

一、概念

1. 软件：
2. 软件危机：
3. 计算机辅助软件工程：简称 CASE（Computer Aided Software engineering）将工具和代码生成器进行集成，为很多软件系统提供了可靠的解决方案
4. 软件工程
5. 软件工程管理学
6. 软件过程（software process）ISO9000 定义为：“将输入转化为输出的一组彼此相关的资源和活动”。
7. 软件生存周期
8. 瀑布模型
9. 软件问题定义：是指在对拟研发软件进行可行性分析和立项之前，对有关的主要需求问题进行初步调研、确认和描述的过程。
10. 结构化分析方法
11. 耦合
12. 内聚
13. 对象
14. 类
15. 继承
16. 封装
17. 软件质量。软件质量是与所确定的功能和性能需求的一致性；与所成文的开发标准的一致性；与所有专业开发的软件所期望的隐含特性的一致性。
18. 软件质量保证。软件质量保证是一个复杂的系统，它采用一定的技术、方法和工具，来处理 and 调整软件产品满足需求时的相互关系，以确保软件产品满足或超过在该产品的开发过程中所规定的标准。
19. 软件可靠性：表明了一个程序按照用户的要求和设计的目标，执行其功能的正确程度。
20. 软件质量保证

二、简答题

1. 软件危机主要表现。
2. 解决软件危机的措施
3. 软件工程三要素
4. 软件生存周期各阶段的主要任务有哪些？
5. 简述可行性分析的任务及内容
6. 技术可行性分析的内容
7. 社会可行性分析
8. 软件开发计划的内容包括哪几个方面？其主要内容具体有那些？
9. 需求分析的目的和重点是什么

10. 数据流图四种基本成分组成。
11. 什么是结构化分析方法?该方法使用什么描述工具?
12. 什么是数据流图?其作用是什么?
13. 简述软件概要设计的基本任务。
14. 什么是变换型数据流图?
15. 什么是事务型的数据流图?
16. 详细设计的基本任务?
17. 结构化程序设计方法的基本要点:
 - (1)采用自顶向下、逐步求精的程序设计方法。
 - (2)使用三种基本控制结构构造程序。任何程序都可由顺序、选择、重复三种基本控制结构构造。这三种基本结构的共同点是单入口、单出口。
18. 面向对象开发方法包括以下几个过程: (1)面向对象的分析(2)面向对象的设计(3)面向对象的实现。
19. 面向对象方法开发软件,通常需要建立三种形式的模型,它们分别是描述系统数据结构的对象模型、描述系统控制结构的动态模型和描述系统功能的功能模型。
 - (1) 对象模型
对象模型就是表示静态的、结构化的系统的“数据”性质。它是客观世界实体的对象以及对对象彼此间的关系的映射的模拟,描述了系统的静态结构。
 - (2) 动态模型
动态模型是表示瞬时的、行为化的系统的“控制”性质,该模型描述了系统的控制结构,它表示了瞬时的、行为化的系统控制性质,它关心的是系统的控制、操作的执行顺序,它从对象的事件和状态的角度出发,表现了对对象的相互行为。
它描述的系统属性是触发事件、事件序列、状态、事件与状态的组织。使用状态图作为描述工具。
 - (3)功能模型
功能模型表示变化的系统的“功能”性质,它指明了系统应该“做什么”,因此更直接地反映了用户对目标系统的需求。
功能模型由多张数据流图组成。数据流图说明数据流是如何从外部输入、经过操作和内部存储到外部的。
20. 提高软件质量和可靠性的技术有哪些?
21. 软件质量保证的主要任务是什么?
22. 良好的编码风格有哪几个方面?
23. 软件测试方法:一般分为动态测试方法与静态测试方法。动态测试方法中又根据测试用例的设计方法不同,分为黑盒测试与白盒测试两类。
 - 黑盒技术设计测试用例的方法一般有以下 4 种:
 - 等价类划分
 - 边界值分析
 - 错误推测
 - 因果图
 - 白盒测试:
 - (1) 逻辑覆盖:
 - (1)语句覆盖。语句覆盖是指使用足够多的测试数据,使被测试程序中每个语句至少执行一次。

(2)判定覆盖。判定覆盖是指设计出足够多的测试用例，使得被测程序中每个判定表达式都执行一次“真”和一次“假”的运行，从而使程序的每一个分支至少都通过一次，因此判定覆盖也称分支覆盖。

(3)条件覆盖。条件覆盖测试将使得判定表达式中每个条件的各种可能的值都至少出现一次。

(4)判定 / 条件覆盖。该覆盖标准指设计足够的测试用例，使得判定表达式中的每个条件的所有可能取值至少出现一次，并使每个判定表达式所有可能的结果也至少出现一次。

(5)条件组合覆盖。条件组合覆盖是比较强的覆盖标准，顾名思义，按此标准设计的测试用例，使得每个判定表达式中条件的各种可能的值的组合都至少出现一次。

(6)路径覆盖。路径覆盖就是要求设计足够多的测试数据，可以覆盖被测程序中所有可能的路径。

24. 软件测试的步骤。一般在软件产品交付使用之前要经过单元测试、集成测试、确认测试和系统测试。

单元测试主要针对模块的 5 个基本特征进行测试：模块接口、局部数据结构、重要的执行路径、错误处理、边界条件。

集成测试。集成测试是指在单元测试的基础上，将所有模块按照设计要求组装成一个完整的系统而进行的测试，故也称组装测试或联合测试。集成测试的方法主要有自底向上测试和自顶向下测试。

确认测试。确认测试又称有效性测试。它的任务是检查软件的功能与性能是否与需求说明书中确定的指标相符合。因而需求说明是确认测试的基础。

25. 软件维护有改正性维护、适应性维护、完善性维护和预防性维护。

改正性维护：我们知道，测试并不能发现所有的错误，因此必然有一部分隐含的错误在使用时才会被发现。对此类错误进行确定和修改的过程，就称为改正性维护。

适应性维护：随着计算机的飞速发展，计算机的软硬件环境在不断发生变化，硬件的性价比越来越高，操作系统的功能越来越强、越稳定、使用越来越方便。为了使应用软件适应这种变化而修改软件的过程称为适应性维护。

完善性维护：用户在使用软件的过程中，用户的工作流程、应用环境都会发生变化，因此会提出增加新的功能和改善性能的要求。这种增加软件功能、增强软件性能和提高软件运行效率而进行的维护活动称为完善性维护。

预防性维护： 为了提高软件的可维护性和可靠性而对软件进行的修改称为预防性维护。

26. 软件项目管理的内容

软件项目管理的具体内容包括对开发人员、组织机构、用户、文档资料等方面的管理。

软件开发人员一般分为：项目负责人、系统分析员、高级程序员、初级程序员、资料员和其他辅助人员。

组织机构要求有好的组织结构，合理的人员分工，有效的通讯。目前主要有主程序员、专家组、民主组织三种组织机构。

软件项目管理很大程度上是通过对文档资料的管理来实现的。因此要把开发过程中的一切初步设计、中间过程、最后结果建立成一套完整的文档资料。

27. 质量管理主要内容包括三个过程：质量计划制定、质量保证和质量控制，具体说明

28. 团队结构

主程序员团队：有一名技术能力出色的成员被指定为主程序员，主程序员负责领导团队完成任务

民主团队：因为没有集中的交流点，所以每个成员都可以发挥自己的能动性，能取得更高的士气和工作成就感

开放团队：为了创新而存在

28. 质量保障

在软件开发过程中，要监控和执行质量保障计划，在开发活动达到一个里程碑时，要及时根据质量保障计划进行质量验证

质量验证的主要方法有：

评审，评审又称为同级评审，现在是公认的质量保障最佳实践方法

分为六个阶段：规划阶段，总体部署阶段，准备阶段，审查会议阶段，返工阶段和跟踪阶段

在评审中发现问题是整个评审过程的关键阶段

测试

质量度量

29. 软件配置管理

配置管理：用技术和管理指导和监督方法，来表示和说明配置项的功能和物理特征，控制对这些特征的变更，记录和报告变更处理及其实现状态，并验证与需求规格的一致性

配置项：置于软件配置管理之下的软件配置的各种有关项目，包括各类管理文档，评审记录与文档，软件文档，源码及其可执行码，运行所需的系统软件和支撑软件以及有关数据等

基线：已经经过正式评审的规格说明或制品，可以作为进一步开发的基础，并且只有通过正式的变更控制过程才能变更

请问类与类之间通常有哪四种关系？分别简述之。

类与类之间的四种关系：关联、泛化(继承)、依赖和细化。

关联关系：表示两个类的对象之间存在某种语义上的联系。

泛化(继承)关系：通用元素和具体元素之间的一种分类关系。

依赖关系：描述两个模型元素（类、用例等）之间的语义连接关系

细化关系：对同一个事物在不同抽象层次上描述

CMM 的基本思想是什么？为什么要把能力成熟度划分成 5 个等级？

设计题

1. 已知有如下的伪代码程序：

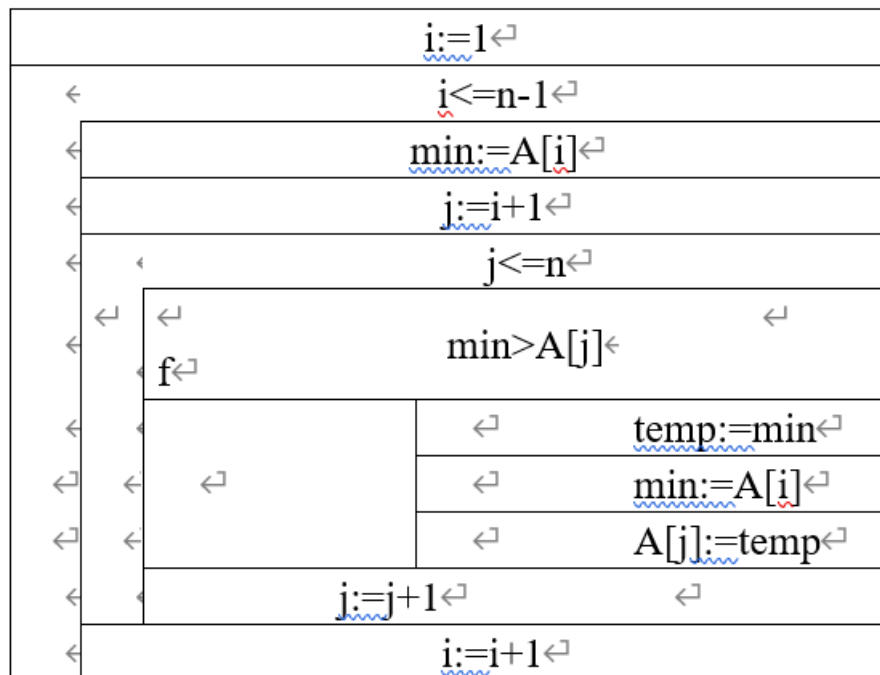
```
START
l:=1;
WHILE i<=n-1 DO
    min:=A[i];
```

```

j:=i+1;
WHILE j<=n DO
  IF min>A[j]
  THEN
    BLOCK
      temp:=min;
      min:=A[j];
      A[j]:=temp;
    ENDBLOCK
  ENDIF;
  j:=j+1;
ENDDO
i:=i+1;
ENDDO
STOP

```

画出系统的流程图或 NS 图



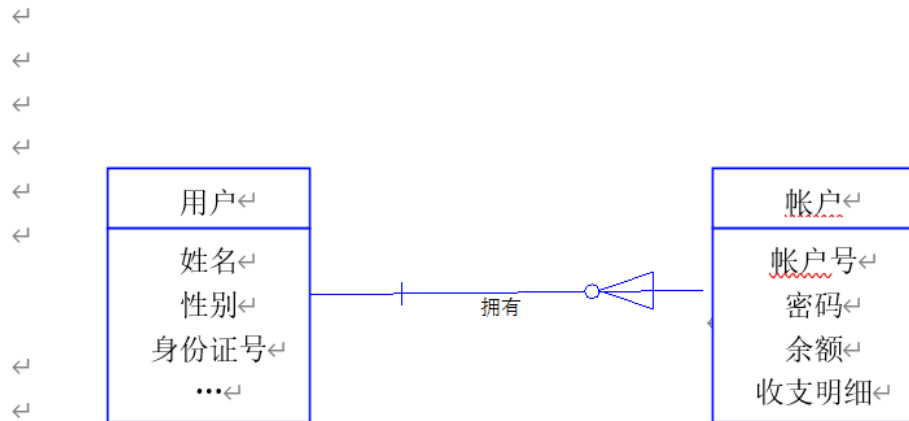
- 设计一个简化的网上个人银行查询系统，用户可以通过 Internet 查询自己帐户的收支明细、余额和修改密码。

请画出 E-R 图。

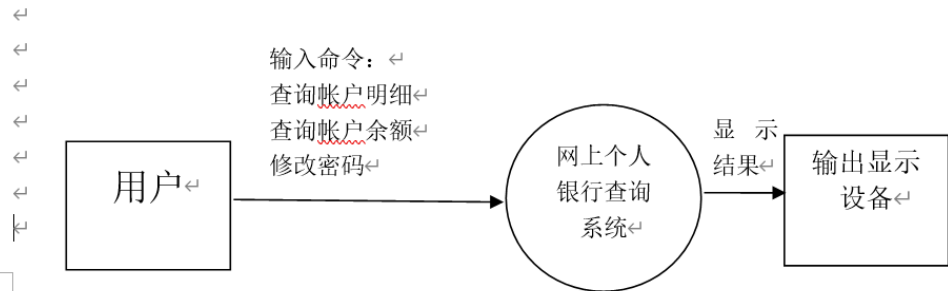
DFD 图的第 0 层和第 1 层。

编写两个关键词条的数据字典。

E-R 图

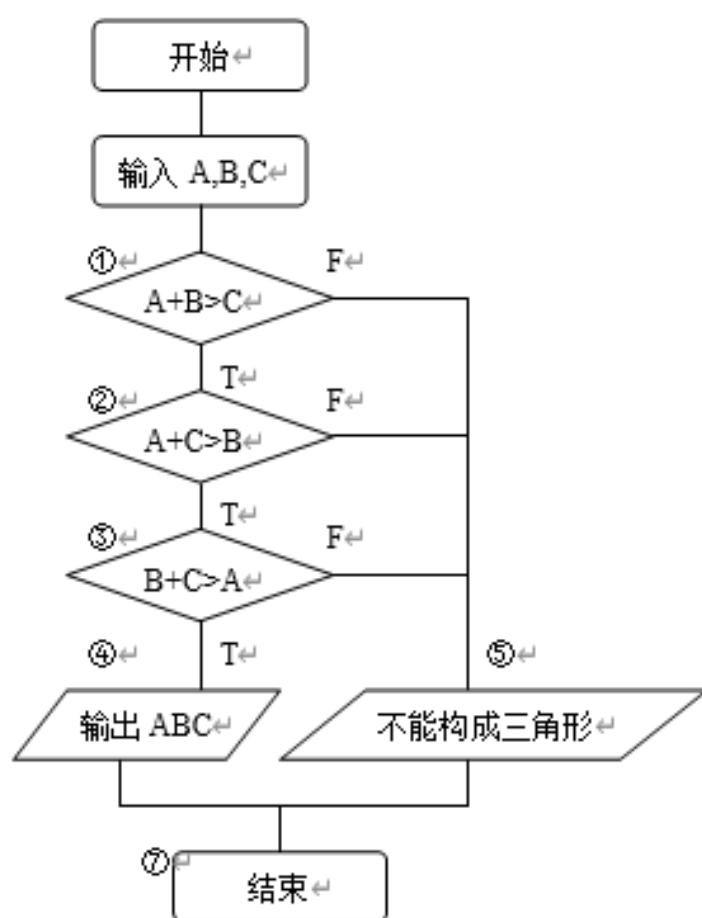


DFD 图第 0 层

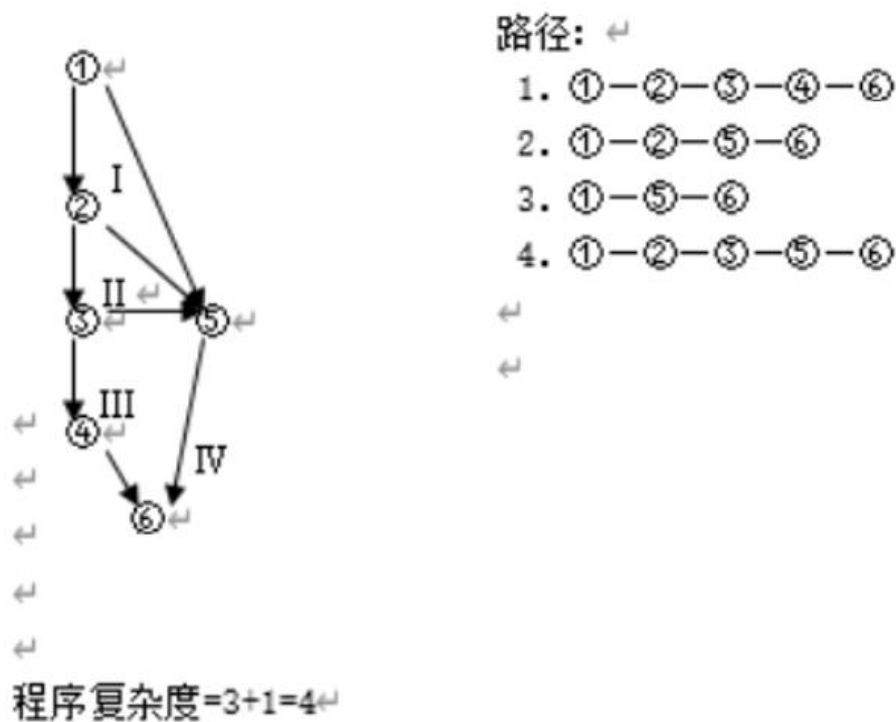


3. 输入三整数,判断是否构成三角形,如构成三角形,则输出三条边的值,否则输出”不能构成三角形”. 要求:1.用程序流程图表示该问题的算法; 2.计算程序复杂度; 3.设计路径覆盖的测试用例。

答: (1)



(2)



(3)

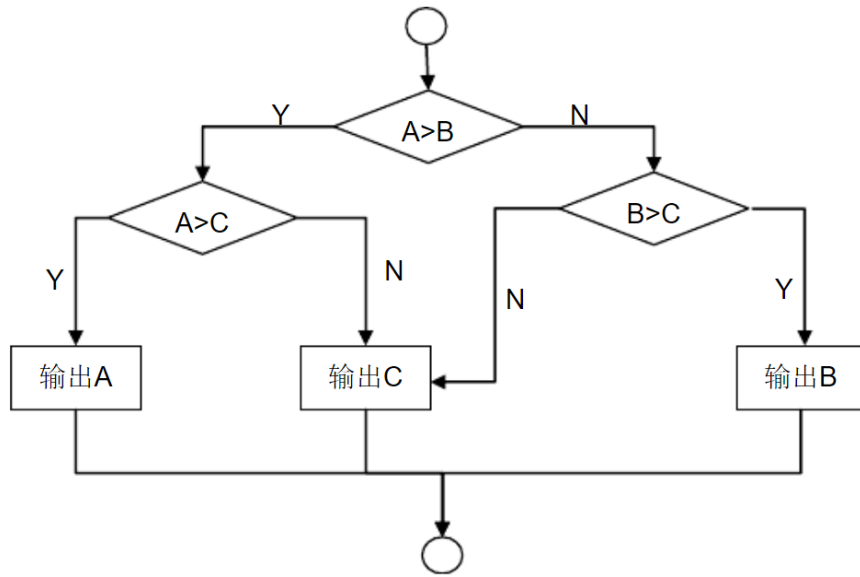
测试用例

路径	测试用例 (A, B, C)	结果
1	A=3, B=4, C=5	A=3, B=4, C=5
2	A=5, B=8, C=2	不能构成三角形
3	A=2, B=2, C=5	不能构成三角形
4	A=5, B=2, C=2	不能构成三角形

已知有一段代码实现了 “输出 ABC 三个数中的最大值”，要求完成下列问题：

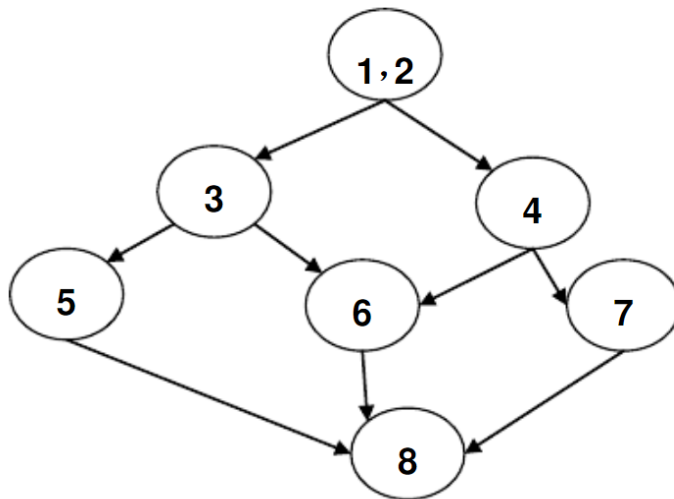
- 画出代码的流程图
- 根据该程序流程图画出该程序的流图；
- 计算该流图的环形复杂度；

(1) 流程图



https://blog.csdn.net/weixin_46232841

(2) 流图



https://blog.csdn.net/weixin_46232841

(3) 环形复杂度

$$V(G) = \text{区域数} = \text{边数} - \text{节点数} + 2 = \text{判定点} + 1 = 4$$

考务系统的数据流、数据字段（教材案例）

招聘系统的数据流图（教材案例）

图书管理系统的类图、用例图、时序图（教材案例）

医药管理系统的类图、用例图、时序图（教材案例）（ppt）

网上教学系统

The screenshot shows a web browser window with multiple tabs. The active tab is titled 'UML建模案例分析-网络教学系统UML建模'. The browser address bar shows the URL 'https://max.book118.com/html/2018/0607/171255290.htm'. The main content area displays a PPT slide with a blue background and white text. The slide title is 'UML建模案例分析' and the subtitle is '-网络教学系统UML建模'. The slide number '1 / 38' is visible at the bottom. The browser interface includes a search bar, navigation buttons, and a taskbar at the bottom showing the system clock as 10:38 on 2021/11/20.

UML建模案例分析-网络教学系统UML建模

1 / 38

UML建模案例分析-网络教学系统UML建模

10:38 2021/11/20

§1.1 系统功能需求

- 系统的功能需求主要包括以下几个方面：
- 学生可以登录网站浏览信息、查找信息和下载文件。
- 教师可以登录网站输入课程简介、上传课件文件、发布消息、修改和更新消息。
- 系统管理员可以对页面维护以及批准用户的注册申请。

UML 教学案例_百度网盘

UML建模案例:分析(网络教学系统).ppt

§2.2 系统的用例图

- 创建用例图之前首先需要确定参与者。
- 系统中的参与者主要有三类:
 - 教师
 - 学生
 - 系统管理员

UML建模案例:分析(网络教学系统).ppt

16 / 38

8°C 多云

2021/11/20

UML 教学案例_百度网盘

UML建模案例:分析(网络教学系统).ppt

1. 系统用户参与的总的用例图

```
graph LR
    User((User)) -.-> SL([System Login])
    Teacher((Teacher)) -.-> CI([Course Intro])
    Teacher -.-> UCAI([Upload CAI])
    Teacher -.-> MI([Message Issue])
    Teacher -.-> MU([Message Update])
    Student((Student)) -.-> D([Download])
    Student -.-> LTI([Look through info])
    Student -.-> AS([Artical Search])
    MU -.->|<<include>>| ID([Identity])
```

UML建模案例:分析(网络教学系统).ppt

16 / 38

8°C 多云

2021/11/20

UML建模案例(网络教学系统).ppt

2. 学生参与的用例图

The diagram shows a stick figure actor labeled 'Student' connected by dashed arrows to four use cases: 'Download', 'Look through info', 'Article Search', and 'Identity'. A dashed arrow labeled '<<include>>' points from 'Download' to 'Identity'.

UML建模案例(网络教学系统).ppt

17 / 38

UML建模案例(网络教学系统).ppt

3. 教师参与的用例图

The diagram shows a stick figure actor labeled 'Teacher' connected by dashed arrows to four use cases: 'Course Intro', 'Upload CAI', 'Message Issue', and 'Message Update'.

UML建模案例(网络教学系统).ppt

18 / 38

UML建模案例(网络教学系统).ppt

4. 系统管理员参与的用例图

The diagram shows an actor labeled 'Administrator' connected by dashed lines to three use cases: 'Page Maintenance', 'Process Registration', and 'Information Update'. There are also solid arrows pointing from 'CAI Process' to 'Page Maintenance' and from 'Information Update' to 'Page Maintenance'.

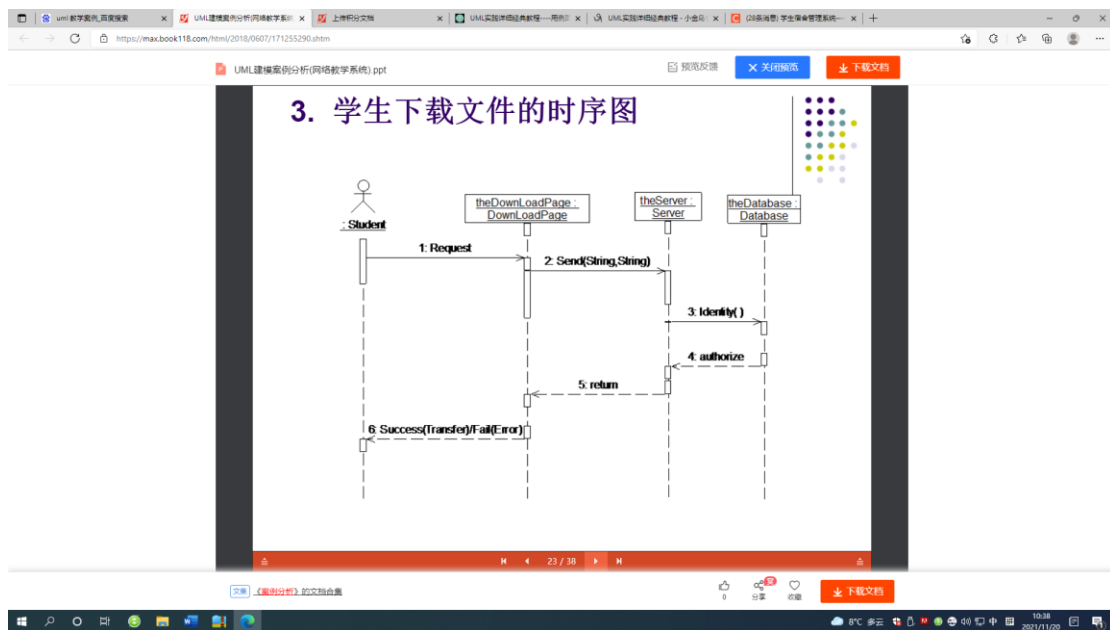
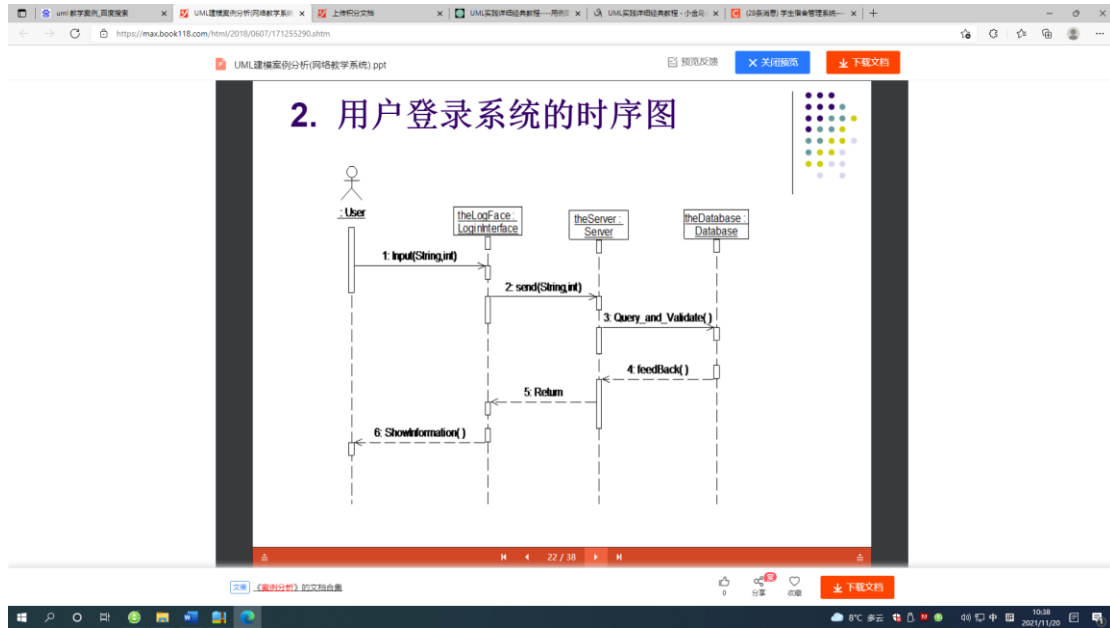
19 / 38

UML建模案例(网络教学系统).ppt

1. 系统管理人员管理网站的时序图

The sequence diagram involves four lifelines: Administrator, manage.. ManageInterface, theDatabase.. Database, and thePage.. Page. The interactions are as follows:
1. Administrator calls Login().
2. Administrator calls Add_or_delete_Article() on ManageInterface.
3. ManageInterface calls show() on Database.
4. Database returns success to ManageInterface.
5. Administrator calls UpdateCAI() on ManageInterface.
6. ManageInterface calls show() on Database.
7. Database returns success to ManageInterface.
8. Administrator calls Add_or_delete_User() on ManageInterface.
9. ManageInterface calls Notify() on Page.
10. Page returns success to ManageInterface.

21 / 38



UML建模案例(网络教学系统).ppt

1. 用户登录系统的活动图

```
graph TD
    Start(( )) --> InputURL([InputURL])
    InputURL --> ShowHomePage([Show HomePage])
    ShowHomePage --> InputLoginInfo([Input Login Information])
    InputLoginInfo --> PressOK([Press "OK" Button])
    PressOK --> CertifyUserInfo([Certify UserInfo])
    CertifyUserInfo -- Success --> ShowSuccessPage([Show Success Page])
    CertifyUserInfo -- Failure --> InputLoginInfo
    ShowSuccessPage --> End(( ))
```

The diagram illustrates the user login process. It starts with a start node leading to 'InputURL', which triggers 'Show HomePage'. This leads to 'Input Login Information', followed by 'Press "OK" Button'. The next step is 'Certify UserInfo', which has a decision point. If successful, it leads to 'Show Success Page' and then to the end. If it fails, it loops back to 'Input Login Information'.

UML建模案例(网络教学系统).ppt

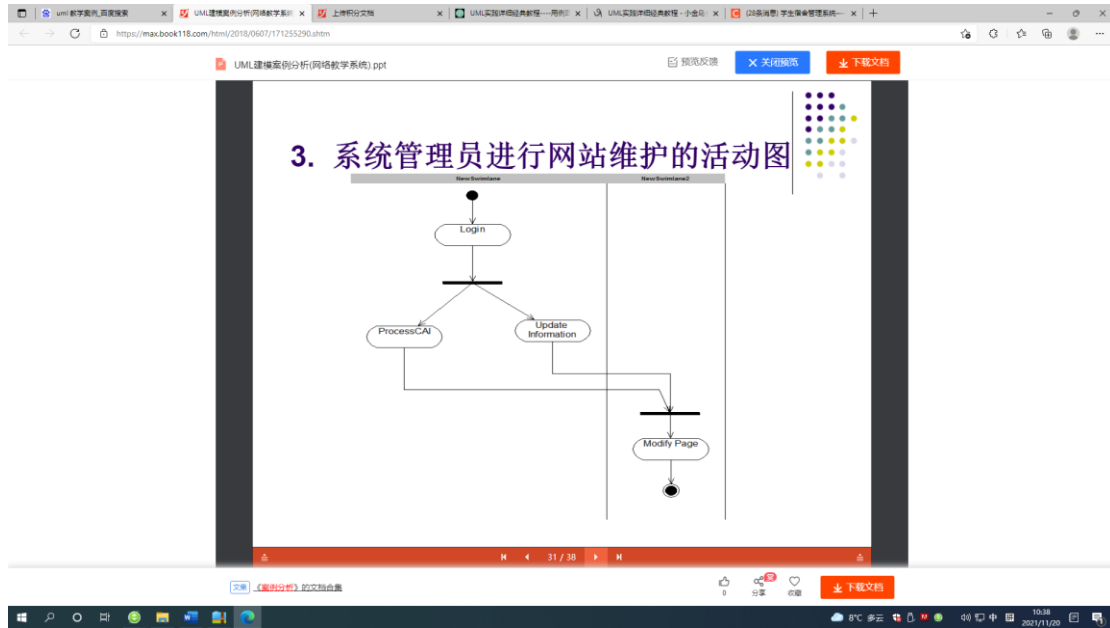
UML建模案例(网络教学系统).ppt

2. 教师上传课件的活动图

```
graph TD
    Start(( )) --> ApplyFileUpload([Apply FileUpload])
    ApplyFileUpload --> CertifySize([Certify Size And Other Aspect])
    CertifySize --> Decision1{ }
    Decision1 -- Error --> ApplyFileUpload
    Decision1 -- OK --> StoreFile([Store File])
    StoreFile --> AdminAuthorize([Administrator Authorize])
    AdminAuthorize --> Decision2{ }
    Decision2 -- Pass --> UpdatePage([Update Page])
    Decision2 -- Not Pass --> DeleteFile([Delete File])
    UpdatePage --> End(( ))
    DeleteFile --> End
```

The diagram illustrates the teacher upload courseware process. It starts with a start node leading to 'Apply FileUpload', which triggers 'Certify Size And Other Aspect'. This leads to a decision point. If the result is 'Error', it loops back to 'Apply FileUpload'. If the result is 'OK', it leads to 'Store File', then 'Administrator Authorize', and then another decision point. If the result is 'Pass', it leads to 'Update Page' and then to the end. If the result is 'Not Pass', it leads to 'Delete File' and then to the end.

UML建模案例(网络教学系统).ppt



UML 建模案例: 网络教学系统

(2) 一些其他的类

```
classDiagram
    class CourseIntro {
        +courseName
        +College
        +Teacher
        +scorePoint
        +Time
        +Place
        +teachingPlan
        +Modify()
    }
    class Article {
        +articleNum
        +articleTitle
        +teacherToIssue
        +create Time
        +content
        +Issue()
        +Delete()
        +Modify()
    }
    class FileUploadOrDownload {
        +fileName
        +fileType
        +fileSize
        +shortIntro
        +fileURL
        +Creator
        +createTime
        +CheckSize()
        +Modchy()
        +Store()
        +Cancle()
    }
```

CourseIntro

- courseName
- College
- Teacher
- scorePoint
- Time
- Place
- teachingPlan
- Modify()

Article

- articleNum
- articleTitle
- teacherToIssue
- create Time
- content
- Issue()
- Delete()
- Modify()

FileUploadOrDownload

- fileName
- fileType
- fileSize
- shortIntro
- fileURL
- Creator
- createTime
- CheckSize()
- Modchy()
- Store()
- Cancle()

2. 各个类之间的关系

```
classDiagram
    class CourseIntro {
        +courseName
        +College
        +Teacher
        +scorePoint
        +Time
        +Place
        +teachingPlan
        +Modify()
    }
    class Article {
        +articleNum
        +articleTitle
        +teacherToIssue
        +create Time
        +content
        +Issue()
        +Delete()
        +Modify()
    }
    class FileUploadOrDownload {
        +fileName
        +fileType
        +fileSize
        +shortIntro
        +fileURL
        +Creator
        +createTime
        +CheckSize()
        +Modchy()
        +Store()
        +Cancle()
    }
    class Teacher {
        +name
        +identityCardID
        +score
        +telephoneNum
        +Teachen()
        +UpdateCourseInfo()
        +UpdateTeachbookInfo()
    }
    class Student {
        +number
        +name
        +sex
        +age
        +classNum
        +grade
        +Student()
    }
    CourseIntro "1" -- "n" Teacher
    Article "1" -- "n" Teacher
    FileUploadOrDownload "1" -- "n" Teacher
    FileUploadOrDownload "1" -- "n" Student
```

CourseIntro

- courseName
- College
- Teacher
- scorePoint
- Time
- Place
- teachingPlan
- Modify()

Article

- articleNum
- articleTitle
- teacherToIssue
- create Time
- content
- Issue()
- Delete()
- Modify()

FileUploadOrDownload

- fileName
- fileType
- fileSize
- shortIntro
- fileURL
- Creator
- createTime
- CheckSize()
- Modchy()
- Store()
- Cancle()

Teacher (from Use Case View)

- name
- identityCardID
- score
- telephoneNum
- Teachen()
- UpdateCourseInfo()
- UpdateTeachbookInfo()

Student (from Use Case View)

- number
- name
- sex
- age
- classNum
- grade
- Student()