行( )维护。
- A. 改正性                          B. 适应性
- C. 完善性                          D. 预防性
- (4) 维护软件时为了保证修改不会影响原来可以正常使用的功能，应安排( )测试。
- A. 单元                              B. 集成
- C. 验收                              D. 回归
- (5) 维护过程本质上是( )的过程。
- A. 修改程序                      B. 修改文档
- C. 修改软件配置                  D. 修改和压缩软件定义和开发



正向工程 6 类活动。

- A. 功能性
  - B. 可靠性
  - C. 可使用性
  - D. 可维护性
- (14) 软件生存期中的( )的工作都与软件可维护性有密切的关系。
- A. 编码阶段
  - B. 设计阶段
  - C. 测试阶段
  - D. 每个阶段
- (15) 软件可维护性是指纠正软件系统出现的错误和缺陷,以及为满足新的要求进行修改、( )的容易程度。
- A. 维护
  - B. 扩充与压缩
  - C. 调整
  - D. 再工程
- (16) 维护中,因误删除一个标识符而引起的错误是( )副作用。
- A. 文档
  - B. 数据
  - C. 编码
  - D. 设计
- (17) 目前广泛使用 7 个特性来衡量软件的可维护性,其中可测试性、可靠性、可理解性和可修改性主要在( )过程中侧重应用。
- A. 校正性维护
  - B. 适应性维护
  - C. 完善性维护
  - D. 预防性维护
- (18) McCall 提出了表明软件质量的 11 个质量特性。它们是正确性、可靠性、可使用性、完整性、( )、可移植性、可复用性、效率、可测试性、互联性。
- A. 模块独立性、安全性
  - B. 文档完备性、灵活性
  - C. 可维护性、安全性
  - D. 可维护性、灵活性
- (19) McCall 提出的软件质量的 11 个质量特性分为三组,属于产品修正特性的有( )。
- A. 可移植性、可复用性、互联性
  - B. 可维护性、可移植性、可复用性
  - C. 可维护性、可测试性、灵活性
  - D. 正确性、可使用性、可测试性
- (20) 软件从一个计算机系统或环境转移到另一个计算机系统或环境的容易程度是指软件的( )。
- A. 兼容性
  - B. 可移植性
  - C. 容错性
  - D. 可接近性





- A. 校正性维护
- B. 适应性维护
- C. 完善性维护
- D. 预防性维护

### 9.3 判断题

- (1) (            )当通过验收测试，软件开发就完成了。
- (2) (            )文档是影响软件可维护性的决定因素。
- (3) (            )适应性维护是在软件使用过程中，用户会对软件提出新的功能和性能要求，为了满足这些新的要求而对软件进行修改，使之在功能和性能上得到完善和增强的活动。
- (4) (            )进行软件维护活动时，直接修改程序，无需修改算法。
- (5) (            )软件生命周期最后一个阶段是书写软件文档。
- (6) (            )软件生存周期中，工作量所占比例最大的阶段是维护阶段。
- (7) (            )软件生存周期中，只有编码阶段的工作和软件可维护性有密切关系。
- (8) (            )软件维护中，因修改全局或共用数据引起的错误称为编码副作用。
- (9) (            )软件维护中大部分工作是由于用户的需求改变而引起的。
- (10) (            )需求分析的修改不属于软件维护中的副作用。

### 9.4 简答题

- (1) 什么是软件维护的副作用？
- (2) 什么是软件维护？维护的目的是什么？软件维护有哪几种？
- (3) 软件维护有哪三种类型的维护活动？
- (4) 软件维护的特点是什么？
- (5) 软件维护的流程是什么？
- (6) 什么是软件可维护性？可维护性度量的特性是什么？
- (7) 提高可维护性的方法有哪些？
- (8) 可维护性度量的质量特性有哪些？
- (9) 什么是程序修改的副作用？
- (10) 程序修改的副作用有哪几种？试举例说明。

- (11) 维护技术有哪些？
- (12) 影响软件维护成本的因素有哪些？
- (13) 软件正向工程与再工程的区别是什么

## 第 10 章 软件项目管理与质量保证

### 10.1 填空题

- (1) 在软件项目管理中，控制包括\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_和\_\_\_\_\_。
- (2) 能协调软件开发，使得混乱减少到最小的方法是使用\_\_\_\_\_。
- (3) 根据软件工程标准制定的机构与适用范围，它分为\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_和\_\_\_\_\_ 5 个等级。
- (4) 行业标准是由行业机构、学术团体或国防机构制定的适合某个行业的标准。IEEE 指\_\_\_\_\_。
- (5) 软件工程的分解是从\_\_\_\_\_和\_\_\_\_\_，即\_\_\_\_\_和\_\_\_\_\_两个方面进行的。
- (6) 在软件项目管理过程中一个关键的活动是\_\_\_\_\_，它是软件开发工作的第一步。
- (7) 软件配置管理是一种\_\_\_\_\_、\_\_\_\_\_和\_\_\_\_\_的技术。
- (8) 软件质量保证是建立一套有计划，有系统的方法，来向管理层保证拟定出的\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_和\_\_\_\_\_能够正确地被所有项目所采用。
- (9) 软件配置管理活动的目标就是为了\_\_\_\_\_、\_\_\_\_\_确保变更正确实现并向其他有关人员报告变更。
- (10) 软件架构是一系列相关的\_\_\_\_\_，用于指导大型软件系统各个方面的设计。
- (11) 在软件的生产过程中，总有大量的各种信息要记录，因此，\_\_\_\_\_在产品的开发过程中起着重要的作用。

## 10.2 判断题

- (1) (            ) 软件项目管理是软件生命周期一切活动的管理。
- (2) (            ) 软件工程学中除重视软件开发技术的研究外，另一个重要组成内容是软件的工程管理。
- (3) (            ) 软件配置管理简称 SCI。
- (4) (            ) 一个项目是否开发，从经济上来说是否可行，归根结底是取决于项目计划。
- (5) (            ) 软件开发规范的体现和指南是文档。
- (6) (            ) 在变更控制中，存取控制管理各个用户存取和修改一个软件配置对象的权限。
- (7) (            ) 用户文档是在软件开发过程中，作为软件开发人员前一阶段工作成果的体现和后一个阶段工作依据的文档。
- (8) (            ) 软件开发在需求分析、设计、测试等几个阶段所需不同层次的技术人员大致是中级、中级、高级、中级。
- (9) (            ) 版本控制应该考虑系统的具体版本进行描述和生产。
- (10) (            ) 在软件项目管理过程中一个关键的活动是编写需求说明书，它是软件开发工作的第一步。
- (11) (            ) 软件项目计划是由程序员与用户单位共同经过“可行性研究与计划”阶段后制定的。
- (12) (            ) GB 是指中华人民共和国国家军用标准。
- (13) (            ) 文档仅仅描述和规定了软件的使用范围及相关的操作命令。
- (14) (            ) 软件质量保证的目的是使软件过程对于管理人员来说是不可见的。
- (15) (            ) 计算机软件的著作权属于软件开发者。
- (16) (            ) 开发文档是在软件开发过程中，由软件开发人员制定的需提交管理人员的一些工作计划或工作报告。
- (17) (            ) 需求工程是一个不断反复的需求定义、文档记录、需求演进的过程，并最终在验证的基础上冻结需求。
- (18) (            ) 软件配置管理的目的是使错误降为最小并最有效地提高生产效率。

- (19) (            ) 软件质量保证工作是没有计划的。
- (20) (            ) 对象-关系映射是随着面向对象的软件开发方法发展而产生的。面向对象的开发方法是当今企业级应用开发环境中的主流开发方法，关系数据库是企业级应用环境中永久存放数据的主流数据存储系统。

### 10.3 简答题

- (1) 软件人员组织方式通常有几种？分别有什么特点？
- (2) 什么是软件配置项？什么是基线？什么是版本？
- (3) 软件质量保证体系主要包括哪些内容？
- (4) 软件可靠性包含哪些要素，具体内容有哪些？
- (5) CMM 是什么？CMMI 是什么？CMMI 与 CMM 有何关系？
- (6) CMMI 分为几个层次？

## 参考文献

- [1] (美)Frederick P. Brooks, Jr. 李琦注. 人月神话[M], 北京: 人民邮电出版社, 2007. 10
- [2] (美)Frederick P. Brooks, Jr. 王海鹏, 高博 译. 设计原本: 计算机科学巨匠 Frederick P. Brooks 的思考[M], 北京: 机械工业出版社, 2011. 01
- [3] 孙家广, 刘强. 软件工程—理论、方法与实践[M]. 北京: 高等教育出版社, 2010. 11
- [4] 齐治昌, 谭庆平, 宁洪. 软件工程(第 3 版) [M], 北京: 高等教育出版社, 2012. 05
- [5] 许家珩. 软件工程: 方法与实践(第 2 版) [M], 北京: 机械工业出版社, 2011. 12
- [6] 刘伟. 设计模式[M], 北京: 清华大学出版社, 2011. 10
- [7] 张海藩. 软件工程导论(第 5 版) [M]. 北京: 清华大学出版社, 2012. 05
- [8] (美)Stephen R. Schach. 邓迎春, 韩松 译. 软件工程: 面向对象和传统的方法 [M]. 北京: 机械工业出版社, 2012. 01
- [9] 殷人昆, 郑人杰等. 实用软件工程(第 3 版) [M], 北京: 清华大学出版社, 2010. 11
- [10] 刘超, 张莉. 可视化面向对象建模技术—标准建模语言 UML [M], 北京: 北京航空航天大学出版社, 2001. 03
- [11] (美)Roger S. Pressman. 郑人杰, 马素霞, 白晓颖 译. 软件工程: 实践者的研究方法(第 6 版) [M]. 北京: 机械工业出版社, 2007. 01
- [12] Jason Baragry. Understanding Software Engineering: From Analogies with other Disciplines to Philosophical Foundations [PhD thesis]. Victoria: La Trobe University, 2000.
- [13] 中国标准出版社. 计算机软件工程规范国家标准汇编(第二版) [M]. 中国标准出版社, 2011. 08