**《软件工程》期末总复习题（新版）**

**第一套试卷**

**一、选择题：** （软件工程作业1 第1～4章）

1. 开发软件所需高成本和产品的低质量之间有着尖锐的矛盾，这种现象称做（C）。

A.软件工程 B.软件周期 C.软件危机 D.软件产生

1. 瀑布模型本质上是一种（A ）模型。

A.线性顺序 B.顺序迭代 C.线性迭代 D.早期产品

1. 瀑布模型存在的问题是（B ）。

A．用户容易参与开发 B．缺乏灵活性C．用户与开发者易沟通 D．适用可变需求

1. 螺旋模型是一种将瀑布模型和（A）结合起来的软件开发模型。

A．增量模型 B．专家系统 C．喷泉模型 D．变换模型

1. 原型化方法是用户和设计者之间执行的一种交互构成，适用于（A）系统。

A．需求不确定性高的 B．需求确定的 C．管理信息 D．实时

1. 下列有关软件工程的标准，属于国际标准的是(D )

A.GB B.DIN C.ISO D.IEEE

1. 结构化方法是一种基于（D）的方法。

A.数据结构 B.程序结构 C.算法 D.数据流

1. 软件可行性研究实质上是要进行一次( A)需求分析、设计过程。

A、简化、压缩的 B、详细的 C、彻底的 D、深入的

1. 可行性研究的目的是（D）

A、分析开发系统的必要性 B、确定系统建设的方案

C、分析系统风险 D、确定是否值得开发系统

1. 设年利率为i，现存入p元，不计复利，n年后可得钱数为（C）。

A、P×(1+i\*n) B、P× (i+1)\*n C、p×(1+i)n D、p×(i+n)

1. 可行性研究是在（A）之前

A、系统开发 B、测试 C、试运行 D、集成测试

1. 可行性研究需要的时间长短取决于系统的规模，一般来说，可行性研究的成本只是预期总成本的（C）。

A、1% B、20% C、8% D、50%

1. 我国正式颁布实施的（B）对计算机病毒的定义具有法律性、权威性。

A．《计算机软件保护条例》

B．《中华人民共和国计算机信息系统安全保护条例》

C．《中华人民共和国著作权法》

D．《计算机病毒防治管理办法》

1. 效率是一个性能要求，因此应当在以下哪个阶段规定？B

A、可行性分析B、需求分析C、概要设计D、详细设计

1. 需求规格说明书的作用不包括（D）

A、软件验收的依据 B、用户与开发人员对软件要做什么的共同理解

C、软件可行性研究的依据 D、软件设计的依据

1. 数据字典是用来定义（D）中的各个成份的具体含义的。

A、程序流程图B、功能结构图 C、系统结构图D、数据流程图

1. 数据流程图是（A）方法中用于表示系统逻辑模型的一种图形工具。

A、SA B、SD C、SP D、SC

1. DFD中的每个加工至少有(A)。

A、一个输入流或一个输出流 B、一个输入流和一个输出流

C、一个输入流 D、一个输出流

1. 需求分析阶段的关键任务是确定（D）

A、软件开发方法 B、软件开发工具C、软件开发费D、软件系统的功能

1. 在SD方法中全面指导模块划分的最重要的原则是（D）。

A、程序模块化B、模块高内聚C、模块低耦合D、模块独立性

1. 在模块的3种基本属性中，反映模块内部特性的是（C）。

A、接口 B、功能 C、逻辑 D、状态

1. 模块的耦合性可以按照耦合程度的高低进行排序，以下哪一项符合从低到高的正确次序（D）。

A、数据结构耦合，公共耦合，控制耦合，内容耦合

B、数据耦合，控制耦合，数据结构耦合，公共耦合

C、无直接耦合，数据结构耦合，内容耦合，控制耦合

D、无直接耦合，数据耦合，控制耦合，内容耦合

1. PDL是描述处理过程( C )

A、做什么B、为什么做 C、怎么做D、对谁做

1. （D）工具在软件详细设计过程中不采用。

A、判定表 B、IPO图 C、PDL D、DFD图

1. 为高质量地开发软件项目，在软件结构设计时，必须遵循（A）原则。

A.信息隐蔽B、质量控制 C、程序优化D、数据共享

1. 当一个模块直接使用另一个模块的内部数据，这种模块之间的耦合为（D）

A、数据耦合B、公共耦合C、标记耦合D、内容耦合

1. 详细设计与概要设计衔接的图形工具是(B)。

A、DFD图 B、SC图 C、PAD图 D、程序流程图

1. 下列几种类型中，耦合性最弱的是(D)。

A、内容型B、控制型C、公共型D、数据型

1. 软件结构使用的图形工具，一般采用（C）图。

A、DFD B、PAD C、SC D、ER

1. 在七种偶合中，最低偶合是（D）。
2. 内容偶合B、公共偶合C、数据偶合D、非直接偶合

**二、简答题**  （软件工程作业1 第1～4章）

1. 什么是软件工程？软件工程的目标是什么？

答：软件工程是①将系统化的、规范的、可度量的方法应用于软件的开发、运行和维护过程，即将工程化应用于软件开发和管理之中，②对①中所选方法的研究。

目标：

1. 合理预算开发成本，付出较低的开发费用。
2. 实现预期的软件功能，达到较好的软件性能，满足用户的需求。

3 提高软件的可维护性，降低维护费用。

4 提高软件开发生产率，及时交付使用

5 增强软件过程的可见性和可控性，保证软件的质量。

1. 指出瀑布模型中下列任务的顺序：验收测试，项目计划，单元测试，需求评审，成本估计，概要设计，详细设计，系统测试，设计评审，编码，需求规格说明书。

答：项目计划，成本估计，需求规格说明书，需求评审，概要设计，详细设计，设计评审，编码，单元测试，系统测试，验收测试

1. 可行性研究报告的主要内容是什么？

答：可行性研究主要包括以下几个部分：

（1）概述（2）技术可行性（3）项目成熟程度（4）市场需求情况和风险分析（5）投资估算及资金筹措（6）经济与社会效益（7）综合实力和产业基础（8）项目实施进度计划（9）其它尚需要说明的必要的方面（10）结论

1. 什么是需求分析？需求分析阶段的基本任务是什么？

答：所谓"需求分析"，是指对要解决的问题进行详细的分析，弄清楚问题的要求，包括需要输入什么数据，要得到什么结果，最后应输出什么。可以说，“需求分析”就是确定要计算机“做什么”。

任务：确定软件项目的目标和范围。调查使用者的要求，分析软件必须做什么，编写需求规格说明书等它相关文档，并进行必要的需求审查。除此之外，还包括需求变更控制，需求风险控制，需求版本控制等对需求的管理工作

1. 什么是结构化分析方法？该方法使用什么描述工具？

答：结构化的分析方法是面向数据流的方法，因此，此方法研究的核心是数据的组成和数据流向和对数据的加工处理。结构化分析方法用抽象模型的概念，按照软件内部数据传递、变换的关系，自顶向下逐层分解，直至找到满足功能要求的所有可实现的软件元素为止

描述工具：1系统流程图 2 数据流程图 3数据字典 4 IPO图 5层次方框图 6实体—关系图 7状态—变迁图

1. 结构化分析方法通过哪些步骤来实现?

答：1）    确定系统的功能要求

2)     确定系统的数据要求

3)     确定系统的操作要求和界面要求

4)     确定系统的性能要求

5)     确定系统的运行要求

6)     获得当前系统的物理模型

7)     抽象出当前系统的逻辑模型

8)     建立目标系统的逻辑模型。

9)     修正开发计划

10） 如果需要则开发系统原型

1. 什么是数据流图?其作用是什么?其中的基本符号各表示什么含义?

答：数据流图：简称DFD，就是采用图形方式来表达系统的逻辑功能、数据在系统内部的逻辑流向和逻辑变换过程，是结构化系统分析方法的主要表达工具及用于表示软件模型的一种图示方法。

　　数据流图的基本符号的意思：

　　1.矩形表示数据的外部实体；

　　2.圆角的矩形表示变换数据的处理逻辑；

　　3.少右面的边矩形表示数据的存储；

4.箭头表示数据流。

1. 什么是数据字典？其作用是什么？它有哪些条目？

答：数据字典，主要用来描述数据流程图中的数据流、数据存储、处理过程和和数据源点/终点。

作用：数据流程图描述了系统的逻辑结构，其中的四个基本图形元素的含义无法在数据流程图中详细说明，因此数据流程图需要与其他工具配合使用，数据字典就是这样的工具之一。

包括的条目：

数据流词条

数据元素词条

数据存储词条

数据加工处理词条

数据源点及终点词条

1. 什么是概要设计？有哪些基本任务？

答：概要设计：根据用户的需求先确定软件系统的总体结构和总的设计原则

基本的任务：

设想供选择的方案。

推荐最佳方案。

功能分解，确定软件结构。

设计软件结构。

制定测试计划。

数据设计。

书写文档。包括总体设计规格说明书。用户手册。测试计划。

设计审查和复审。

1. 详细设计的基本任务是什么？有那几种描述方法？

答：详细设计阶段的任务是要为编写程序代码设计“图纸”，由程序员按“图纸”用某种高级程序设计语言编写程序代码。主要方法有设计程序流程图，表格设计符号以及过程设计语言。

1. 什么是变换流？什么是事物流？

答：变换型数据流可以划分为明显的三部分：逻辑输入，中心变换，逻辑输出

事务流的特点是数据沿某个输入路径流动，该路径将外部信息转换成事务，其中发射出多条事务处理路径的中心处理被称为中心事务

1. 模块的内聚有哪几种？模块间的耦合有哪几种？

答：内聚分为七种类型：

巧合内聚——一个模块执行多个完全互不相关的动作，那么这个模块就有巧合内聚

逻辑内聚——当一个模块执行一系列相关的动作时，称其有逻辑内聚。

时间性内聚——当一个模块内的多个任务是与时间有关时，这个模块具有时间性内聚。

过程内聚——模块执行的若干动作之间有顺序关系。

通信内聚——模块执行的若干动作之间有顺序关系，并且所有动作是在相同的数据上执行。

信息内聚——一个模块中执行一系列动作，每个动作都有自己的入口点和处理代码，所有的动作都作用在相同的数据结构上，这样的模块叫做信息内聚。

功能性内聚——一个模块中各个部分都是完成某一具体功能必不可少的组成部分

耦合分为七类：

内容耦合——如果一个模块直接参考另一个模块的内容，则这两个模块是内容耦合。

公共耦合——如果多个模块都访问同一个公共数据环境，则称它们是公共耦合。

外部耦合——如果两个模块都访问同一个全局简单变量而不是同一全局数据结构，而且，不是通过参数表传递该全局变量的信息，则这两个模块属于外部耦合。

控制耦合——如果模块A向模块B传递一个控制信息，则称这两个模块是控制耦合的

数据结构耦合——当一个模块调用另一个模块时传递了整个数据结构，这两个模块之间具有数据结构耦合。

数据耦合——如果两个模块传递的是数据项，则这两个模块是数据耦合。

非直接耦合——如果两个模块之间没有直接关系，它们之间的联系完全通过主模块的控制和调用来实现的，这就是非直接耦合。

三、应用题 （软件工程作业1 第1～4章）

1．某旅馆的电话服务如下：可以拨分机号和外线号码。分机号是从7201至7299。外线号码先拨9，然后是市话号码或长话号码。长话号码是以区号和市话号码组成。区号是从100到300中任意的数字串。市话号码是以局号和分局号组成。局号可以是455，466，888，552中任意一个号码。分局号是任意长度为4的数字串。写出在数据字典中，电话号码的数据条目的定义(即组成)。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 简称 | 类型 | 长度 | 取值范围 |
| 分机号 | Extension\_Number | 字符 | 4 | 7201..7299 |
| 外线市话号 | Addition\_Internumber | 字符 | 9 | 9+[455|466|888|552]+4{数字}4 |
| 外线长话号 | Addition\_Extennumber | 字符 | 12 | 9+[100|101|102|..|298|299|300]+[455|466|888|552]+4{数字}4 |

电话号码={分机|外线}

分机=[7201——7299]

外线=9+{市话号码|长话号码}

市话号码=局号+分局号

局号={455，466，888，552}

分局号=4{0-9}4

长话号码=区号+市话号码

区号=[100—300]

2．下面是旅客订飞机票的需求描述，试画出分层的数据流程图。

顾客将订票单交给预订系统：（1）如果是不合法订票单，则输出无效订票信息；（2）对合法订票单的预付款登录到一个记账文件中；（3）系统有航班目录文件，根据填写的旅行时间和目的地为顾客安排航班；（4）在获得正确航班信息和确认已交了部分预付款时发出取票单，并记录到取票单文件中。

顾客在指定日期内用取票单换取机票：（1）系统根据取票单文件对取票单进行有效性检查，无效的输出无效取票信息；（2）持有有效取票单的顾客在补交了剩余款后将获得机票；（3）记账文件将被更新，机票以及顾客信息将被登录到机票文件。

订单中有订票日期、旅行日期、时间要求（上午、下午、晚上）、出发地、目的地、顾客姓名、身份证号、联系电话。



0层流程图



预定流程图



取机票流程图

3．一个简化的图书馆信息管理系统有以下功能：(1)借书：输入读者借书证，系统检查借书证是否有效；查阅借书文件，检查该读者所借图书是否超过10本，若已达10本，显示信息“已经超出借书数量”，拒借；未达10本，办理借书（检查库存、修改库存信息并将读者借书信息登入借书记录）(2)还书：输入书号和读者号，从借书记录中读出与读者有关的记录，查阅所借日期，如果超过3个月，作罚款处理。否则，修改库存信息与借书记录。(3)查询：可通过借书记录、库存信息查询读者情况、图书借阅情况及库存情况，打印各种统计表。

请就以上系统功能画出分层的DFD图，并建立重要条目的数据字典。



0层数据流程图



还书流程图



借书流程图



查询流程图

数据字典：

|  |  |  |  |
| --- | --- | --- | --- |
| 元素名称 | 名称 | 类型 | 说明 |
| IPO1 | 还书处理 | 处理 | 根据书号做还书处理，违规做处罚 |
| IPO2 | 有效性检查 | 处理 | 检查读者号，图书号的有效性 |
| IPO3 | 借书处理 | 处理 | 按读者号，图书号进行借书处理 |
| IPO4 | 查询 | 处理 | 管理员对一些基本信息进行查询 |
| IPO5 | 打印报表 | 处理 | 打印查询到的信息 |
| IPO6 | 处罚管理 | 处理 | 根据规则对延期给予相应的处罚 |
| IPO7 | 数量检查 | 处理 | 对读者借出的数量是否符合规定进行检查 |
| DS001 | 读者信息 | 数据存储 | 读者信息录入，修改，删除，保存 |
| DS002 | 库存信息 | 数据存储 | 图书的基本的情况以及所有图书的整体情况 |
| DS003 | 错还信息 | 数据存储 | 存储借还书信息，系统自动处理，不能人工录入 |
| DS004 | 处罚信息 | 数据存储 | 存储延期的处罚信息 |

**第二套试卷**

**一、选择题**  （**软件工程作业2 第5～7章）**

1. 从本质上说，面向对象是“先”确定（A）后确定执行的动作。

A.动作的主体 B.属性 C.关系 D.方法

1. （B）是对象的静态特征。

A.方法 B.属性 C.关系 D.操作

1. 对象的（D）使对象的动态特征

A.特征 B.属性 C.关系 D.操作

1. 属于某个类的一个具体对象称为该类的（A）。

A.实例 B.虚拟 C.继承 D.多态

1. UML是（A）。
2. 一种可视化的建模语言
3. 是一种可视化的程序设计语言
4. 是一种过程
5. 是一种方法
6. 封装的的作用是（A）。

A.复用 B.对象 C.实例 D.方法

8. 面向对象分析的核心在于建立（B）模型。

A. 建立正确的模型

B. 识别问题域对象

C.识别对象之间的关系

D.上面所有

9. 面向对象分析过程中建立的模型有（D ）。

* 1. A. 数据模型、功能模型、活动模型
  2. B. 对象模型、功能模型、测试模型
  3. C. 属性模型、功能模型、对象模型
  4. D. 对象模型、功能模型、动态模型

10. 下列的（ C）不是分析建模的目的。

* 1. A. 定义可验证的软件需求
  2. B. 描述客户需求
  3. C. 开发一个简单的问题解决方案
  4. D. 建立软件设计的基础

11. 下列的（ C）不属于面向对象分析模型。

* 1. A. 用例图
  2. B. 类图
  3. C. 实体关系图
  4. D. 顺序图

12. （A ）用于描述系统中概念层次的对象。

A.分析类

B. 界面类

C.实体类

D.控制类

13. 在基于用例的面向对象分析过程中，定义交互行为的关键在于通过描述分析类实例之间的（A）将用例的职责分配到分析类中。

A. 消息传递

B 关联关系

C 继承关系

D 上下文关系

14.用例的实现细节不会在（B）描述。

A. 用例说明

B. 用例图

C.活动图

D.顺序图

15.用例的职责通常分配给（D）的对象。

A. 发送消息

B. 接收消息

C.发送和接收双方

D.分析类

16.组织机构图是（C）。

A. UML的一个最新图

B. 类图的一种

C.用于识别角色的辅助图

D.用例图的一种

17.在图书馆系统中，假如已经构造了一个一般借书者类，后来发现图书馆的学生和教师在借书中有不同要求。请问在面向对象设计中用（B）方法可以方便地设计这两个类？

A.信息隐藏 B.继承 C.动态联编 D.代码复制

18．框架是一组可用于不同应用的（A）集合

A. 类 B.对象 C.模块 D.代码

19．下面哪个选项不能描述子系统之间的关系。（D）

A. 请求——服务关系 B.继承关系 C.依赖关系 D.数据关系

20．每个子系统的接口上定义了若干（B），体现了子系统的功能。

A. 说明 B.操作 C.属性 D.关系

21．通常将具有共性的非功能要求设计在（B），以减少重新开发的工作量。

A. 接口层 B.中间层 C.最低层 D.最高层

22.每个用例至少应该有一个（C），它通常没有属性而只有方法，只是起协调和控制作用。

A. 接口类 B.实体类 C.控制类 D.界面类

23.类有三种类型即（A）。

A. 边界类、实体类、控制类

B. 实体类、数据类、界面类

C. 控制类、实体类、数据类

D. 界面类、控制类、公共类

24.下面那个选项不是正确的面向对象设计思想。（C）

A.对接口进行设计

B.发现变化并且封装它

C.先继承后组合

D.先组合后继承

25.定义类的属性类型时尽量使用已有的类型，太多的自定义类型会降低系统的（A）指标。

A.可维护性 B.安全性 C.开发效率 D.可移植性

**二、简答题**  （**软件工程作业2 第5～7章）**

1. 举例说明类和对象的关系。

答：学生可作为一个类---学生类，每个学生就是这个学生类的一个实例，例如，学生张三就是学生类的一个实例。

1. 面向对象分析的关键步骤有哪些？应建立哪几个模型？

答：关键步骤有识别问题域的对象并分析它们相互之间的关系，建立简洁、精确、可理解的正确模型；应建立的模型有功能模型，对象模型，动态模型。

1. 什么是实体类、边界类和控制类？为什么将分析类划分成这三种类型？

答：（1）实体类保存要存入永久存储体的信息。实体类通常在事件流或交互图中，是对用户最有意义的类。

边界类位于系统与外界的交界处，包括所有的窗体、报表、系统硬件接口、与其他系统的接口。

控制类负责协调其他类的工作。每个用例中至少应该有一个控制类，它控制用况中的事件顺序。

（2）分为三种类型是因为它们各自的功能不同。界面类——描述系统与角色之间的接口。

控制类——在分析模型内表示协调、顺序、事务处理以及控制其他对象的类。

实体类——为需要长久保存的信息进行建模的类。

1. UML中有哪几类个视图，它们的作用分别是什么？

答：（1）用例视图（Use-Case view）

用例视图用于描述系统的功能集。它是从系统外部以用户角度，对系统做的抽象表示。用例视图所描述的系统功能依靠于外部用户或另一个系统触发激活，为用户或另一个系统提供服务，实现与用户或另一个系统之间的交互。

用例视图中可以包含若干个用例，用例表示系统能够提供的功能，用例视图是其他视图的核心和基础。其他视图的构造依赖于用例视图中所描述的内容，因为系统的最终目标是实现用例视图中描述的功能，同时附带一些非功能性的特性，因此用例视图影响着所有其他的视图。

（2）逻辑视图（Logical view）

如果说用例视图描述系统“做什么”，那么逻辑视图就是描述“怎么做”。系统的静态结构描述类、对象和它们之间的关系，反映的是系统静态特征或结构组成。

（3）组件视图（Component view）

组件视图用来描述系统实现的结构和行为特征，反映系统各组成元素之间的关系。组件视图由组件图实现，主要供开发者和管理者使用。

（4）并发视图（Concurrency View）

并发视图用来描述系统的动态和行为特征。并发视图将任务划分为进程或线程形式，通过任务划分引入并发机制，可以高效地使用资源、并行执行和处理异步事件。除了划分系统为并发执行的进程或线程外，并发视图还必须处理通信和同步问题。

（5）配置视图（Deployment View）

配置视图体现了系统的实现环境，反映系统的物理架构。配置视图还包括一个映射，该映射显示在物理架构中组件是怎样分配的。

5. 顺序图在分析阶段的作用？

答：顺序图描述了一组交互对象间的交互方式，它表示完成某项行为的对象和这些对象之间传递消息的时间顺序

6. 活动图在分析阶段的作用？

答：活动图反映系统中从一个活动到另一个活动的流程，强调对象间的控制流程。活动图特别适合描述工作流和并行处理过程。具体地说活动图可以描述一个操作过程中需要完成的活动；描述一个对象内部的工作；描述如何执行一组相关的动作，以及这些动作如何影响它们周围的对象；说明一个业务活动中角色、工作流、组织和对象是如何工作的。

7.比较结构化设计和面向对象设计区别？

答：结构化设计：系统被划分成多个模块，这些模块被组织成一个树型结构。根就是主模块，叶子是最低级的功能模块。这棵树也表示调用结构：每个模块都调用自己的直接下级模块，并被自己的直接上级模块调用。顶层模块负责收集应用程序最重要的那些执行策略，底层模块实现处理细节。在这个体系结构中越靠上层位置，概念的抽象层次就越高。但是，由于上层模块需要调用下层模块，所以这些上层模块就依赖于下层模块的处理细节。也就是说，当实现细节发生变化时，抽象也会受到影响；如果想复用某一个抽象的话，就必须把它依赖的细节都一起拖过去。

面向对象设计：上层创建的抽象不依赖于任何细节，而细节则高度依赖于上层的抽象。这种依赖关系的变化正是结构化设计与面向对象设计的根本区别。

8.什么是框架，它与“设计”有什么关系？

答：框架是一组可用于不同应用的类的集合。框架中的类通常是一些抽象类并且相互有联系，可以通过继承的方式使用这些类,当面向对象设计进行底层设计时，对每个类进行详细设计，设计类的属性和操作，优化类之间的关系，就可以设计成框架来对类的设计进行构架和优化。

9.系统的物理构架中应该包括哪些信息？

答：用UML的配置图描述系统的物理构架，然后将需求分析阶段捕获的系统功能分配到这些物理节点上。包括的信息为节点的拓扑结构、硬件设备配置、通信路径、各个节点上运行的系统软件配置、应用软件配置。

三、应用题 （ **软件工程作业2 第5～7章）**

1. 一个多媒体商店系统包含一个由媒体文件构成的数据库，有两类媒体文件：图像文件和声音文件。每个媒体文件都有名称和唯一的编码，而且文件包含作者信息和格式信息，声音文件还包含声音文件的时长(以秒为单位)和作者的信息。假设每个媒体文件可以由唯一的编码所识别，系统要提供以下功能：
2. 媒体库中可以添加新的特别媒体文件。
3. 通过给定的文件编码查找需要的媒体文件。
4. 从媒体库中删除指定的媒体文件。
5. 给出媒体库中媒体文件的数量。

考虑类imageFile和audioFile应该具有哪些恰当的属性和方法

答：imageFile类的属性有:Name(文件名称)，Code（编码），AuthorInfo（作者信息），Format（格式信息）；audioFile类除了拥有以上属性外还有属性Time（文件时长）；

imageFile类的方法：Add（imageFile）（添加图像文件），Delete（Code）（删除文件），GetFile（Code）（查找文件），GetNumber（）（返回媒体文件的数量）；

audioFile类的方法：Add（audioFile）（添加声音文件），Delete（Code）（删除文件），GetFile（Code）（查找文件），GetNumber（）（返回媒体文件的数量）。

1. 若把学生看成一个实体，它可以分成多个子实体，如小学生、中学生和大学生等。在面向对象的设计中，可以创建如下4个类：类Student、类Elementary Student、Middle Student；类University Student。试给出这4个类的属性以及它们之间的关系。

答：Student类为基类，其他三个类都是Student的子类。

Student类的属性有：Name（姓名），Sex（性别），Age（年龄），Class（班级）。

Elementary Student类的属性有：IsTeamMember（是否是少先队员），IsGoodStudent（是否是三好学生）。

Middle Student类的属性为：IsMember（是否是团员），Subject（文科理科）。

University Student类的属性为：IsPartyMember（是否党员），Department（学院），Specialty（专业）。

3.某学校领书的工作流程为：学生班长填写领书单，班主任审查后签名，然后班长拿领书单到书库领书。 书库保管员审查领书单是否有班主任签名，填写是否正确等，不正确的领书单退回给班长；如果填写正确则给予领书并修改库存清单；当某书的库存量低于临界值时，登记需订书的信息。每天下班前为采购部门提供一张订书单。 用活动图来描述领书的过程。

解：



4.使用顺序图描述下面的情景，当用户在自己的计算机上向网络打印机发出一个打印任务时，他的计算机便向打印机服务器发送一条打印命令print(file)，打印机服务器如果发现网络打印机处于空闲状态，则向打印机发送打印命令print(file)，否则向打印队列发送一条保存命令store(file)。解：



5.某报社采用面向对象技术实现报刊征订的计算机管理系统，该系统基本需求如下：

（1）报社发行多种刊物，每种刊物通过订单来征订，订单中有代码，名称，订期，单价，份数等项目，订户通过填写订单来订阅报刊。  
 （2）报社下属多个发行站，每个站负责收集登陆订单、打印收款凭证等事务。  
 （3）报社负责分类并统计各个发行站送来的报刊订阅信息。  
 请就此需求建立对象模型。

解：



**第三套试卷**

* + 1. 选择题 （**软件工程作业3 第8～11章）**

1. 结构化语言是介于自然语言和形式语言之间的一种（D）。

A．半形式语言B．计算机语言 C．低级语言 D．高级语言

1. 程序设计语言的技术特性不应包括（C）。

A、数据结构的描述性B、抽象类型的描述性

C、数据库的易操作性D、软件的可移植性

1. 程序的三种基本控制结构是（B）。

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

1. 结构化程序设计主要强调的是（D）。

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

1. 下列关于功能性注释不正确的说法是(B)。

A.功能性注释嵌在源程序中，用于说明程序段或语句的功能以及数据的状态

B.注释用来说明程序段，需要在每一行都要加注释

C.可使用空行或缩进，以便很容易区分注释和程序

D.修改程序也应修改注释

1. 下列关于效率的说法不正确的是(D)。

A.效率是一个性能要求，其目标应该在需求分析时给出

B.提高程序效率的根本途径在于选择良好的设计方法，数据结构与算法

C.效率主要指处理机时间和存储器容量两个方面

D.程序的效率与程序的简单性无关

1. 结构化维护与非结构化维护的主要区别在于(B )。

A.软件是否结构化 B.软件配置是否完整 C.程序的完整性 D.文档的完整性

1. 关于JAVA语言下列说法不正确的是（B）。

A.跨平台的 B.动态指针 C.解释型的 D.面向对象的

1. 使用白盒测试方法时，确定测试数据应根据（A）和指定的覆盖标准。

A程序内部逻辑 B．程序复杂结构

C．使用说明书 D．程序的功能

1. 确认测试主要涉及的文档是（A）。

A、需求规格说明书B、概要设计说明书

C、详细设计说明书D、源程序

1. 测试的关键问题是（D）。

A.如何组织对软件的评审 B.如何验证程序的正确性

C.如何采用综合策略 D.如何选择测试用例

1. 黑盒测试在设计测试用例时，主要研究（A）。

A.需求规格说明 B.详细设计说明

C.项目开发计划 D.概要设计说明与详细设计说明

1. 下面哪些测试属于黑盒测试（B）。

A、路径测试B、等价类划分 C、条件判断 D、循环测试

1. 在下述哪一种测试中，测试人员必须接触到源程序（B）。

A、功能测试 B、结构测试C、功能测试和结构测试 D、性能测试

1. 检查软件产品是否符合需求定义的过程称为（D）。

A、确认测试B、集成测试C、验收测试D、验证测试

1. 软件测试方法中，黑盒、白盒测试法是常用的方法，其中白盒测试主要用于测试（D）。
2. 结构合理性B、软件外部功能C、程序正确性D、程序内部逻辑
3. 软件维护中，因修改交互输入的顺序，没有正确的记录而引起的错误是（A）产生的副作用。

A、文档B、数据 C、编码D、设计

1. 以下属于软件维护阶段文档的是(C)。

A、软件需求说明B、操作手册C、软件问题报告D、测试分析报告

1. 产生软件维护的副作用，是指（C）。

A、开发时的错误 B、隐含的错误

C、因修改软件而造成的错误 D、运行时误操作

1. 维护中，因误删除一个标识符而引起的错误是（C）副作用。

A、文档 B、数据 C、编码 D、设计

1. 可维护性的特性中，相互促进的（A）。

A、可理解性和可测试性B、效率和可移植性

C、效率和可修改性 D、效率和结构好

1. 软件维护的困难主要原因是（C）

A、费用低 B、人员少 C、开发方法的缺陷 D、用户不配合

1. 软件维护费用高的主要原因是（C）。

A、人员少B、人员多C、生产率低D、生产率高

1. 为了适应软硬件环境变化而修改软件的过程是(C)。

A、校正性维护 B、完善性维护 C、适应性维护D、预防性维护

1. 可维护性的特性中，相互矛盾的是(B)

A.可理解性与可测试性 B.效率与可修改性

C.可修改性和可理解性 D.可理解性与可读性

1. 各种不同的软件维护中，以（C）维护所占的维护量最小。
2. 纠错性维护B、代码维护C、预防性维护D、文档维护
3. 在变更控制中，（B）可以用来确保由不同用户所执行的并发变更。

A．异步控制B．同步控制C．存取控制D．基线控制

1. 变更控制是一项最重要的软件配置任务，其中“检出”和（A）处理实现了两个重要的变更控制要素，即存取控制和同步控制。

A．登入B．管理C．填写变更要求D．审查

1. 用图表示软件项目进度安排，下列说法中正确的是（B）

A.能够反映多个任务之间的复杂关系

B.能够直观表示任务之间相互依赖制约关系

C.能够表示哪些任务是关键任务

D.能够表示子任务之间的并行和串行关系

1. 基线可作为软件生存期中各开发阶段的一个检查点。当采用的基线发生错误时，可以返回到最近和最恰当的（C）上

A.配置项B.程序C.基线D.过程

* + 1. 简答题 （**软件工程作业3 第8～11章）**

1. 为建立良好的编程风格应遵循什么原则？

答：1）节简化

①不使用不必要的变量和函数

②避免变量重名，变量重名可导致很难被发现的错误。

③尽量减小代码量。

④尽量减少代码的执行时间，提高执行效率。

⑤避免功能冗余的模块。

⑥尽量不使用全局变量。

2）模块化

①确保物理和逻辑功能密切相关。

②限定一个模块完成一个独立的功能，

提高模块的内聚度。

③检查代码的重复率，重复多的代码，

要抽出来作为一个单独的模块。

3）简单化

①采用直接了当的算法，避免使用技巧

性高和难懂的代码。

②使用简单的数据结构，避免使用多维

数组、指针和复杂的表。

③注意对象命名的一致性。

④以手工的方式简化算术和逻辑表达式。

4）结构化

①按标准化的次序说明数据；

②按字母顺序说明对象名

③使用读者明了的结构化程序部件；

④采用直截了当的算法；

⑤根据应用背景排列程序各个部分；

⑥不随意为效率而牺牲程序的清晰度和可读性；

⑦让机器多做烦琐的工作，如重复、库函数；

⑧用公共函数调用代替重复出现的表达式；

⑨避免循环、分支的嵌套层数过高；

⑩单入口单出口。

5）文档化（documentation）

①有效、适当地使用注释。

②协调使用程序块注释和程序行注释。

③保持文档和程序的同步。

6）格式化（layout)

①始终采用统一缩进规则；

②适当插入括号表明运算次序、排除二义性；

③有效地使用空格符以区别程序的不同意群。

1. 什么是软件测试？

答：软件测试从广义上讲,是指软件产品生存周期内所有的检查、评审和确认活动。从狭义上讲，软件测试是为了发现错误而执行程序的过程。或者说，软件测试是根据软件开发各个阶段的规格说明和程序内部结构而精心设计一批测试用例，用这些测试用例运行程序，以发现程序错误的过程。

1. 软件测试的原则是什么？

答：软件测试的原则有：

1. 应该把测试贯穿在整个开发过程之中。事实上，从需求分析阶段开始，每个阶段结束之前都要进行阶段审查，目的是尽早发现和纠正错误。
2. 每个测试用例都应该包括测试输入数据和这组数据输入作用下的预期输出结果。在实际操作中可以列出一张表格，包括每个测试用例的编号、类型、输入数据、预期输出结果、实际输出结果、出错原因分析。
3. 要对每个测试结果进行全面检查，不要漏掉已经出现的错误迹象。
4. 程序员应该尽量避免检查自己编写的代码。测试工作需要严格的工作作风，程序员在测试自己编写的代码时往往会带有一些倾向性，使得他们工作中常常出现一些疏漏。而且，程序员对设计规格说明书的理解错误而引入的错误更是难于发现。
5. 在设计测试用例时，应该包括有效的、期望的输入情况，也要包括无效的和不期望的输入情况。既能够验证程序正常运行的合理输入，也能够验证对异常情况处理的不合理输入数据以及临界数据。在测试程序时，人们常常过多地考虑合法的和期望的输入条件，以检查程序是否做了它应该做的事情，而忽视了不合法的和预想不到的输入条件。事实上，用户在使用系统时，输入一些错误指令和参数是经常发生的，如果软件遇到这种情况不能做出适当的反应，给出相应的提示信息，就可能会误导用户，甚至造成严重损失。
6. 软件中遗留的错误数量与已经发现的错误数量成正比。根据这个规律，对测试中发现错误成堆的模块更要仔细测试。例如，在某个著名的操作系统中，44%的错误仅与4%的模块有关。
7. 回归测试的关联性要特别引起注意，修改一个错误而引起更多错误的现象并不少见。
8. 测试程序时不仅要检查程序是否做了它应该做的事情，还要检查它是否做了不该做的事情。例如，工资软件中，软件只完成在编职工的工资计算和输出，不在编人员的工资是不进行计算和输出的。如果软件将不在编人员的工资信息也输出显然是不合适的。
9. 严格执行测试计划。在测试之前应该有明确的测试计划，内容包括：要测试的软件功能和内容、测试用例和预期结果、测试的进度安排、需要的工具和资源、测试控制方式和过程等。

10）做好测试记录，为统计和维护提供基础数据。

1. 请说明集成测试、系统测试和验收测试有什么不同？

答：集成测试界于单元测试和系统测试之间，起到“桥梁作用”，一般由开发小组采用白盒加黑盒的方式来测试，既验证“设计”，又验证“需求”。  
系统测试的粒度最大，一般由独立测试小组采用黑盒方式来测试，主要测试系统是否符合“需求规格说明书”。  
验收测试与系统测试相似，主要区别是测试人员不同，验收测试由用户执行。

1. 简述单元测试的内容。

答：单元测试的内容有

1）模块接口测试：对通过被测模块的数据流进行测试。为此，对模块接口，包括参数表、调用子模块的参数、全程数据、文件输入/输出操作都必须检查。

2）局部数据结构测试：设计测试用例检查数据类型说明、初始化、缺省值等方面的问题，还要查清全程数据对模块的影响。

3）路径测试：选择适当的测试用例，对模块中重要的执行路径进行测试。对基本执行路径和循环进行测试可以发现大量路径错误。

4）错误处理测试：检查模块的错误处理功能是否包含有错误或缺陷。例如，是否拒绝不合理的输入;出错的描述是否难以理解、是否对错误定位有误、是否出错原因报告有误、是否对错误条件的处理不正确;在对错误处理之前错误条件是否已经引起系统的干预等。

5）边界测试：要特别注意数据流、控制流中刚好等于、大于或小于确定的比较值时出错的可能性。对这些地方要仔细地选择测试用例，认真加以测试。

5) 此外，如果对模块运行时间有要求的话，还要专门进行关键路径测试，以确定最坏情况下和平均意义下影响模块运行时间的因素。这类信息对进行性能评价是十分有用的。

1. 什么是白盒测试？它适应哪些测试？

答：白盒测试也叫结构测试，测试者需要了解被测程序的内部结构。白盒测试通常根据覆盖准则设计测试用例，有语句覆盖、判定覆盖、条件覆盖、判定/条件覆盖和条件组合覆盖。

白盒测试适用与单元测试、集成测试、系统测试

1. 非渐增式测试与渐增式测试有什么区别？渐增式测试如何组装模块？

答：1）渐增式可以较早地发现模块界面之间的错误，非渐增式则要到最后将所有模块连接起来时才能发现此类错误。

2）非渐增式需要较多的人工

3）渐增式有利于排错。如果模块界面间有错，如果用非渐增式，这类错误到最后联合测试时才能发现，此时很难判断错误发生在哪一部分，但如果用渐增式，这类错误就较容易定位，它通常与最新加上去的模块有关。

4）渐增式比较彻底。

5）渐增式需要较多的机器时间。

6）非渐增式方式，开始时允许几个测试人员并行工作，对大型系统来说，是很有意义的。

1. 采用黑盒技术设计测试用例有哪几种方法？这些方法各有什么特点？

答：黑盒测试设计测试用例的方法有1）等价类划分法 特点是：a、测试的内容相同 b、一个等价类中的某个测试可以发现缺陷，那么这个等价类的其他测试也能发现缺陷 c、一个等价类中的某个测试不能发现缺陷，那么这个等价类的其它测试也不能发现缺陷2）边界值分析法 特点是：用边界值分析时时常收获较大，它是在等价类的边界上选择测试用例3）错误推测法 特点是：没有确定的步骤，很大程度上是凭借经验进行的 4）因果图法 特点是：考虑输入条件间的组合关系；考虑输出条件对输入条件的信赖关系；测试用例发现错误的效率高；能查出功能说明中的某些不一致或遗漏

1. 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。

答：白盒测试的覆盖标准有：语句覆盖->判定覆盖->条件覆盖->判定/条件覆  
盖->条件组合覆盖->路径覆盖 检错能力由弱到强

1. 根据下面程序代码，画出程序流程图，然后设计满足条件/判定覆盖、条件组合覆盖的测试用例。

BEGIN

T:=0

IF ( X>=80 AND Y>=80 )THEN

T:=1

ELSE IF ( X>=90 AND Y>=75 ) THEN

T:=2

ENDIF

IF (X>=75 AND Y>=90) THEN

T:=3

ENDIF

ENDIF

RETURN



测试用例：

条件/判定覆盖：

1 X=80 Y=80

2 X=90 Y=75

3 X=50 Y=50

条件组合覆盖：

1 X=80 Y=80

2 X=80 Y=75

3 X=75 Y=80

4 X=75 Y=75

5 X=90 Y=75

6 X=90 Y=70

7 X=80 Y=75

8 X=80 Y=70

9 X=75 Y=90

10 X=75 Y=90

11 X=70 Y=90

12 X=70 Y=85

1. 软件的可维护性是软件设计师最关注的性能，谈谈为了获得软件良好的可维护性，在设计时应该注意哪些问题？

答：为获得软件的良好可维护性，在设计阶段应该做一些变更实验，检查系统的可维护性、灵活性和可移植性，设计时应该将今后可能变更的内容与其他部分分离开来，并且遵循高内聚、低偶合的原则。

1. 在软件文档中，你认为哪些文档对于软件的维护最重要？

需求分析文档、设计文档、程序文档对软件维护很重要

1. 软件维护时的源程序修改策略是什么？

答：软件维护时的源程序修改策略是：由于软件维护最终落实在修改源程序和文档上。为了正确、有效地修改源程序，通常要先分析和理解源程序，然后才能修改源程序，最后重新测试和验证源程序。

1. 软件项目计划包括那些内容？

答：软件项目计划的内容包括：

1.范围。对该软件项目的综合描述,定义起所要做的工作以及性能限制,它包括:

　　(1)项目目标。

　　(2)主要功能。

　　(3)性能限制。

　　(4)系统接口。

　　(5)特殊要求。

　　(6)开发概述。

2.资源。

　　(1)人员资源。

　　(2)硬件资源。

　　(3)软件资源。

　　(4)其他。

3.进度安排。

进度安排的好坏往往会影响整个项目的按期完成,因此这一环节是十分重要的。制定软件进度与其他工程没有很大的区别 ，其方法主要有：

　　(1)工程网络图。

　　(2)Gantt图。

　　(3)任务资源表。

　　(4)成本估算。

　　(5)培训计划。

15.什么是软件配置管理？什么是基线？

答：配置管理能够系统地处理变更，从而使得软件系统可以随时保持其完整性。配置管理又可成为变更控制，可以用来评估提出的变更请求，跟踪变更，并保持系统在不同时间的状态。软件配置管理是一套规范、高效的软件开发基础结构。基线是：已经通过正式复审和批准的某规约或产品，它因此可以作为进一步开发的基础，并且只能通过正式的变化控制过程改变。

* + 1. 应用题 （**软件工程作业3 第8～11章）**

2.有的学生总是问老师“我应该掌握什么程序设计语言更好？”你认为该如何回答这个问题？

答：首先必须掌握C语言，面向对象的程序设计语言，各有各的优势，看你想往哪方面用了，如果是服务器端的应用最好是java，跨平台性好；如果是，一般的桌面应用，C#是较好的选择；至于易上手程度，VB,Delphi等是上手比较快的；至于数据库应用本人认为PB是最好的选择，VB,Delphi也是比较好的选择；当特别强调效率的时候VC是不二的选择，其中VC最为复杂，也最为灵活，“偷懒的程序员用VB，聪明的程序员用Delphi, 真正的程序员用VC”，

如果想要设计一些系统级的核心程序，最好用VC，C语言也可，但是不如VC强大。无论什么语言，通一门即可，它们只是工具，设计模式和算法才是灵魂。

3.编写C语言程序，要求输入一个学生的两门课成绩（百分制），计算该生的总分并要求输出成绩等级‘A’、‘B’、‘C’、‘D’、‘E’。总分在180分以上为‘A’，160~179分为‘B’，140~159分为‘C’，120~139分为‘D’，120分以下为‘E’。使用switch语句编写，具体要求： ①成绩通过键盘输入，输入之前要有提示信息。②若输入的成绩不是百分制成绩，则给出错误提示信息，并且不再进行下面的等级评价；若输入的成绩是百分制成绩，则计算总分，并根据要求评价等级。

答：

#include <stdio.h>

#include <conio.h>

/\*原型声明\*/

void input(int \*,int\*);

int sum(int,int);

void convert(int,char\*);

void degree(char,int);

void main(void)

{

/\*定义变量\*/

int nfst\_course=-1,nsec\_course=-1; /\*nfst\_course第一门课成绩，nsec\_course第二门课成绩\*/

int \*pfst=&nfst\_course,\*psec=&nsec\_course;

char cdegree='\0'; /\*成绩等级\*/

char \*idegree=&cdegree;

int nsum;

clrscr();/\*清屏\*/

do

{

input(pfst,psec); /\*输入成绩\*/

}while(nfst\_course==-1&&nsec\_course==-1);/\*正确输入时跳出循环\*/

/\* printf("add nfst\_course and nsec\_course\n");\*/

nsum=sum(nfst\_course,nsec\_course);/\*求和\*/

convert(nsum,idegree);/\*转换\*/

degree(cdegree,nsum); /\*评价等级\*/

}

void input(int \*pfst,int \*psec) /\*输入成绩数据\*/

{int fst,sec;

int bflag=1; /\*标志变量\*/

while(bflag)

{printf("Plsease input the score of first course:");

scanf("%d",&fst);

printf("First course:%d\n",fst);

if(fst>=0&&fst<=100)

{\*pfst=fst;bflag=0;break;}

else

{ clrscr();/\*清屏\*/

printf("Result is not Percentile system,please reinput\n"); /\*非百分制，需重新输

入\*/

/\*return;\*/

}/\*else\*/

}/\*while\*/

bflag=1; /\*使标志变量值为真\*/

while(bflag)

{printf("Plsease input the score of second course:");

scanf("%d",&sec);

printf("Second course:%d\n",sec);

if(fst>=0&&fst<=100)

{\*psec=sec;bflag=0;break;}

else

{ clrscr();/\*清屏\*/

printf("result is not Percentile system,please reinput\n");

/\*return;\*/

}/\*else\*/

}/\*while\*/

return;

}/\*input\*/

int sum(int first,int second) /\*求和\*/

{

return(first+second);

}

void convert(int score,char \*idegree) /\*把分数转换成等级\*/

{

if(score>=180)

\*idegree='A';

else if(score>=160&&score<179)

\*idegree='B';

else if(score>=140&&score<159)

\*idegree='C';

else if(score>=120&&score<139)

\*idegree='D';

else

\*idegree='E';

}

void degree(char cdegree,int nsum) /\*等级判定\*/

{

switch(cdegree)

{

case 'A': printf("sum of sorce is:%d",nsum);printf("the degree of the student's score is

A\n");break;

case 'B': printf("sum of sorce is:%d",nsum);printf("the degree of the student's score is

B\n");break;

case 'C': printf("sum of sorce is:%d",nsum);printf("the degree of the student's score is

C\n");break;

case 'D': printf("sum of sorce is:%d",nsum);printf("the degree of the student's score is

D\n");break;

case 'E': printf("sum of sorce is:%d",nsum);printf("the degree of the student's score is

E\n");break;

default: printf("Error!\n");break;

}

}

4.请修改下面的程序，使它的可阅读性更好。

WHILE P DO

IF A>O THEN A1 ELSE A2 ENDIF;

S1;

IF B>0 THEN B1;

WHILE C DO S2;S3 ENDWHILE;

ELSE B2

ENDIF;

B3

ENDWHILE;

修改后：

WHILE P

DO{

IF A>O THEN A1;

ELSE A2

ENDIF

S1;

IF B>0 THEN {B1;

WHILE C

DO

S2;

S3;

ENDWHILE}

ELSE B2;

ENDIF

B3;

}

ENDWHILE