19052711 笼统的总结

重点在 3/5/6/7章 用心看，大题汇总写在最后，前面是一些题目，接下来是一些章节知识汇总

可能有一些错误，请见谅哦

1．描述软件开发过程中各种活动如何执行的模型称为 **周期** 模型。

2．可行性研究实质上是进行一次简化、压缩了的 **需求分析和设计过程** 。

3．需求分析阶段所要编写的主要文档有： **需求规格说明书** 、 **（初步的）用户使用手册** 、 **确认测试计划** 。

4．面向数据流的设计是以需求分析阶段产生的数据流图为基础，按一定的步骤映射成软件结构，因此又称为 **结构化设计** 。

5．软件详细设计的目标不仅是逻辑上正确地实现每个软件模块的功能，还应使软件设计的处理过程 **清晰易懂** 。

6．源程序中注释是帮助理解程序的重要手段，注释分为 **序言性注释** 和 **功能性注释** 两类。

7．用等价类划分方法设计测试用例时，如果被测程序的某个输入条件规定了取值范围，则可以划分出

**一个合理的，两个不合理的** 等价类。

8．UML表述的动态模型中，顺序图描述对象之间的 **动态交互关系** ，重点体现对象间消息传递的 **时间顺序** 。

9．对象是面向对象开发方法的基本单位，它是一组 **属性** 和它可以执行的一组 **操作** 的封装体。应用执行对象的 **操作（或方法）** 可以改变该对象的 **属性（或状态）** 。

10．类和类之间的关系通常有 **关联** 、 **泛化** 、 **依赖** 、 **细化** 四种关系。

UML中顺序图反映对象之间发送消息的时间顺序，它与（ 协作图 ）是同构的。

在UML表示的动态模型中，使用（ 顺序图与协作图 ）来描述对象之间动态行为的交互关系。

在面向对象的方法中，系统的静态结构是由（ 对象模型 ）描述的。

软件可维护性的特性中，相互促进的特性是（ 可理解性和可测试性 ）

黑盒测试是从（ 用户 ）观点出发的测试，白盒测试是从（ 开发人员 ）观点出发的测试。

详细设计的任务是确定每个模块的（ 内部特性 ）。

模块的内部过程描述就是模块内部的（ 算法设计 ），它的表达形式就是详细设计语言。

软件需求分析阶段工作，可以分成以下4个方面的内容：对需求的识别、分析和综合、规格说明以及（ 需求分析评审 ）。

对软件进行分解，是为了（ 降低模块的复杂程度 ）。

将几个逻辑上相似的成分放在一个模块中，该模块的内聚度是（ 逻辑性 ）的。

软件设计将涉及软件的结构、过程和模块的设计，其中，软件过程是指（ 模块的操作细节 ）。

确认测试是针对需求的，测试计划是在前期编写的，测试执行是在编码结束后

划分模块时，一个模块的（ 作用范围应在其控制范围之内 ）。

人机界面设计需要处理好：系统响应时间，用户帮助设施，出错信息处理，命令交互 4个设计问题

1. **实现**

实现：通常把编码和测试统称为实现：

软件测试在软件生命周期中横跨两个阶段。

软件测试横跨两个阶段：（完成一个模块后）单元测试以及软件系统测试，前者由编写者测试，后者由专门测试人员测试

软件测试的目标(简答)：

测试是为了发现程序中的错误而执行程序的过程。

好的测试方案是极有可能发现迄今为止尚未发现的错误的测试方案，

成功的测试是发现了至今为止尚未发现的错误的测试。

测试方法

黑盒测试：功能测试，如果已经知道了产品应该具有的功能，可以通过测试来检验是否每个功能 都能正常使用； 应使每个测试方案只覆盖一个无效的等价类。

等价划分

边界值分析

错误推测

白盒测试：结构测试，如果知道产品的内部工作过程 ，可以通过测试来检验产品内部动作是否按 照规格说明书的规定正常进行。

逻辑覆盖：是以程序内部的逻辑结构为基础的设计测试用例的技术。是对一系列测试过程的总称， 这组测试过程逐渐进行越来越完整的通路测 试。它属白盒测试。 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 点覆盖 边覆盖。

控制结构测试：基本路径测试，调减测试，循环测试，

测试步骤：

模块测试，

子系统测试，

系统测试，

验收测试，(把软件系统作为单一的实体进行测试，测试内容与系统测试基本类似，但是在用户积极参与下进行的。也称确认测试)

平行运行。

测试的目标是：发现软件中的错误

软件 可靠性：

程序在给定的时间间隔内，按照规格说明书的规定，成功地运行的概率（0到t内没有失效）

软件的可用性：

程序在给定的时间点，按照规格说明书的规定，成功的运行的概率。（时刻t正常运行，之前可能会失效被修复了）

**第八章**

软件维护的种类

1.改正性维护：20% 诊断改正错误

2.适应性维护：25% 为变化的环境适当配合

3.完善性维护：50% 功能增加与修改

1. 预防性维护：5% 改进未来的可维护性或可靠性

软件维护的特点：

结构化维护与非结构化维护差别巨大；

维护的代价高昂

维护的问题很多

**第九章 面向对象方法学导论**

OO = object + class + inheritance + communication with messages

面向对象 = 对象 + 分类 + 继承 + 消息通信

面向对象即使用对象又使用类和继承等机制，而且对象直接仅能通过传递消息实现彼此通信，

面向对象方法学的优点：

与人类习惯的思维方法一致。

稳定性好。

可重用性好。

较易开发软件产品。

可维护性好。

面向对象的概念：

对象：对问题域中某个实体的抽象，设立某个对象就表示系统具有保存有关它的信息与它进行的交互能力。特点：以数据为中心，对象是主动的，实现了数据的封装，本质上有并行性，模块独立性好。

其他概念：

类：具有相同数据和数据操作的一组相似对象的定义。

实例：由某个实例的类所描述的一个具体的对象。

消息：要求某个对象执行在定义它的那个类中所定义的某个操作的规格说明，接收消息的对象，消息选择符，零个或多个变元。

方法：对象所能执行的所有操作，就是类中所定义的服务。

属性：类中所定义的数据，即对客观世界所具有的性质的抽象。

封装：有清晰的边界，有确定的接口(协议)，受保护的内部实现，

继承：直接获得已有的性质和特性。

多态性：在类等级的不同层次中可以共享一个行为的名字。

重载：函数重载，在同一作用域内的若干个参数特征不同的函数可以使用相同的函数名字，运算符重载，同一运算符可以施加于不同类型的操作数上。

面向对象建模：

3种 形式的模型，、

第一是描述系统数据结构 的对象模型(UML类图)，

第二是描述系统控制结构的动 态模型(状态转化图)；

第三是描述系统功能的功能模型(用例图)。

对象模型：表示静态的、结构化的系统的“数 据”性质。 它是对模拟客观世界实体的对象以及对象彼 此间的关系的映射，描述了系统的静态结构。

类与类之间通常有关联、泛化(继承)、依赖和 细化等4种关系。

第一章，软件工程导论：

软件危机：计算机软件开发维护过程中所遇到得一系列严重的问题。

软件危机的典型表现：

对软件开发成本和进度的估计常常很不准确。

用户对以完成的软件系统不满意的现象经常发生。

软件产品质量靠不住

软件常常是不可维护的

软件常常没有适当的文档资料。

软件成本在计算机系统总成本所占比例逐年上升。

软件开发生产率提高的速度，远远跟不上计算机应用迅速普及深入的趋势。

软件工程导论

软件=程序+数据+相关文档

软件生存周期（Software Life Cycle）：一个软件产品从定义、开发、维护到废弃的时间总和称为软件的生存周期。

1.软件定义阶段： 该阶段必须要回答的问题是“需要软件解决的问题 是什么”，

(1) 问题定义： 通过对客户的访问调查，系统分析员扼 要地写出关于问题性质、工程目标和工程规模的书 面报告，经过讨论和必要的修改之后这份报告应该得到客户的确认。

（2）可行性研究： 提交“可行性研究报告”，要回答“对于上 一个阶段所确定的问题有行得通的解决方法 吗？”

（3）软件需求分析： 主要确定目标系统必须具备哪些功能，提交 “需求规格说明书”，描述软件的功能和性 能，确定软件设计的限制和软件与其他系统 元素的接口，定义软件的其他有效性需求。

2.软件开发阶段： 该阶段的任务是设计实现已定义的并经过需 求分析的软件系统。

（4）总体设计 （概要设计）： 需要解决的问题是“应该如何宏观地解决问 题”确定软件德模块功能，得出意义明确的 功能模块，确定每个模块的输入、输出以及相互联系。

（5）详细设计（模块设计）： 给出具体实现这个系统的步骤，但还不是编 写程序,而是设计出程序的详细规格说明（类 似于工程师的工程蓝图），它们包含必要的 细节，程序员可以根据它们写出实际的程序 代码。

（6）编码和单元测试 ：写出正确的容易理解、容易维护的程序模块。程序员应该根据目标系统的性质和实际环境 选取一种适当的高级程序设计语言把详细设 计的结果翻译成用选定的语言书写的程序， 并仔细测试编写出的每一个模块。

（7）综合测试 ：关键任务是通过各种类型的测试及相应的调 试使软件大道预定的要求。 此阶段最重要的测试是：集成测试、验收测 试。（1）集成测试：是根据设计的软件结构，把经过单 元测试的模块按照某种选定的策略装配起来。 （2）验收测试：按照规格说明书的规定，由用户（或在用户积极参加下）对目标系统进行验收。

（8）软件维护： 运行阶段的任务是保障软件的正常运行以及对软件 进行维护。为了排除软件系统中可能隐含的错误， 适应用户需求及系统操作环境的变化，需要继续对 系统进行修改或扩充。

把软件生存周期中各项开发活动的流程用一 个合理的框架-开发模型来规范描述，这就是 软件过程模型，或称为软件生存周期模型。

瀑布模型 ：

瀑布模型的优缺点

1、瀑布模型有以下优点： 1）为项目提供了按阶段划分的检查点。 2）当前一阶段完成后，您只需要去关注后续阶段。 3）可在迭代模型中应用瀑布模型。

2、瀑布模型有以下缺点： 1）在项目各个阶段之间极少有反馈。 2）只有在项目生命周期的后期才能看到结果。 3）通过过多的强制完成日期和里程碑来跟踪各个项目阶段。

快速原型模型 ：(简答题)是快速建立起来的可以在计算机上运行的程序，它 所能完成的功能往往是最终产品能完成的功能的一 个子集。 快速原型模型又称原型模型，它是增量模型的另一 种形式；它是在开发真实系统之前，构造一个原 型，在该原型的基础上，逐渐完成整个系统的开发 工作。

不带反馈环

增量模型：（渐增模型）：分批逐步向用户提交产品——较短时间内向用户提交可完成部分工作的产品；逐步增加产品功能让用户适应By cjy

螺旋模型：每个阶段之前都增加了风险分析过程。

喷泉模型：面向对象的软件过程模型之一。

第二章，可行性研究

可行性研究本质：

实质上要进行一次大大压缩简化了的系统分析和设计的过程。也就是在较高的层次上以较抽象的方式进行系统分析和设计阶段的过程， 从经济可行性、技术可行性、法律可行性和用户操作可行性等方面评价系统是否值得做，是否能做。

结构化开发方法(Structured Developing Method)：

数据流图描述了系统的逻辑结构，一个数据流图中的四个基本图形元素（数据源、数据流、数据存储和数据处理）

数据字典（DD）

数据字典是关于数据信息的集合，是数据流 图中所有元素严格定义的集合

数据字典有以下四类条目：数据项(数据流分量)、数据流、文件(数据存储)、基本加工(处理)。

1.数据流 ：要定义数据流图中的数据流就要用数据流条目。数据流条目给出了某个数据流的定义，通常是列出该数据流的各个组成数据项。

订货单(数据流条目)=配件号(数据项)+配件名+规格+数量+顾客名+地址；

数据元素组成数据的方式：顺序，选择，重复，可选。

数据流条目的描述内容：

（1）名称：数据流名。

（2）别名：数据流的另一个名字。

（3）简述：对数据流的简单描述和说明。

（4）组成：描述数据流由哪些数据项组成，使用如表3-1所示的描述符号来表示数据流的组成

2.数据项条目 ：

数据流的组成成员是数据项，数据项条目是不可再分解的数据单位，是组成数据流和数据存储的最小元素。数据项条目的描述内容如下：

（1）名称：数据项名。

（2）别名：数据项的另一个名字

（3）简述：对数据项的简单描述

1. 注解：对数据项的其它补充说明。

必考的大题：

1. R图（一定）（椭圆形是属性【关键字】，矩形是实体，菱形加mn是联系可能会有属性）

数据流图（一定），数据字典（矩形是数据源与终点，圆形是数据处理，两行线是数据存储，箭头加名称是数据流）

（盒图）

特点：

(1)功能域(一个特定控制结构的作用域 )明确，可以从盒图上一眼就看出来。

(2)不可能任意转移控制。

(3)很容易确定局部和全程数据的作用域。

(4)很容易表现嵌套关系，也可以表示模块 的层次结构。

过程设计语言(PDL)

也称为伪码，它是用正文 形式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。

McCabe——环形复杂度：节点+1、边-点+2

Halstend——运算符N1、操作数N2的总次数. 程序长度=N1+N2

流图与流程图的转换

类图（不明确）UML：属性名：类型名=初值{性质串}

关联：两个类之间的联系 直线，名字前后有黑三角指向

泛化（继承）：通用元素与具体元素 用空心三角形连线，顶角指着通用元素

依赖：两个元素之间的语义连接关系，不独立的依赖于独立的用带箭头的虚线

细化：B是在A上的更详细描述，B细化A，则由B到A的空心三角形的虚线

等价类（一定）以及测试用例【划分等价类，有效等价类以及无效等价类】

1.

|  |  |  |
| --- | --- | --- |
| 输入 | 有效等价类 | 无效等价类 |

2.

|  |  |  |
| --- | --- | --- |
| 输入数据 | 测试范围 | 期望结果 |

3.

|  |  |  |
| --- | --- | --- |
| 输入数据 | 测试范围 | 期望结果 |

结构图（图中一个方框 代表一个模块，框内注明模块的名字或主要 功能；方框之间的箭头(或直线)表示模块的 调用关系。尾部是空心圆表示传递的是数 据，实心圆表示传递的是控制信息。）

层次图（框间的连线表示调用关系而不像层次方框 图那样表示组成关系。层次图中的一个矩形框代表一个模块,最顶层的方 框代表正文加工系统的主控模块，它调用下 层模块完成正文加工的全部功能；第二层的 每个模块控制完成正文加工的一个主要功能.）

HIPO（在H图(层次图)里除了 最顶层的方框之外，每个方框都加了编号。）

界面原型设计（画图）