

软件工程 试题(A)

学号	
姓名	

题号	一	二	三	四	五						总分
分数											

注：本试卷满分 100 分，按 50%折算计入课程成绩。

注
意
行
为
规
范

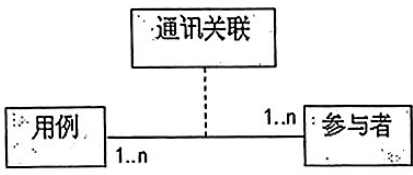
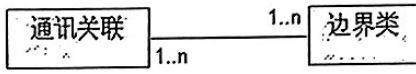

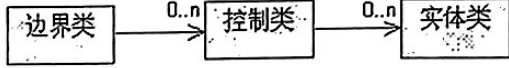
一 选择题 (20 分)

注意：请将选择题答案填入该表中，直接写在题目中的答案视为无效。

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20

- 以下各项软件工程活动中，不属于概念映射的是 A：
 - 根据数据字典描述，设计全局数据结构和文件存储结构
 - 根据从用户采集来的功能需求，进行用例模型与事件流的设计
 - 根据顾客需求描述，识别实体类并设计其属性
 - 根据领域类图，设计关系数据表的数据模式
- 以下关于各种软件开发过程模型的说法，不正确的是 C：
 - 原型法能够很大程度的帮助客户参与软件开发过程 ✓
 - RAD 模型通过在短时间内投入大量人力进行并行开发，以提高开发效率 ✓
 - 增量过程模型较之演化过程模型更能速响应用户需求的变化 ✗
 - 敏捷过程遵循“小步快跑，及时反馈”的原则，通过短周期迭代逼近目标 ✓
- 对以下各项需求的分类，不正确的是 D：
 - 餐饮服务的大部分环节通过网络进行，充分提高服务效率并降低成本——业务需求
 - 游客可浏览餐厅列表，并能查看餐厅的具体信息——功能需求 ✓
 - 在中午和晚上的高峰时期，网站可容纳不少于 200 个并发用户，任何请求均可在 3 秒钟之内得到响应——非功能需求
 - 顾客可通过支付宝进行在线支付，系统需与支付宝系统进行连接——设计约束
- 以下各项内容中， C 不是软件需求所能够起到的作用；
 - 作为客户与软件开发团队之间正式合同的一部分
 - 充分理解现实中的业务问题，并作为软件设计的基础
 - 为最终软件系统的集成测试提供标准和依据
 - 通过形成需求基线，来有效管理系统的演化与变更

主管
领导
审核
签字

5. 以下关于结构化分析的说法，不正确的是_____
- A) DFD 用于描述数据在系统中各逻辑功能模块之间流动和处理的过程，它是一种功能模型
- B) 0 层 DFD 描述了系统与外部环境之间的数据输入/输出关系
- C) DFD 中的“外部实体”定义了处于系统外部的人员、组织或其他软硬件系统等
- D) 如果数据无需永久存储，则它可以从一个加工直接流到另一个加工
6. 以下关于面向对象的说法，不正确的是_____ A
- A) “抽象类”是指所有方法均被定义为 abstract 类型的类
- B) “封装”是指对外界隐藏对象内部属性与操作的具体实现细节，仅对外暴露接口
- C) “多态”的一种表现是：在父类中定义的某个方法被不同子类继承后，可以表现出不同的行为
- D) 若两个类之间具有聚合关系，意味着二者在生命周期上是独立的
7. 如果使用类图来刻画用例建模中涉及到的相关概念，以下图形表达不恰当的是 B
- A) 
- B) 
- C) 
- D) 
8. 以下关于面向对象分析的说法，不正确的是 B
- A) 如果 actor 与系统之间的多次交互总是同时发生且不会单独发生，可合并为一个用例；
- B) actor 与 actor 之间的交互不应包含在 use case 中； 交互
- C) 除非系统需要管理和维护某一 actor 的信息，否则只需将该 actor 作为系统的用户即可，无需作为实体类在领域类图中出现；
- D) 若某一用例的事件流中存在类似于“某 actor 向系统提交...信息/指令”的步骤，那么在该用例对应的时序图中，该步骤反映为边界类向控制类/实体类发出的指令。
9. 在一个交通信息查询系统中，用户可查询到每条“公交线路”经过的多条“街道”、它并包含的多个“站点”、每个“站点”附近的多个“商户”。采用 OO 的思想对该需求进行分析，那么“站点”和“公交线路”之间、“站点”和“商户”之间的关系分别为 C
- A) 继承、依赖 B) 组合、关联 C) 聚合、关联 D) 关联、聚合
10. 对软件设计师而言，其所关注的非功能性需求通常不包括 C；
- A) 体系结构的可扩展性； B) 算法的时空复杂性；
- C) 按时交付性； D) 界面友好性与易用性；
11. 以下 A 不是基于事件的体系结构风格的必要元素；
- A) 事件 B) 事件源 C) 事件处理器 D) 事件管理器

12. 如果两个模块通过传递一个结构体参数进行调用，那么二者之间的耦合类型为 B
A) 特征耦合 B) 数据耦合 C) 控制耦合 D) 外部耦合
13. 如果某一模块内部包含的各功能之间通过数据发生关联，且前一个的输出数据为后一个的输入数据，那么该模块的聚合度类型为 B
A) 功能性内聚 B) 顺序性内聚 C) 通讯性内聚 D) 过程性内聚
14. 以下关于面向对象的设计的说法，不正确的是 D
A) OOD 是在 OOA 基础上的细化，且更加接近底层的技术实现；
B) OOA 关注的是做什么，OOD 关注的是怎么做
C) OOA 和 OOD 之间没有清晰的界限
D) 面向对象设计的关键是识别类
15. 在使用 UML 建立软件设计模型时，用来刻画软件系统在开发过程中的模块划分、模块组织、模块间依存关系的视图为 D
A) Class Diagram B) Sequence Diagram
C) Deployment Diagram D) Package Diagram
16. 以下各项内容中，不应出现在软件设计报告中的是 C
A) 软件系统在运行速度、可用性、响应时间等方面的要求
B) 软件系统的模块划分及其之间的连接关系
C) 软件系统各模块内部的处理算法
D) 用户界面的布局、风格、数据项、操作
17. 以下关于软件测试的说法，不正确的是 D
A) 在有限的时间内，黑盒测试无法检测出软件中存在的所有与顾客需求不一致的错误
B) 单元测试中的驱动模块用以模拟需要调用被测模块的其他模块
C) 自底向上的集成测试的缺点在于无法尽早对软件的宏观控制机制进行检验
D) 若采用自顶向下的集成测试，则测试过程中不需要或很少设计桩模块
18. 不属于确认测试的活动是 B
A) 黑盒测试 B) α 测试 C) β 测试 D) 白盒测试
19. 进行集成测试时主要依据 B
A) 需求规格说明 B) 概要设计文档 C) 详细设计文档 D) 用户使用说明书
20. 为满足软硬件环境和配置变化、或者数据环境的变化而做出的软件修改过程称为 B
A) 纠错性维护 B) 适应性维护 C) 完善性维护 D) 预防性维护

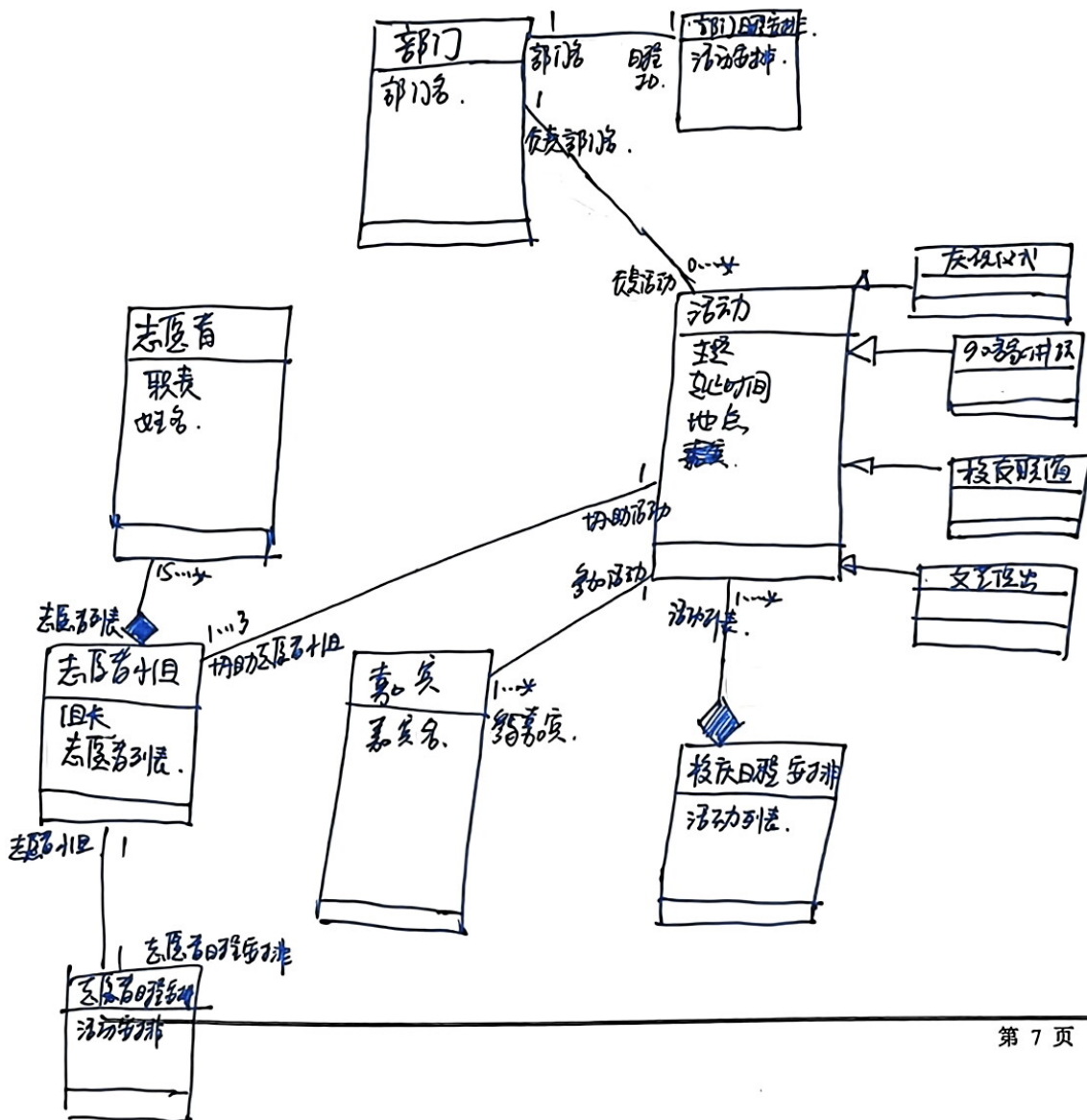
2. (12分) 以下是对哈工大90校庆信息管理系统的部分需求陈述：

该系统需要管理：1) 参与校庆的部门(department)及各自的职责；2) 参与校庆的志愿者(volunteer)及各自的职责；3) 校庆期间组织的各种庆祝活动(activity)及日程安排。4) 管理参加校庆的嘉宾(guest)。庆祝活动分为“90名家讲坛”(scholar_forum)、校友联谊(alumni)、文艺演出(performance)、庆祝仪式(celebration)四类。每项活动由1个部门负责组织，并由1-3个志愿者小组(volunteer_group)参与协助。每个志愿者小组至少包含15名志愿者，其中有一位组长(leader)。每名志愿者在活动中承担1项职责(responsibility)，一个志愿者可以参加多个小组并在多个活动中承担职责。活动的其他信息还包括主题(title)、起止时间(start_time, end_time)、地点(location)、嘉宾等，每个活动至少有1名嘉宾。将所有活动汇总起来，形成校庆日程安排(a_schedule)。针对每个部门和每个志愿者组织或参与的活动，形成相应的部门日程安排(d_schedule)与志愿者日程安排(v_schedule)。

根据上述陈述，识别出其中所有实体类，绘制领域类图。类图中标识出类的名字和关键属性、类间关系的多重性类和角色名。所有信息以中文形式给出。

所有的颜色作为

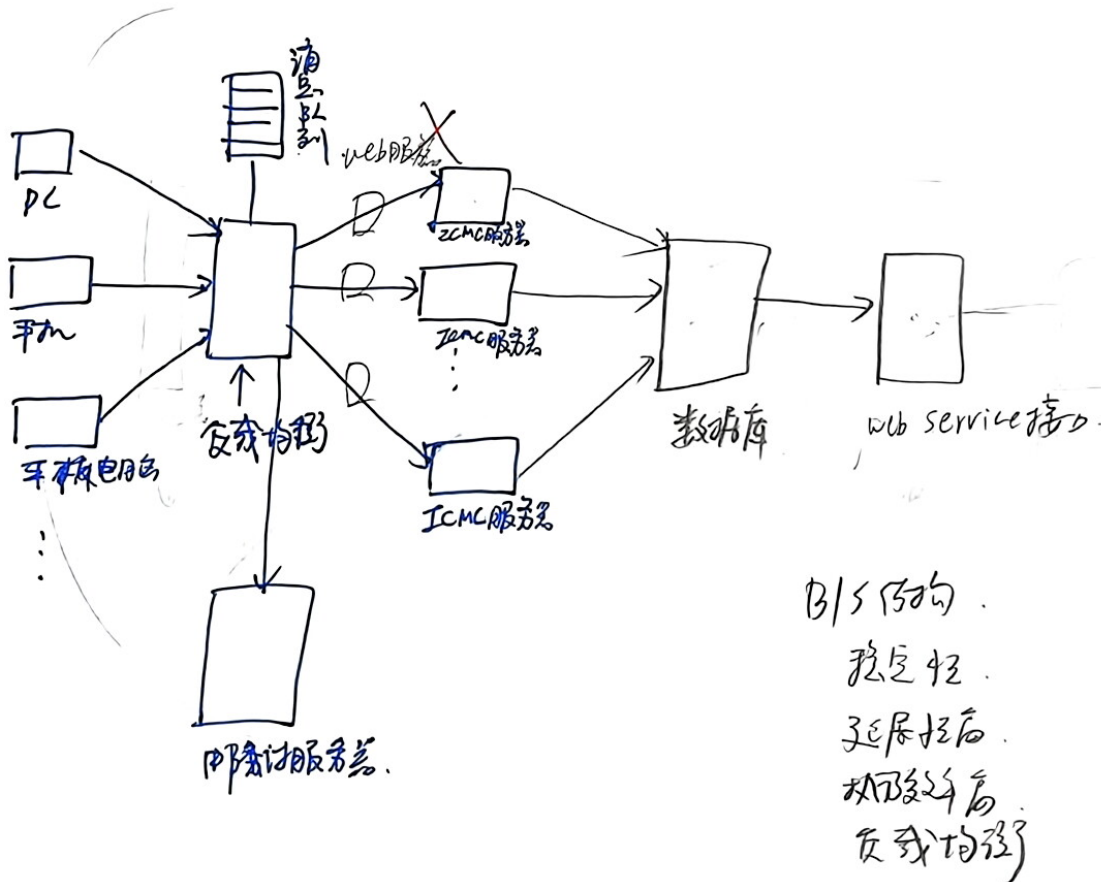
关系属性在加列对后属性中。

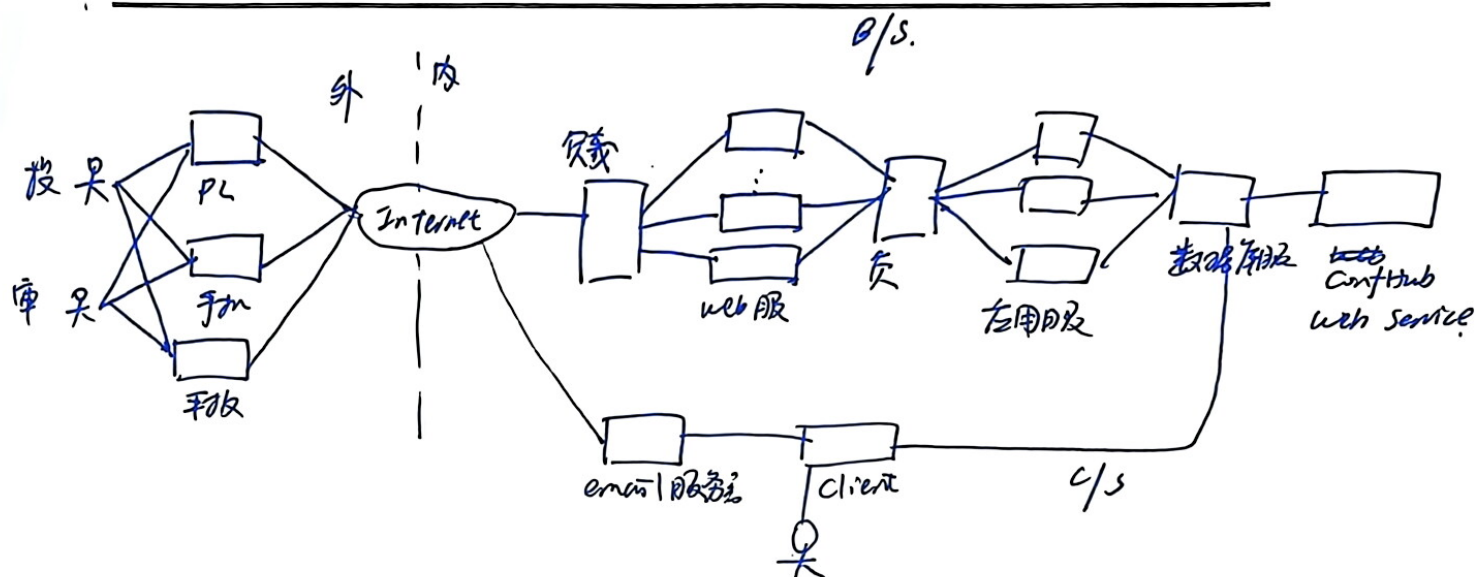


3. (8分) 某国际会议管理系统(ICMS)负责收集投稿人的论文, 并将论文分配给审稿人进行评审, 根据评审结果决定论文录用与否。该系统的若干非功能需求如下:

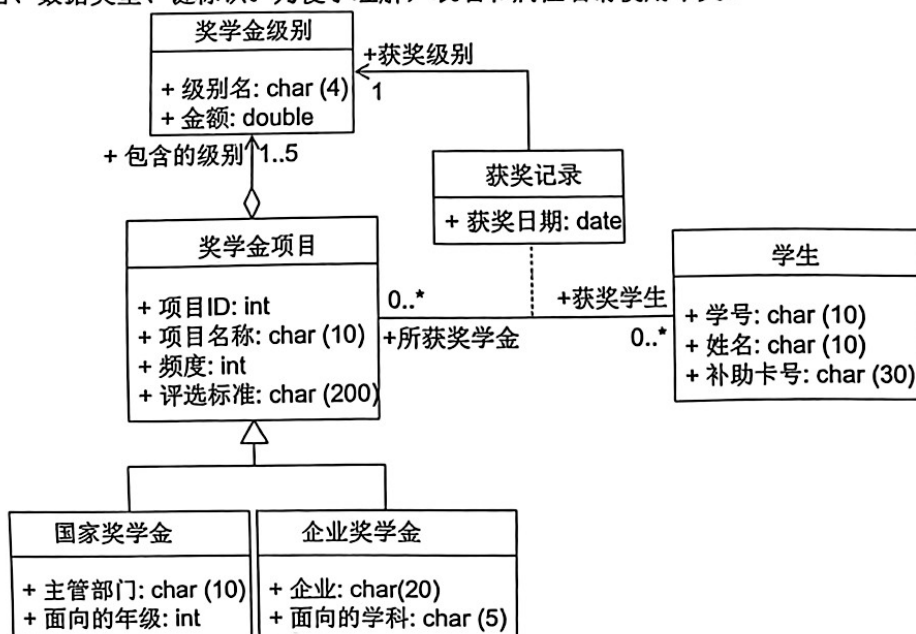
- (1) 投稿人和审稿人分布在世界各地, 可能使用 PC、手机、平板电脑等设备访问系统;
- (2) 由于投稿数量较大, 会议组织者需要频繁的访问该系统以进行稿件分配、决定论文录用与否、编排会议日程等工作, 故而对系统执行效率和响应速度的要求很高;
- (3) 会议截稿日期之前的最后 24 小时是投稿的高峰期, 系统的并发用户数量可能达到 1000 以上, 要保证此期间系统的高可用性和快速响应时间;
- (4) 投稿人、审稿人、组织者之间需要进行频繁的 email 交互(例如: 查询稿件情况、通知审稿结果等), 系统需要有单独的邮件服务器以确保邮件通讯的可靠性;
- (5) 由于筹备经费紧张, 组织者没有性能足够强大的服务器, 只有一批普通的服务器可用;
- (6) Internet 上有一个称为 ConfHub 的网站, 它收集在各地举行的国际会议信息并统一发布。该网站提供了一个 web service 接口, 允许各会议组织者将即将举行的会议信息发布到该网站上。因此, ICMS 需要与 ConfHub 连接起来, 自动将会议信息发布到 ConfHub 上。

请根据上述非功能需求, 为该系统选择恰当的分层体系结构模式(例如 C/S、B/S、混合结构、其它结构等), 简要绘制该系统的物理部署图。需说明你做出该体系结构设计的理由。





4. (9 分) 下图是哈工大学生奖学金发放系统的局部领域类图。该类图中所有实体类均需要持久化存储，且数据被存储于关系型数据库中。请为该系统设计关系数据模式，针对每一张表，给出表名、属性名、数据类型、键标识。为便于理解，表名和属性名请使用中文。



①

学生.

学号 char(10)

姓名 char(10)

补助号 char(30)

②

获奖记录.

~~获奖记录~~获奖记录ID Integer

获奖日期 date

所属奖项ID int获奖子生号 char(10).获奖级别 ~~等级~~ char(10).

③

奖学金级别

级别名 char(10)

金额 double

④

奖学金.

种类 Integer.

级别ID int

年度 int

评选标准 char(200)

主管部门 char(10)

面向年级 int

备注 char(20)

面向学科 char(5)

四 黑盒测试题 (12 分)

南非世界杯足球赛的门票中含有电子信息，验票模块可根据该信息自动检查是否是有效门票。若该电子信息的格式如下：

- 第 1-4 位是表示比赛日期的数字，第 1-2 位表示月(只能为 06 或 07)，第 3-4 位表示日(范围为 01-30)，但月和日组合起来只能在 6 月 10 日和 7 月 11 日之间。若违反该条规则，则输出“非法日期”；
- 第 5-7 位表示比赛地点，每一位都是大写字母；若违反，则输出“非法地点”；
- 第 8-12 位是表示球场座位的数字，其范围是[00001, 99999]；若违反，则输出“非法座位”；
- 若满足上述所有条件，则输出“合法门票”。

完成以下题目：

- (2 分)按结构化需求分析中的规范，写出“门票电子信息”的数据字典定义；
- (6 分)针对该验票模块，请使用等价类划分方法识别该模块的有效等价类和无效等价类；
- (4 分)针对等价类识别的结果，为该模块设计一组黑盒测试用例(包括输入数据和期望结果)。

- 2、
- | | |
|---------------------------------------|--|
| 有效 | 无效 |
| ①. 第 1-4 位组成日期在 6 月 10 日和 7 月 11 日之间 | ②. 第 1-4 位存在非数字 |
| | ③. 第 1-4 位组合形成 10 月 10 日组合在 6 月 10 日之后 |
| | ④. 第 1-4 位组合形成 10 月 10 日组合在 7 月 11 日之前 |
| ⑤. 5-7 位均为大写字母 | ⑥. 第 5-7 位存在非大写字母 |
| ⑦. 第 8-12 位均为数字
范围在 [00001, 99999] | ⑧. 第 8-12 位存在非数字 |
| | ⑨. 第 8-12 位组合数字 00000 |

用例	期望结果	覆盖等价类
3. ①. 0611ABC00001	合法门票	①. ⑤. ⑦
②. A611ABC00001	非法日期	②
③. 0511ABC00001	非法日期	③
④. 0723ABC00001	非法日期	④
⑤. 0611abc00001	非法地点	⑥
⑥. 0611ABCabcd	非法座位	⑧
⑦. 0611ABC00000	非法座位	⑨

五 白盒测试题 (18 分)

右侧给出了一个使用动态规划法求数组中最长递增子序列的算法代码。它输入一个整型数组 array, 输出该数组中包含的最长递增子序列的长度。例如在数组 {1, -1, 2, -3, 4, -5, 6, -7} 中, 其最长的递增子序列为 {1, 2, 4, 6}, 长度为 4, 那么该算法的输出为 4。

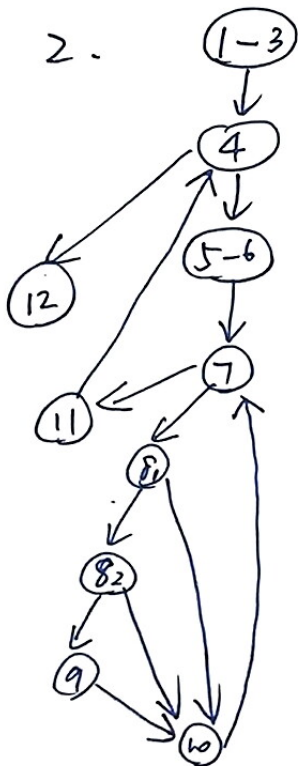
1. (3 分) 绘制程序流程图;
2. (3 分) 绘制控制流图;
3. (2 分) 计算其圈复杂度;
4. (5 分) 列出所有的基本路径;
5. (5 分) 在保证测试充分的情况下, 为该程序设计白盒测试用例。每个测试用例需包含两部分内容: 输入参数 array 的值、输出结果的期望值。
6. (选做, 附加 3 分) 使用(5)的测试用例对该程序进行测试, 给出每个测试用例的实际输出结果。若实际结果与期望结果不一致, 尝试指出代码的错误之处并修改。

```

int LIS (int [ ] array)
{
    1  int result = 0;
    2  int [ ] LIS = new int[array.Length];
        //array.Length为求数组长度的操作
    3  int i = 0;
    4  while (i < array.Length) {
    5      LIS[i] = 1;
    6      int j = 0;
    7      while (j < i) {
    8          if (array[i] >= array[j] && LIS[j]+1 > LIS[i])
    9              LIS[i] = LIS[j] + 1;
    10         j++;
    11     }
    12     i++;
    }
    return max(LIS); //max函数求得LIS中最大
        //元素, 可看作基本语句,
        //无需考虑内部实现
}

```

注意: 在完成上述题目时, 请尽量使用右图中标识出的程序行号。



3. 圈复杂度: 5. $13 - 10 + 2 = 5$

4. 基本路径:
- ① 1-3 → 4 → 12
 - ② 1-3 → 4 → 5-6 → 7 → 11 → 12
 - ③ 1-3 → 4 → 5-6 → 7 → 8 → 9 → 10 → 7 → 11 → 12
 - ④ 1-3 → 4 → 5-6 → 7 → 8 → 9 → 10 → 7 → 11 → 12
 - ⑤ 1-3 → 4 → 5-6 → 7 → 8 → 9 → 10 → 7 → 11 → 12