2024-软件工程-作业3

一. 单选题（共22题，44分）

1. (单选题, 2分)

结构化设计方法(SD)与结构化分析方法(SA)一样，遵循( )的模型，采用自顶向下，逐步细化的技术。通常SD方法继续SA的工作，根据数据流图设计程序的结构。

A瀑布

B抽象

C快速原型

D实体-关系

2. (单选题, 2分)

( )把已确定的软件需求转换成特定形式的软件表示，使其得以实现。

A系统设计

B软件设计

C逻辑设计

D详细设计

3. (单选题, 2分) 以下（）不是总体设计环节的工作。

A确定系统的最佳实现方案

B结构设计，即确定系统由哪些模块组成，以及这些模块之间的关系

C过程设计，即确定每个模块的具体实现算法和使用的局部数据结构

D制定测试计划，在软件开发的早期阶段考虑测试问题，可以促使软件设计人员在设计时注意提高软件的可测试性

4. (单选题, 2分) 下面关于信息隐藏的描述中，错误的是（）

A信息隐藏是指把一些关系密切的元素物理地放得彼此靠近。

B信息隐藏使得一个模块内包含的信息对于那些不需要这些信息的模块来说，是不能访问的。

C信息隐藏隐藏的是模块的实现细节，而不是有关模块的一切信息。

D信息隐藏原理使得软件易于修改。

5. (单选题, 2分)

在进行软件模块结构设计时应当遵循的最主要的准则是( )。

A抽象

B模块化

C信息隐蔽

D模块独立

6. (单选题, 2分)

( )是数据说明、可执行语句等程序对象的集合，它是单独命名的并可通过名字访问。

A模块

B程序块

C数据块

D复合语句

7. (单选题, 2分)

模块( )，则说明模块的独立性越强。

A耦合越强

B耦合越弱

C扇入数越高

D扇入数越低

8. (单选题, 2分)

模块之间的信息可以做“控制信息”用，也可以做为( )用。

A数据

B控制流

C数据结构

D控制结构

9. (单选题, 2分)

( )是指把一些关系密切的软件元素物理地放置到彼此靠近的位置。

A内聚

B局部化

C信息隐蔽

D模块独立

10. (单选题, 2分)

为了提高模块的独立性，模块之间最好是( )。

A控制耦合

B公共耦合

C内容耦合

D数据耦合

11. (单选题, 2分) 模块内聚性用于衡量模块内部各成份之间彼此结合的紧密程度，一组语句在程序中多处出现，为了节省内存空间把这些语句放在一个模块中，该模块的内聚性是( )的

A功能内聚

B逻辑内聚

C信息内聚

D偶然内聚

12. (单选题, 2分) 模块内聚性用于衡量模块内部各成份之间彼此结合的紧密程度，将几个逻辑上相似的成分放在同一个模块中，通过模块入口处的一个判断决定执行哪一个功能，该模块的内聚性是（ ）的。

A逻辑内聚

B顺序内聚

C功能内聚

D时间内聚

13. (单选题, 2分) 模块内聚性用于衡量模块内部各成份之间彼此结合的紧密程度。模块中所有成分引用共同的数据，该模块的内聚性是（ ）的

A通信内聚

B过程内聚

C顺序内聚

D功能内聚

14. (单选题, 2分) 模块内聚性用于衡量模块内部各成份之间彼此结合的紧密程度，如果一个模块内的处理元素和同一个功能密切相关，而且这些处理必须顺序执行（通常一个处理元素的输出作为下一个处理元素的输入），则该模块的内聚度是（）。

A逻辑内聚

B过程内聚

C顺序内聚

D功能内聚

15. (单选题, 2分) 模块内聚性用于衡量模块内部各成份之间彼此结合的紧密程度，模块中所有成份结合起来完全一项任务，该模块的内聚性是( ）的。它具有简明的外部界面，由它构成的软件易于理解、测试和维护。

A逻辑内聚

B过程内聚

C顺序内聚

D功能内聚

16. (单选题, 2分)

在多层系统结构图中，其模块的层次数称为结构图的( )。

A宽度

B深度

C粒度

D控制域

17. (单选题, 2分) 下面关于扇出的描述中，错误的是（）

A扇出是一个模块直接控制（或调用）的模块数目。

B扇出过大表示模块过于复杂，管理和协调了过多的下级模块，可以通过适当增加中间层控制模块来降低扇出。

C扇出过小时可以把下级模块进一步分解成若干个子功能模块，或者合并到其上级模块中。

D在设计的较好的软件结构中，通常顶层扇出比较小，中间层的扇出比较大。

【设计良好的软件结构，通常顶层扇出比较大，中间扇出小，底层模块则有大扇入。】

18. (单选题, 2分)

( )着重反映的是模块间的隶属关系，即模块间的调用关系和层次关系。

A数据流图

B程序流程图

C系统结构图

D实体关系图

19. (单选题, 2分) Word、Excel等应用系统适合采用（ ）结构风格。

A层次系统

B事件系统

C解释器

D管道-过滤器

20. (单选题, 2分) 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提供软件的完整描述

E选项B和D

F选项B、C和D

21. (单选题, 2分) 下面的说法（ ）是错误的。

A软件体系结构的最佳表示形式是一个可执行的软件原型

B软件体系结构描述是不同项目相关人员之间进行沟通的使能器

C良好的分层体系结构有利于系统的扩展与维护

D设计模式是从大量成功实践中总结出来且被广泛公认的实践和知识

【解析】软件体系结构的最佳表现形式是一个清晰的架构设计，包括结构图、文档和相关说明，而不仅仅是一个可执行的软件原型

22. (单选题, 2分) 软件体系结构风格代表了软件体系结构设计中的惯用模式，在下面几种体系结构风格中，（ ）支持基于抽象程度递增的系统设计。

A解释器风格

B过程控制风格

C层次系统风格

D黑板风格

二. 多选题（共4题，12分）

23. (多选题, 3分) 关于软件设计的描述中正确的是（）。

A软件设计就是程序设计，即给出各个模块的具体实现算法。

B软件设计是软件开发过程中承前启后的工作，它依据软件需求规格说明书建立软件设计方案，作为下一步程序编码的依据。

C是在软件开发中形成质量的地方：设计提供了可用于质量评估的软件表示。

D是将需求准确转换为完整的软件产品或系统的唯一办法。

24. (多选题, 3分) 下面关于概要设计和详细设计的描述中，正确的是（）

A概要设计又称为总体设计，其基本目的是回答“概括地说，系统应该如何实现？”这个问题。

B详细设计的目的是要回答“怎样具体地实现所要求的系统”这个问题。

C概要设计将软件需求转化为数据结构和软件的系统结构，即系统的模块划分。

D详细设计通过对系统的结构表示（每个模块的内部工作）进行细化，得到软件的详细的数据结构和算法。

25. (多选题, 3分) 下面关于控制耦合的描述中，正确的是（）

A、如果两个模块间传递的信息是控制信息，即模块A通过向模块B发送一个控制变量，模块B根据该控制变量的值决定在多个功能中执行哪一个，这种情况下模块A和B之间存在控制耦合。

B、在控制耦合下，被调用模块B其实是一个单入口多功能模块，对模块B的任何改动都会影响其调用模块A。

C、可以通过适当的分解将这种控制耦合转换为数据耦合。

D、在控制耦合下，被调用模块B本身是一个逻辑内聚模块。

26. (多选题, 3分) 下面关于面向数据流的设计方法的描述中，正确的是（）

A、面向数据流的设计方法可以利用前面需求阶段得到的数据流图，按照一定的映射规则生成相应的软件结构图。

B、任何一个信息处理系统的信息流都可以看作是一个变换流，只有当其具有明显的“事务”特征（即有一个明显的事务中心）时，才按照事务分析的映射规则进行转换。

C、数据流图有可能全局特征是变换流，而局部区域属于事务流，反之易可，所以在进行向软件结构图的转换过程中要区分全局特征和局部特征。

D、按照变换分析或事务分析从数据流图出发转换得到的软件结构图就是最终的软件总体设计结果，不需要再进行任何的调整和优化。

三. 判断题（共8题，16分）

27. (判断题, 2分)

设计是将问题转化成解决方案的创造性活动。

A对

B错

28. (判断题, 2分) 内聚是指构建一个组件时所用的内部“粘合剂”。

A对

B错

29. (判断题, 2分)

如果两个组件之间有大量的依赖关系，我们说它们是松散耦合的。

A对

B错

【大量的依赖关系表示紧密耦合，而不是松散耦合】

30. (判断题, 2分) 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。这种耦合与外部耦合的区别是：外部耦合的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。因此公共耦合比外部耦合的耦合度要高。

A对

B错

31. (判断题, 2分) 层次图方框间的连线表示组合关系，即上层模块由下层模块组成。

A对

B错

【层次图方框间的连线表示调用关系，而不是组合关系】

32. (判断题, 2分) 变换分析得到的结构图是一个三分支结构，即包括输入部分、变换中心部分和输出部分；而事务分析得到的结构图是一个二分支结构，即一个接收分支和一个发送分支。

A对

B错

33. (判断题, 2分) 管道／过滤器风格支持并行执行。

A对

B错

34. (判断题, 2分) 软件体系结构是软件需求活动的一种工作产品。

A对

B错

【软件体系结构是设计活动的一种工作产品】

四. 简答题（共7题，28分）

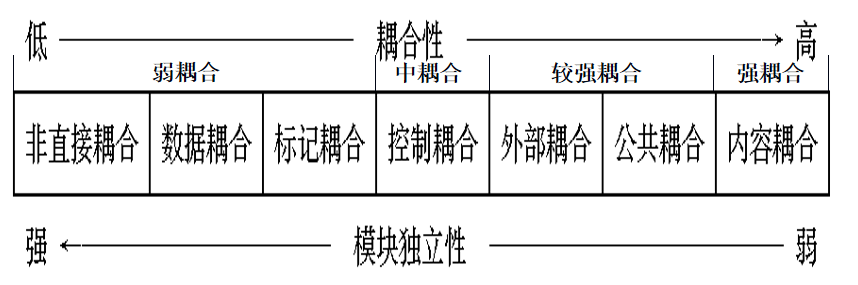
35. (简答题, 4分)

请将下述有关模块独立性的各种模块之间的耦合形式，按其耦合度从低到高排列起来。

① 内容耦合 ② 控制耦合 ③ 非直接耦合 ④ 标记耦合

⑤ 数据耦合 ⑥ 外部耦合 ⑦ 公共耦合

③非直接耦合<⑤数据耦合<④标记耦合<②控制耦合<⑥外部耦合<⑦公共耦合①内容耦合



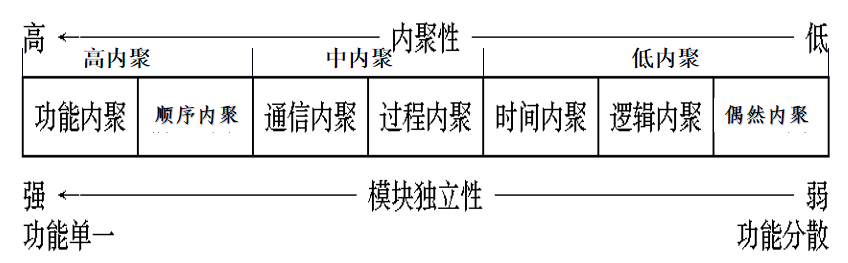
36. (简答题, 4分)

请将下述有关模块独立性的各种模块内聚，按其内聚度（强度）从高到低排列起来。

① 偶然内聚 ② 时间内聚 ③ 功能内聚 ④ 通信内聚

⑤ 逻辑内聚 ⑥ 顺序内聚 ⑦ 过程内聚

③功能内聚>⑥顺序内聚>④通信内聚>⑦过程内聚>②时间内聚>⑤逻辑内聚>①偶然内聚



37. (简答题, 4分)

从下列关于模块化程序设计的叙述中选出5条正确的叙述。

① 程序设计比较方便，但比较难以维护。

② 便于由多个人分工编制大型程序。

③ 软件的功能便于扩充。

④ 程序易于理解，也便于排错。

⑤ 在主存储器能够容纳得下的前提下，应使模块尽可能大，以便减少模块的个数。

⑥ 模块之间的接口叫做数据文件。

⑦ 只要模块之间的接口关系不变，各模块内部实现细节的修改将不会影响别的模块。

⑧ 模块间的单向调用关系叫做模块的层次结构。

⑨ 模块越小，模块化的优点越明显。一般来说，模块的大小都在10行以下。

② 便于由多个人分工编制大型程序。

③ 软件的功能便于扩充。

④ 程序易于理解，也便于排错。

⑦ 只要模块之间的接口关系不变，各模块内部实现细节的修改将不会影响别的模块。

⑧ 模块间的单向调用关系叫做模块的层次结构。

答案解析：如果程序结构的模块化满足评价的标准（高内聚，低耦合），这样的结构是容易编码，容易测试，容易理解，容易修改，容易维护的。程序的功能也容易扩充。特别适合于大型程序编制时，多人分工合作，协同完成任务的情形。 因为是采用自顶向下，逐层分解来划分模块结构的，所以模块之间的调用关系是分层次的模块结构，就叫做模块的层次结构。模块之间的信息传递叫做模块的接口，模块之间传递信息可以通过参数表、全局变量或全局数据结构、数据文件、专门的通信模块，不是专指数据文件。 划分模块时，模块大小要适中。模块太大，控制路径数目多、涉及的范围广、变量的数目多、总体复杂性高，可理解性、可修改性、可靠性就会变差。模块太小，模块个数增大，调用的系统开销就会增大。所以要有一个权衡。

38. (简答题, 4分)

解释模块的概念。举例说明软件结构中的扇出、扇入、深度、宽度的意义。

模块是可单独命名和可编址的部分

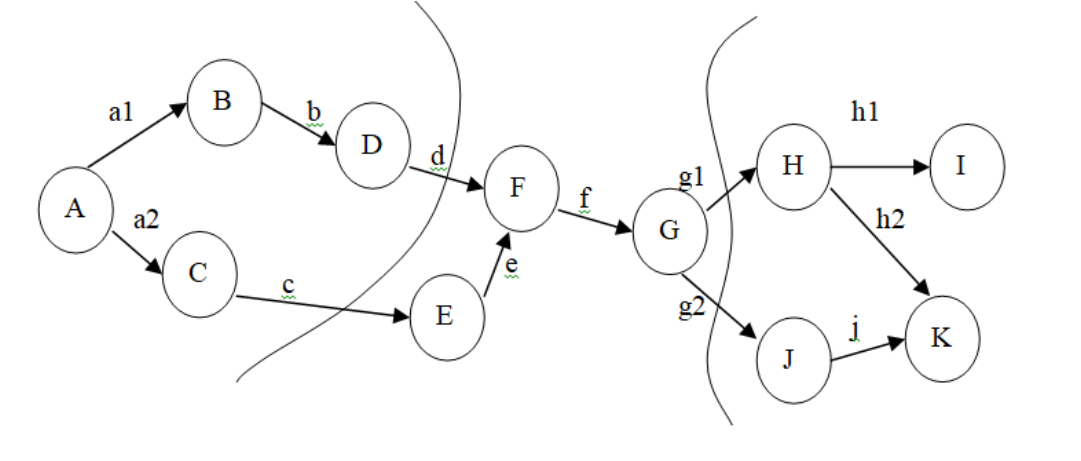
深度：软件结构中控制的层数。一般而言它与系统的复杂度和系统大小直接对应。

宽度：软件结构中同一个层次上的模块总数的最大数。

扇出：一个模块直接控制（调用）的模块数目。扇出过大说明模块过分复杂；过小也不好，不利于系统平衡分解，3到9为宜。

扇入：一个模块的扇入是指直接控制该模块的模块数目。扇入越大说明共享该模块的上级模块越多。

1. (简答题, 4分) 把下面的数据流图用SD方法映射成软件结构图，要求写出映射步骤和并画出软件结构图，并在图中标出模块调用时的参数和返回值。



第1步 复查基本系统模型。

第2步 复查并精化数据流图

第3步 确定数据流图具有变换特性还是事务特性

第4步 确定输入流和输出流的边界，从而孤立出变换中心

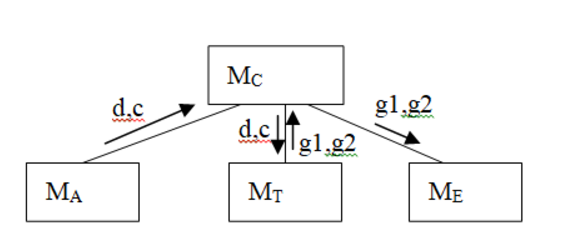
第5步 完成“第一级分解(first level factoring)”

第6步 完成“第二级分解”

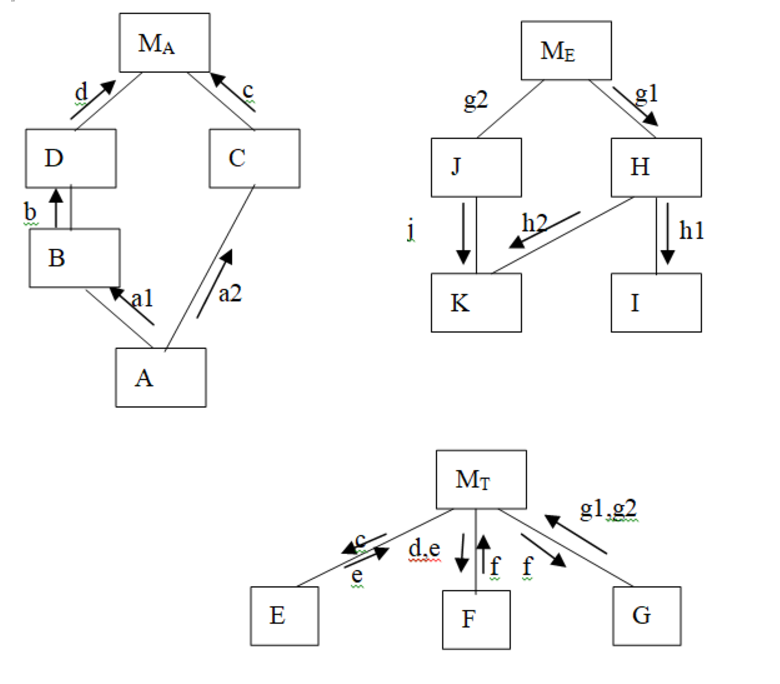
第7步 使用设计度量和启发式规则对第一次分割得到的软件结构进一步精化

变换分析：

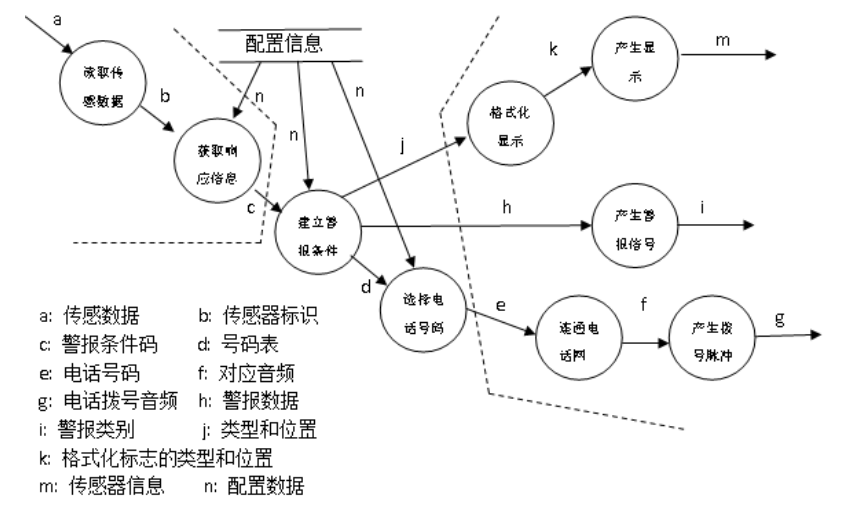
一级分解结果：



二级分解结果：



1. (简答题, 4分) 把下面的数据流图用SD方法映射成软件结构图，要求写出映射步骤和并画出软件结构图，并在图中标出模块调用时的参数和返回值（用字母表示）。



映射步骤如下：

第1步 复查基本系统模型。

第2步 复查并精化数据流图。

第3步 确定数据流图具有变换特性还是事务特性。

第4步 确定输入流和输出流的边界，从而孤立出变换中心。

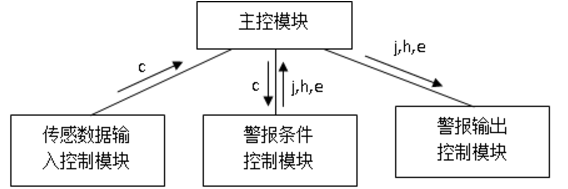
第5步 完成“第一级分解”

第6步 完成“第二级分解”。

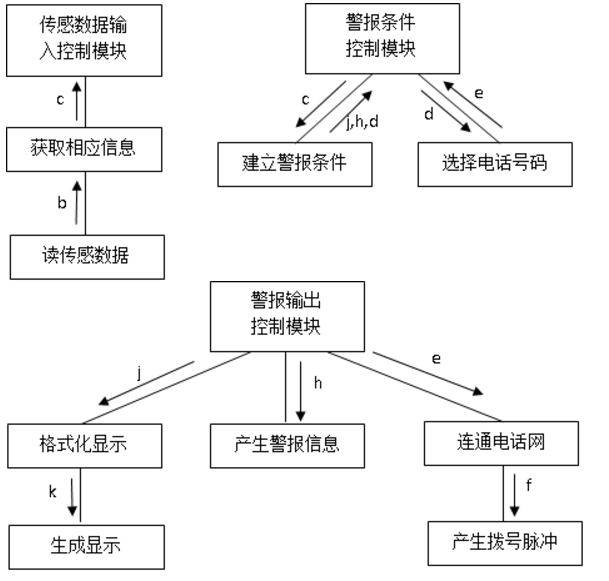
第7步 使用设计度量和启发式规则对第一次分割得到的软件结构进一步精化。

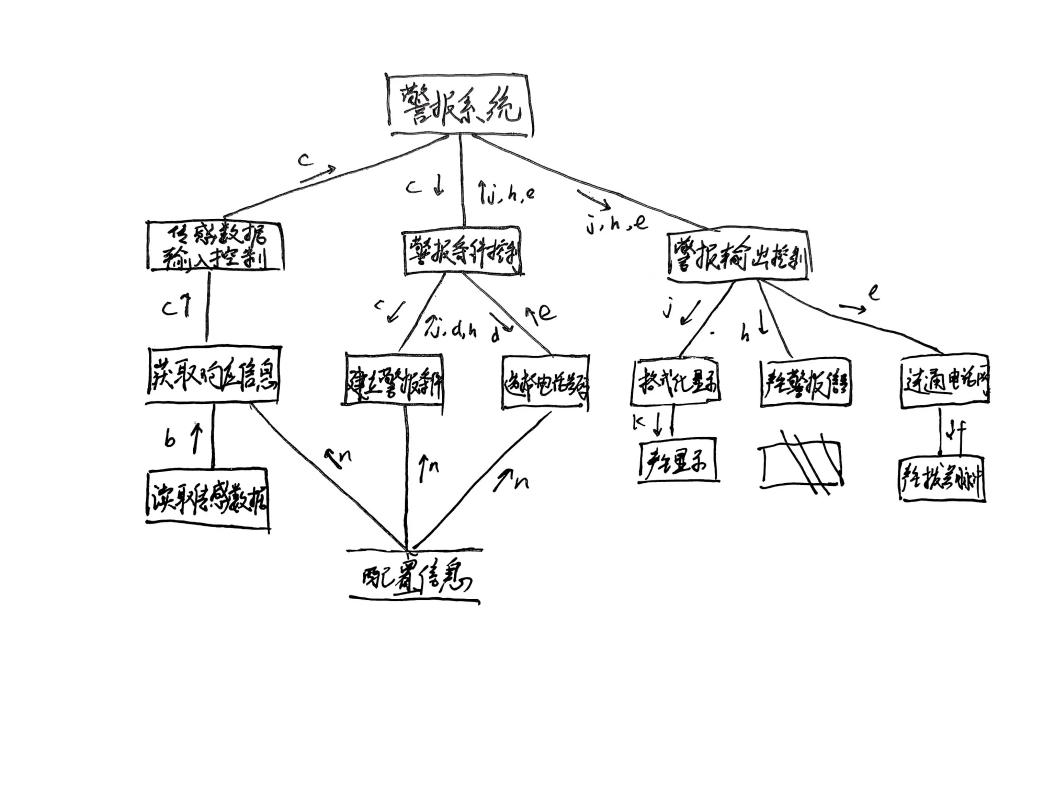
从图中可见该数据流图是变换型。

首先进行一级分解，结果如下图所示：



然后进行二级分解：结果为：





1. (简答题, 4分) 何谓隐式调用体系结构风格？该风格有何显著特点？

隐式调 用风格的思想是构件不直接调用一个过程，而是发布或广播一个或多个事件。系统中的其它构件为它感兴趣的事件注册过程。当一个事件被发布，系统自动调用在这个事件中注册的所有过程，这样，一个事件的发布就“隐式地”激发了另一模块中的过程。

（另一答案）隐式调用风格：这种风格的构件是一些模块，这些模块的接口既提供了一些过程，也有一些事件的集合。过程可以用通用的方式调用，构件也可以在系统事件中注册一些过程，当发生这些事件时，过程被调用。

该风格的特点：事件的发布者并不知道哪些构件会被这些事件影响。这样不能假定构件的处理顺序，甚至不知道哪些过程会被调用。。