

提纲

简答题(30分5,6个)分析与设计题(70分)

二、简答题：都是比较大的概念，软件的定义、过程模型及各个阶段、敏捷开发、构建和模块、耦合与内聚

三、分析设计题:比较活，需求分析、识别用例、用例描述、用例叙述、uml设计、建模、画图的元素要对(实线、虚线、形状)、多写字

软件

定义

1. 计算机软件是由专业人员开发并长期维护的工作产品。
2. 指令的集合，通过执行这些指令可以满足预期的特性、功能、性能需求。
3. 数据结构，使得程序可以合理利用信息。
4. 软件描述信息，以硬拷贝和虚拟形式存在，描述程序的操作和使用。

特点

过程模型及各个阶段

五种 框架活动

1. 沟通
2. 策划
3. 建模
4. 构建
5. 部署

各个阶段

1. 起始阶段

包括沟通、策划。识别基本的业务需求。建立业务模型和大概的需求分析。

2. 细化阶段

沟通、建模。完善模型和需求分析，做出分析设计以及相应的支持关键框架和服务代码，再反过来判断分析合理性。

3. 构建阶段

构建。主要完成代码实现工作和测试，完善相应配置工作。

4. 转换阶段

构建活动的后期阶段以及通用部署活动的第一部分。将其部署到实际使用环境，做好维护准备，应对可能的更新升级问题。

5. 生产阶段

部署。对持续使用的软件进行监控，提供运行环境的支持，提交并评估缺陷报告和变更请求。

五个过程阶段性并发运行，而不一定是顺序的。

敏捷开发

敏捷

普遍存在的变更是敏捷的动力，软件工程师必须加快步伐以适应出现的快速变更。

敏捷过程

具有 **可适应性**，利用 **增量式开发策略**，使得软件存在一定的可变性，适应变更的新的需求。

敏捷原则

1. 尽快交付软件
2. 需求会发生变化
3. 要提供软件增量，使用 **增量式开发策略**
4. 与利益相关合作
5. 开发成员在良好环境下工作和高效交流
6. 技术卓越和良好设计，还有简单性。
7. 首要目标是得到满足客户需求的可工作软件。
8. 团队工作的速度和方向是可持续的。
9. 能开发结构良好的体系结构，实现可靠的设计并达到客户满意度。
10. 反思如何围绕目标更有效地工作。

构建和模块

耦合与内聚

耦合性

耦合性也叫块间联系，指软件系统结构中各模块间相互联系紧密程度的一种度量。

模块之间联系越紧密，其耦合性就越强，模块的独立性则越差，模块间耦合的高低取决于模块间接口的复杂性，调用的方式以及传递的信息。

内聚性

内聚性也叫块内联系，指模块的功能强度的度量，即一个模块内部各个元素彼此结合的紧密程度的度量。内聚性是对一个模块内部各个组成元素之间相互结合的紧密程度的度量指标。

模块中组成元素结合的越紧密，模块的内聚性就越高，模块的独立性也就越高。理想的内聚性要求模块的功能应明确、单一，即一个模块只做一件事情。

模块的内聚性和耦合性是两个相互对立且又密切相关的概念。

耦合分类 耦合性由弱到强

1. 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是通过主模块的控制和调用来实现的，这就是非直接耦合。这种耦合的模块独立性最强。

2. 数据耦合

如果一个模块访问另一个模块时，彼此之间是通过数据参数(不是控制参数、公共数据结构或外部变量)来交换输入、输出信息的，则称这种耦合为数据耦合。由于限制了只通过参数表传递数据，按数据耦合开发的程序界面简单、安全可靠。因此，数据耦合是松散的耦合，模块之间的独立性比较强。在软件程序结构中至少必须有这类耦合。

3. 印记耦合

如果一组模块通过参数表传递记录信息，就是标记耦合。事实上，这组模块共享了这个记录，它是某一数据结构的子结构，而不是简单变量。这要求这些模块都必须清楚该记录的结构，并按结构要求对此记录进行操作。在设计中应尽量避免这种耦合，它使在数据结构上的操作复杂化了。如果采取“信息隐蔽”的方法，把在数据结构上的操作全部集中。

4. 控制耦合

如果一个模块通过传送开关、标志、名字等控制信息，明显地控制选择另一模块的功能，就是控制耦合。这种耦合的实质是在单一接口上选择多功能模块中的某项功能。因此，对所控制模块的任何修改，都会影响控制模块。另外，控制耦合也意味着控制模块必须知道所控制模块内部的一些逻辑关系，这些都会降低模块的独立性。

5. 外部耦合

一组模块都访问同一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，则称之为外部耦合。例如C语言程序中各个模块都访问被说明为extern类型的外部变量。外部耦合引起的问题类似于公共耦合，区别在于在外部耦合中不存在依赖于一个数据结构内部各项的物理安排。

6. 公共耦合

若一组模块都访问同一个公共数据环境，则它们之间的耦合就称为公共耦合。公共的数据环境可以是全局数据结构、共享的通信区、内存的公共覆盖区等。这种耦合会引起下列问题：

所有公共耦合模块都与某一个公共数据环境内部各项的物理安排有关，若修改某个数据的大小，将会影响到所有的模块。

无法控制各个模块对公共数据的存取，严重影响软件模块的可靠性和适应性。

公共数据名的使用，明显降低了程序的可读性。

公共耦合的复杂程度随耦合模块的个数增加而显著增加。若只是两个模块之间有公共数据环境，则公共耦合有两种情况。

若一个模块只是往公共数据环境里传送数据，而另一个模块只是从公共数据环境中取数据，则这种公共耦合叫做松散公共耦合。若两个模块都从公共数据环境中取数据，又都向公共数据环境里送数据，则这种公共耦合叫做紧密公共耦合。只有在模块之间共享的数据很多，且通过参数表传递不方便时，才使用公共耦合。否则，还是使用模块独立性比较高的数据耦合好些。

7. 内容耦合

耦合性最高，这是最差的设计

如果发生下列情形，两个模块之间就发生了内容耦合。

1. 一个模块直接访问另一个模块的内部数据；
2. 一个模块不通过正常入口转到另一模块内部；
3. 两个模块有一部分程序代码重叠（只可能出现在汇编语言中）
4. 一个模块有多个入口。

在内容耦合的情形，所访问模块的任何变更，或者用不同的编译器对它再编译，都会造成程序出错。好在大多数高级程序设计语言已经设计成不允许出现内容耦合。它一般出现在汇编语言程序中。这种耦合是模块独立性最弱的耦合。

内聚分类 内聚性由弱到强

1. 偶然内聚

模块中的代码无法定义其不同功能的调用。但它使该模块能执行不同的功能，这种模块称为巧合强度模块。

2. 逻辑内聚

这种模块把几种相关的功能组合在一起，每次被调用时，由传送给模块参数来确定该模块应完成哪一种功能。

3. 时间内聚

把需要同时执行的动作组合在一起形成的模块为时间内聚模块。

4. 过程内聚

构件或者操作的组合方式是，允许在调用前面的构件或操作之后，马上调用后面的构件或操作，即使两者之间没有数据进行传递。

5. 通信内聚

指模块内所有处理元素都在同一个数据结构上操作（有时称之为信息内聚），或者指各处理使用相同的输入数据或者产生相同的输出数据。

6. 顺序内聚

指一个模块中各个处理元素都密切相关于同一功能且必须顺序执行，前一功能元素输出就是下一功能元素的输入。即一个模块完成多个功能，这些模块又必须顺序执行。

7. 功能内聚

这是最强的内聚，指模块内所有元素共同完成一个功能，联系紧密，缺一不可。

耦合是影响软件复杂程度和设计质量的一个重要因素，在设计上我们应采用以下原则：如果模块间必须存在耦合，就尽量使用数据耦合，少用控制耦合，限制公共耦合的范围，尽量避免使用内容耦合。

内聚标志一个模块内各个元素彼此结合的紧密程度，它是信息隐蔽和局部化概念的自然扩展。内聚是从功能角度来度量模块内的联系，一个好的内聚模块应当恰好做一件事。它描述的是模块内的功能联系。耦合是软件结构中各模块之间相互连接的一种度量，耦合强弱取决于模块间接口的复杂程度、进入或访问一个模块的点以及通过接口的数据。程序讲究的是低耦合，高内聚。就是同一个模块内的各个元素之间要高度紧密，但是各个模块之间的相互依存度却要没那么紧密。

内聚和耦合是密切相关的，同其他模块存在高耦合的模块意味着低内聚，而高内聚的模块意味着该模块同其他模块之间是低耦合。在进行软件设计时，应力争做到高内聚，低耦合。

画图的元素

UML 图

初始状态



最终状态



中间状态

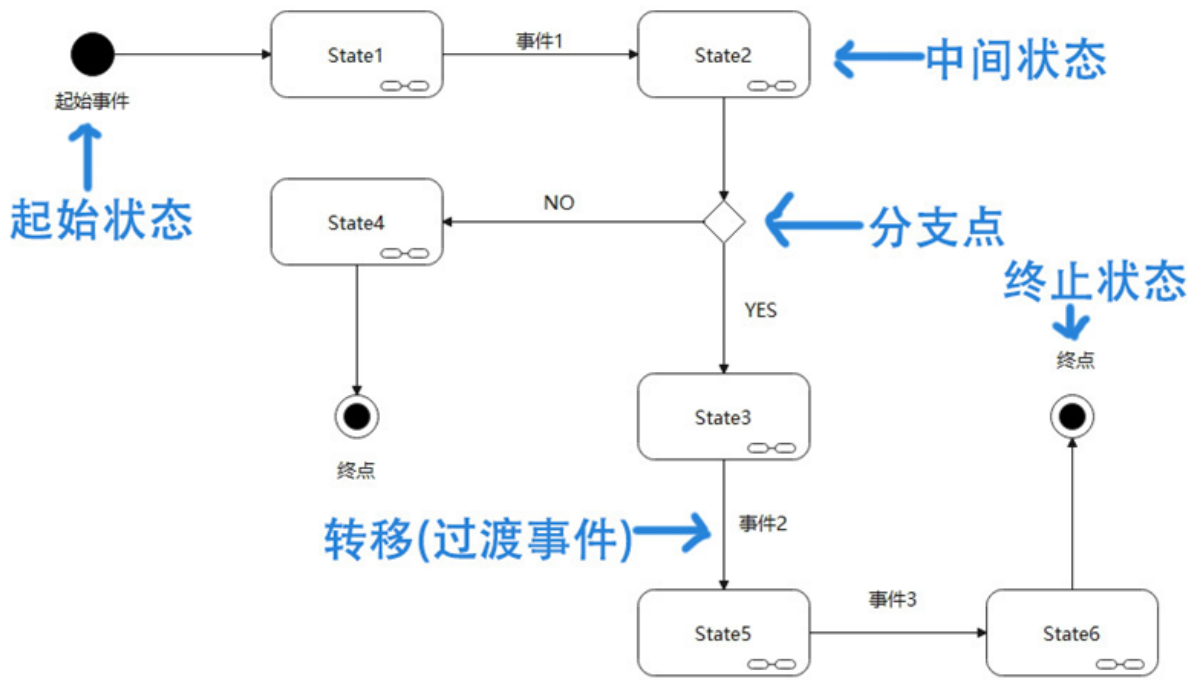


圆角矩形表示功能、方法、函数

箭头表示通过的流，上面写用户行为/事件/条件

菱形表示决策分支

实水平线表示并行发生的活动。



CSDN @跳探戈的龙虾

CRC 模型索引卡

类的说明，格式是

类：

说明

职责：

协作者：（在哪些类的协助下，这些类完成子功能）

类模型

类图由

1. 类名
2. 成员变量
3. 成员方法

组成

类之间用实线箭头连接。指向父类，箭头上标明关系：组成，从属，依赖，关联

活动图

流程图，包含决策（菱形，横线上写条件）、方框中包含方法、状态，横线上写条件/事件。从第一行中间开始往下。

顺序图

左上角开始，每列都是一个时期，纵向是同一时期发生的流程。横向是不同的时期。

状态图

类似过程图，以类为研究对象，方框中包含类的状态/方法，横线上是事件。

泳道图

把系统的不同组件（包括人）分隔开，按区域写流程图，对应功能写入对应区域。