

作业 1:

一、简答题

1. UML 定义

UML(Unified Modeling Language 统一建模语言) 是为软件系统的制品进行描述 (specifying)、可视化 (visualizing)、构造 (constructing)、文档化 (documenting) 的一种语言。

2. UML 事物之间有哪 4 种关系(另外加上衍生出的聚合、组合), 及其符号是什么?

1. 关联 (Association)

一种结构化关系, 指一种对象和另一种对象有联系。给定关联的两个类, 可以从其中的一个类的对象访问到另一个类的相关对象

关联关系用一条实线表示 0..1 *

2. 依赖 (Dependency)

对于两个对象 X、Y, 如果对象 X 发生变化, 可能会引起对另一对象 Y 的变化, 则称 Y 依赖于 X

依赖关系用一条带箭头的虚线表示

3. 泛化 (Generalization)

定义了一般元素和特殊元素之间的分类关系, 与 C++ 及 JAVA 中的继承关系有些类似

泛化关系用一条带有空心箭头的实线表示

4. 实现 (Realization)

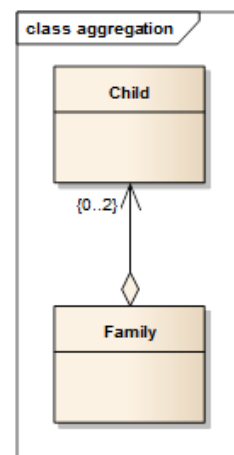
将一种模型元素 (如类) 与另一种模型元素 (如接口) 连接起来, 其中接口只是行为的说明而不是结构或者实现, 真正的实现由前一个模型元素来完成

实现关系一般用带有空心箭头的虚线表示

5. 聚合

和个体之间的关系, 即 has-a 的关系, 此时整体与部分之间是可分离的, 他们可以具有各自的生命周期, 部分可以属于多个整体对象, 也可以为多个整体对象共享。聚合关系是使用实例变量实现的. 从 java 语法上是分不出关联和聚合的。关联关系中两个类是处于相同的层次, 而聚合关系中两不类是处于不平等的层次, 一个表示整体, 一个表示部分。

聚合关系一般用带有空心四边形和箭头的实线表示



请注意不要雷同

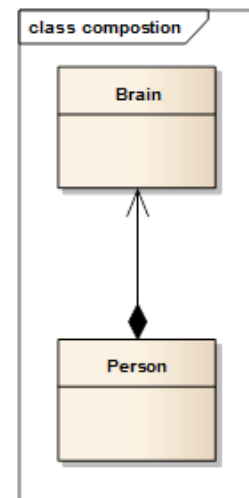
banban

<https://github.com/dream4789/Computer-learning-resources.git>

6. 组合

体现的是一种 contains-a 的关系，这种关系比聚合更强，也称为强聚合；同样体现整体与部分间的关系，但此时整体与部分是不可分的，整体的生命周期结束也就意味着部分的生命周期结束。组合跟聚合几乎相同，唯一的区别就是“部分”不能脱离“整体”单独存在，就是说，“部分”的生命期不能比“整体”还要长。

组合关系一般用带有实心四边形和箭头的实线表示



主要是关联关系的细化需要注意强弱，由弱到强分别是 依赖 < 关联 < 聚合 < 组合

3. UML4+1 视图

逻辑视图关注的静态元素是：层、子系统、类、接口，用类图来描述。关注的动态因素是协作关系，用时序图、协作图、状态图等来描述。是否需要在架构设计中体现类和类之间的关系？这取决于设计的层级。

开发视图关注的元素是程序包（SDK、解析器、中间件）、文件组织结构、编译依赖关系、目标单元（jar、exe、dll 等）。它和逻辑视图的静态元素通常有映射关系。

运行视图关注进程、线程、对象等运行时概念，以及相关的并发、同步、通信等问题。运行架构和开发架构的关系：开发架构一般偏重程序包在编译时期的静态依赖关系，而这些程序运行起来之后会表现为对象、线程、进程，运行架构比较关注的是这些运行时单元的交互问题。

部署视图关注“目标程序及其依赖的运行库和系统软件”最终如何安装或部署到物理机器，以及如何部署机器和网络来配合软件系统的可靠性、可伸缩性等要求。部署视图和运行视图的关系：运行视图特别关注目标程序的动态执行情况，而部署视图重视目标程序的静态位置问题；部署视图还要考虑软件系统和包括硬件在内的整个 IT 系统之间是如何相互影响的。

用例视图用例（场景）视图用来支持软件系统的需求分析，它定义系统的边界，关注的是系统的外部功能的描述。它从系统的使用者的角度，描述系统的外部的：静态的功能、动态行为。

4. 构造型

构造型用来定义一些特定于某个领域或某个系统的构造块。用来扩展建模元素，增加建模元素的语义。构造型表示对现有建模概念的精化，并且定义在 UML 的简档之中。简档是一组相关构造型、标记和约束的集合，其目的是使用 UML 专用于特定领域或平台。UML 预定义了大量构造型，同时允许用户自行定义构造型，因此，在 UML 中构造型提供了扩展机制。

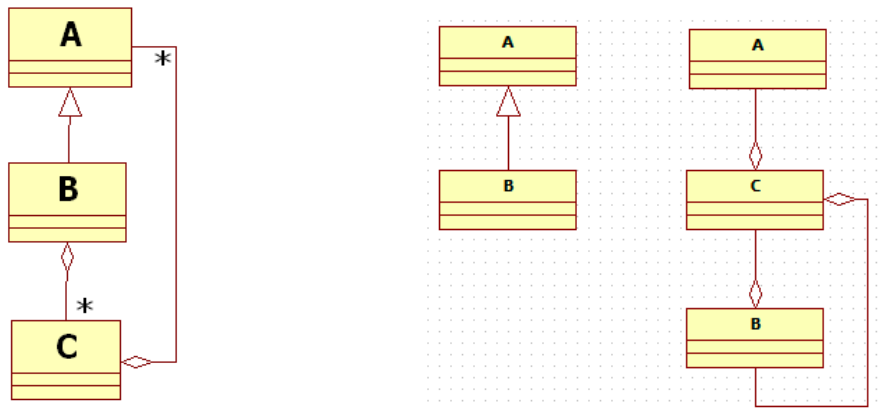
请注意不要雷同

banban

<https://github.com/dream4789/Computer-learning-resources.git>

二、分析设计题

1. 分解关系环，转化为聚合+泛化结构



2. 原先文件夹的设计（图 1）没有问题。后续针对新需求的改进（图 2）有何致命错误？（提示：分解关系环，看包含关系是否正确）如何解决？（提示：若仿图 1 为“需保密”的单做一套，两者无联系，这样会不太方便，因为可能有些“需保密的”是从“普通的”藏过来的）



图 1

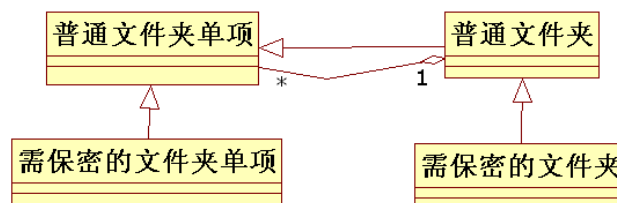
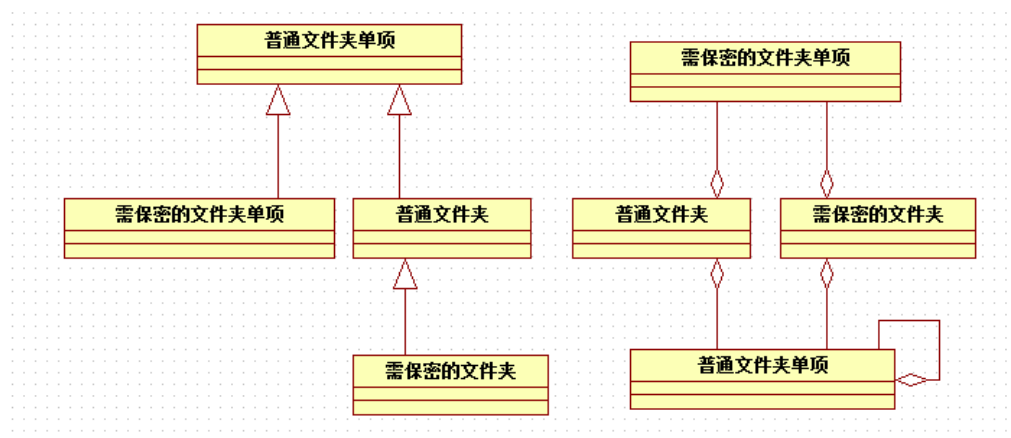


图 2

致命错误：聚合成环了。

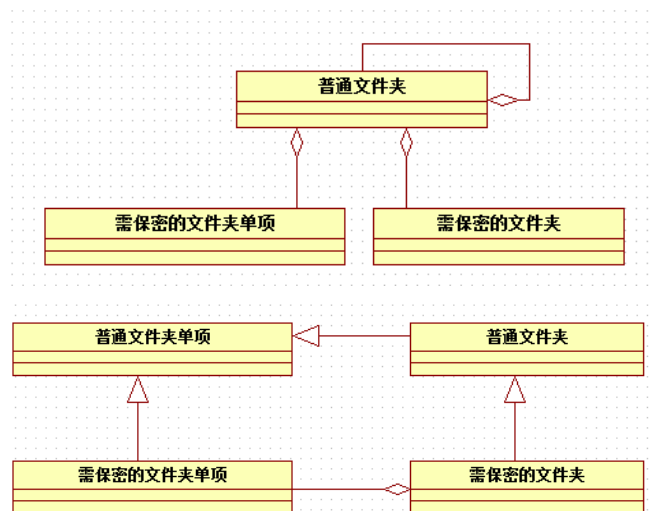


解决方案：需保密的文件夹 聚合 需保密的文件夹单项

请注意不要雷同

banban

<https://github.com/dream4789/Computer-learning-resources.git>



三、编程题（运行截图要能看出你的个人信息，证明不是抄别人的）

学生、课程、选课，多对多，要求双向可见

- （1. 若用 Java 参考笔记中“系”“开设”“课程”一对多代码，请把线程安全的 Vector 换成高效的 ArrayList

```

class Student {
    public static void main(String[] args) {
        Student stu1 = new Student("田七");
        Student stu2 = new Student("张三");
        Student stu3 = new Student("李四");
        Student stu4 = new Student("王五");

        Course c1 = new Course("毛概");
        Course c2 = new Course("算法");
        Course c3 = new Course("python");
        Course c4 = new Course("数据库");

        stu1.setCourse(c1, 65);
        stu1.setCourse(c2, 90);
        stu1.setCourse(c3, 34);
        stu1.setCourse(c4, 45);
        stu2.setCourse(c1, 100);
        stu2.setCourse(c2, 90);
        stu2.setCourse(c3, 93);
        stu2.setCourse(c4, 64);
        stu3.setCourse(c4, 98);
        stu4.setCourse(c4, 88);
        c3.studnetGrade();
    }
}
  
```

请注意不要雷同

banban

<https://github.com/dream4789/Computer-learning-resources.git>

```

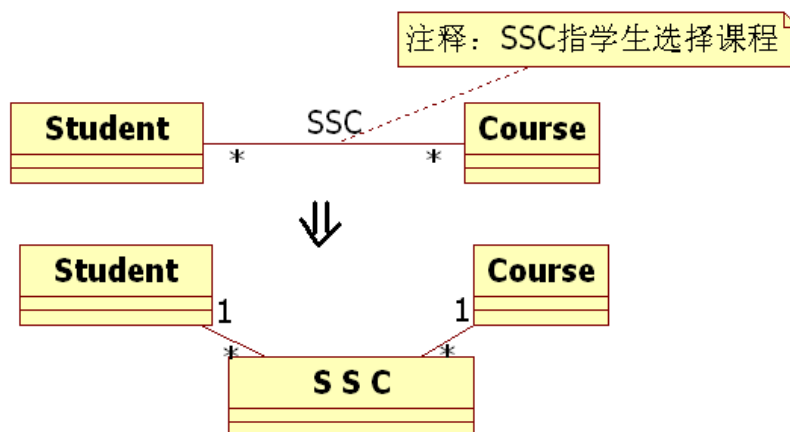
        System.out.println();
        c4.studnetGrade();
        System.out.println();
        stu1.courseGrade();
        System.out.println();
        stu2.courseGrade();
    }

```

- (2. 若用 C++, 学生与 SSC, 课程与 SSC 应理解成类组合（容器里放指针不太合适）；注意深拷贝和析构。也可用 Python。
- (3. 其他方案：由于不像关系数据库对字段有原子性要求，如果非不用 CSS 类，可以在学生类中用 Map 记录课程和得分
- (4. 尝试哪家 UML 工具能根据类图自动生成代码（框架）（先把学生、课程补上 name 属性，SSC 补上分数属性）

做到 234 的高手、请在醒目地方说明下思路(或提交时 209070001 张三 OOAD[UML]作业 1 编程题有改进.doc)

(1. 的方案可以如下：



或者更好的方案（方便记录多个学期的选课记录，如第一学期选了 c, 高数'第二学期选了 c++, 离散）

