



哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

2024 年春季学期
计算学部《软件工程》课程

实验报告

Lab4 OO 分析与设计

姓名	学号	联系方式
徐柯炎	2021110683	2259245769@qq.com/15189789979
高浚豪	2021111700	2152246548@qq.com/13359705210
朱浩铭	2021111749	1115233598@qq.com/13019012603

目 录

1 本组项目概述	1
2 用例模型	1
2.1 角色清单	1
2.2 用例模型	1
2.3 用例 1 的事件流描述	2
2.4 用例 2 的事件流描述	2
2.5 用例 3 的事件流描述	2
3 类识别	3
3.1 边界类	3
3.2 控制类	3
3.3 实体类	4
4 领域模型	5
4.1 分析类图	5
4.2 领域类图	6
5 时序模型	7
5.1 用例 1 的时序模型	7
5.2 用例 2 的时序模型	7
5.3 用例 3 的时序模型	8
6 部署模型	8
7 小结	9

[文档全部完成之后，请更新上述区域]

1 本组项目概述

能够通过系统查看教师的空闲状态并对教师进行预约,教师可以通过系统修改和设置自己的空闲时间段,并能够拒绝或接收学生的预约情况,管理员可以在后台添加和删除教师以及学生的信息。

2 用例模型

2.1 角色清单

给出本系统的 actor (用户、外部软件系统、外部硬件系统、系统时钟等)。

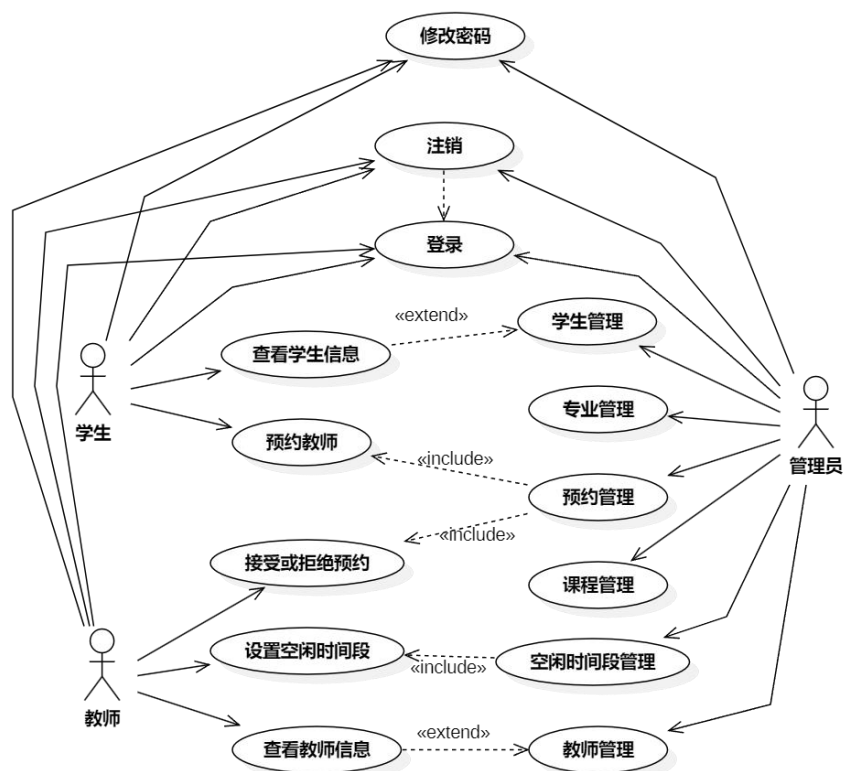
用户: 教师、学生、管理者

外部软件系统: 浏览器, 网页

外部硬件系统: 电脑, 鼠标

系统时钟: 电脑日期, 确保预约在系统规定的时间范围内进行

2.2 用例模型



从所设计的用例中选取 3 个最核心、最主要的，详细描述其事件流，如下。

2.3 用例 1 的事件流描述

本用例所涉及的 actor(s):

管理员, 教师

常规事件流:

- (1) 管理员打开教师管理界面
- (2) 网页展示所有教师的信息
- (3) 根据输入的教师编号查询指定教师
- (4) 编辑教师的信息
- (5) 存储到教师信息的数据库中
- (6) 修改指定教师信息成功

备选事件流:

- (1) 教师管理界面没有成功打开
- (2) 输入的教师编号不存在
- (3) 没有成功保存编辑的教师信息

2.4 用例 2 的事件流描述

本用例所涉及的 actor(s):

管理员, 学生

常规事件流:

- (1) 管理员打开学生管理界面
- (2) 网页展示所有学生的信息
- (3) 根据输入的学生编号查询指定学生
- (4) 编辑学生的信息
- (5) 存储到学生信息的数据库中
- (6) 修改指定学生信息成功

备选事件流:

- (1) 学生管理界面没有成功打开
- (2) 输入的学生编号不存在
- (3) 没有成功保存编辑的学生信息

2.5 用例 3 的事件流描述

本用例所涉及的 actor(s):

教师, 学生

常规事件流:

- (1) 学生打开预约教师管理界面
- (2) 网页展示所有教师的空闲时间段
- (3) 根据输入的空闲时间段显示教师信息

- (4) 添加预约信息
- (5) 存储到预约信息信息的数据库中
- (6) 添加预约信息成功
- (7) 检查核对预约信息

备选事件流:

- (1) 预约管理界面没有成功打开
- (2) 输入的空闲时间段不存在

3 类识别

3.1 边界类

类名（中文）	类名（英文）	类的作用概述
修改密码界面类	Modify the password interface class	提供修改密码的界面
查看教师信息界面类	View the Teacher Information Interface class	提供查看教师信息的界面
接受或拒绝预约界面类	Accept or reject the appointment screen class	提供选择预约的界面
设置空闲时间段界面类	Set the idle time interface class	提供设置空闲时间段的界面
学生管理界面类	Student Management Interface Class	提供学生信息管理的界面
教师管理界面类	Teacher management interface class	提供教师信息管理的界面
专业管理界面类	Professional management interface class	提供专业信息管理的界面
课程管理界面类	Course Management Interface Class	提供课程信息管理的界面
预约管理界面类	Appointment management interface	提供预约信息管理的界面
空闲时间段管理界面类	Idle time period management interface class	提供空闲时间段信息管理的界面

3.2 控制类

类名（中文）	类名（英文）	类的作用概述
--------	--------	--------

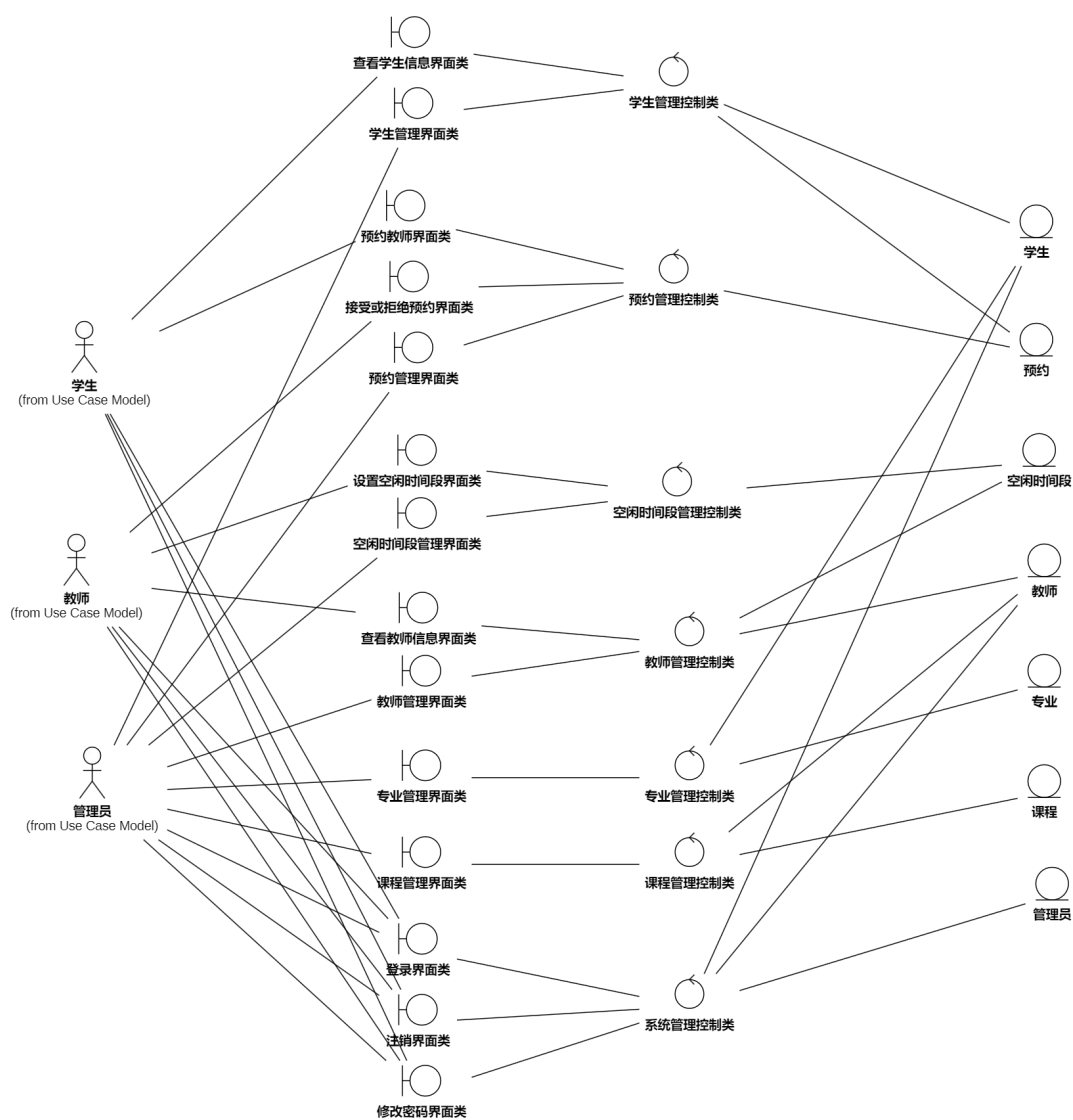
教师管理控制类	TeacherController	管理教师的信息，调用教师的相关方法
系统管理控制类	SystemController	存放所有密码，控制登录界面
学生管理控制类	StudentController	管理学生的信息，调用学生的相关方法
专业管理控制类	ClazzController	管理专业的信息，调用专业的相关方法
课程管理控制类	CourseController	管理课程的信息，调用课程的相关方法
空闲时间段管理控制类	AvailabilityController	管理空闲时间段的信息，调用空闲时间段的相关方法
预约管理控制类	ReserveControlle	管理预约的信息，调用预约的相关方法

3.3 实体类

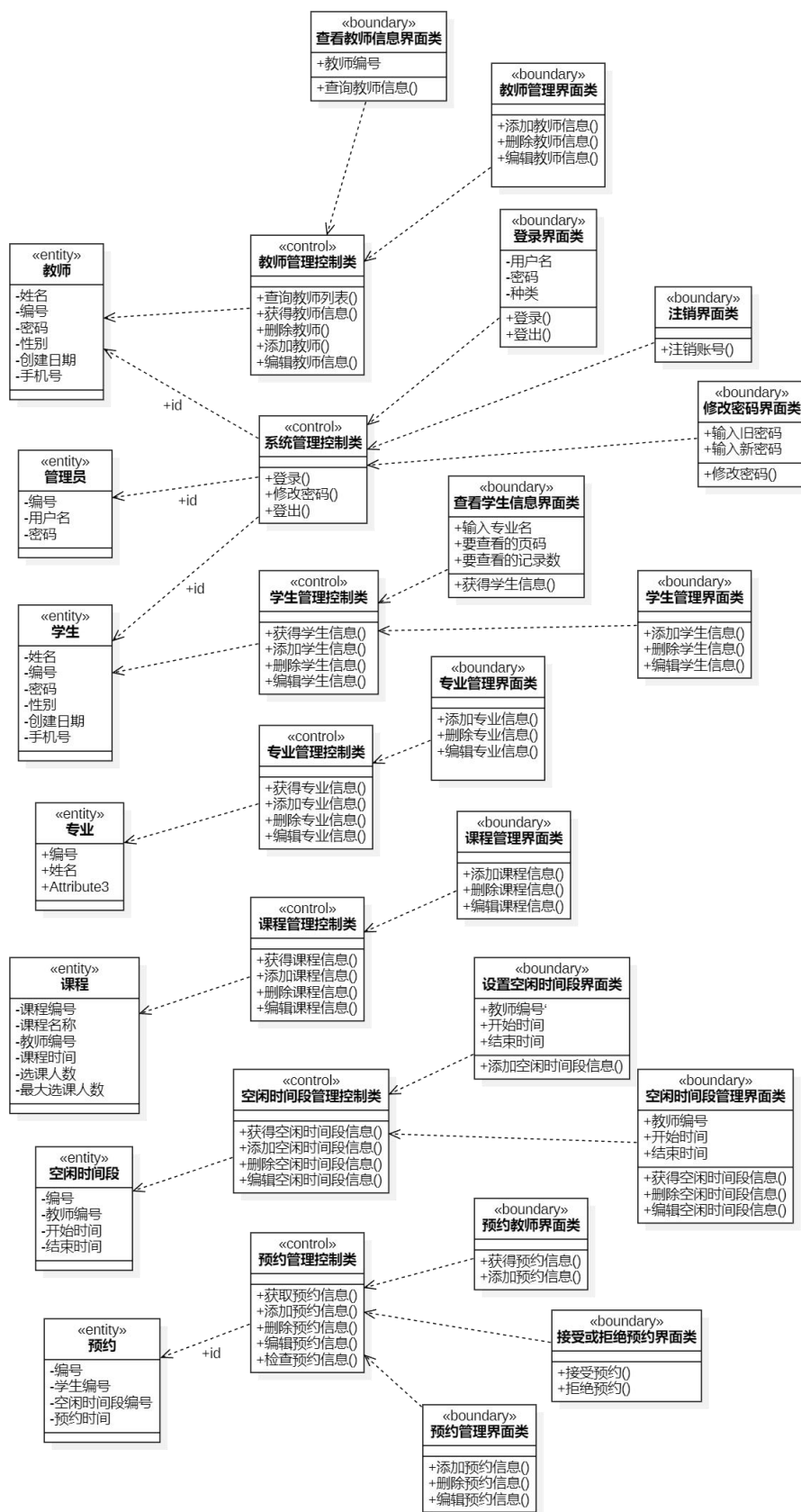
类名（中文）	类名（英文）	类的作用概述
教师	Teacher	存放教师的相关信息
学生	Student	存放学生的相关信息
管理者	Admin	存放管理员的相关信息
专业	Clazz	存放专业信息
课程	Course	存放课程信息
空闲时间段	Availability	存放空闲时间段信息
预约	Reserve	存放预约信息

4 领域模型

4.1 分析类图

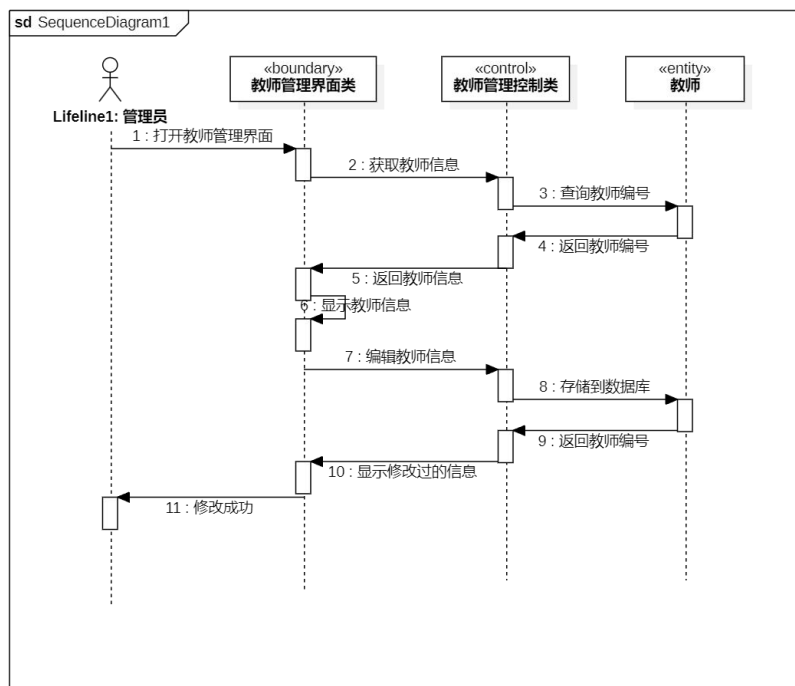


4.2 领域类图

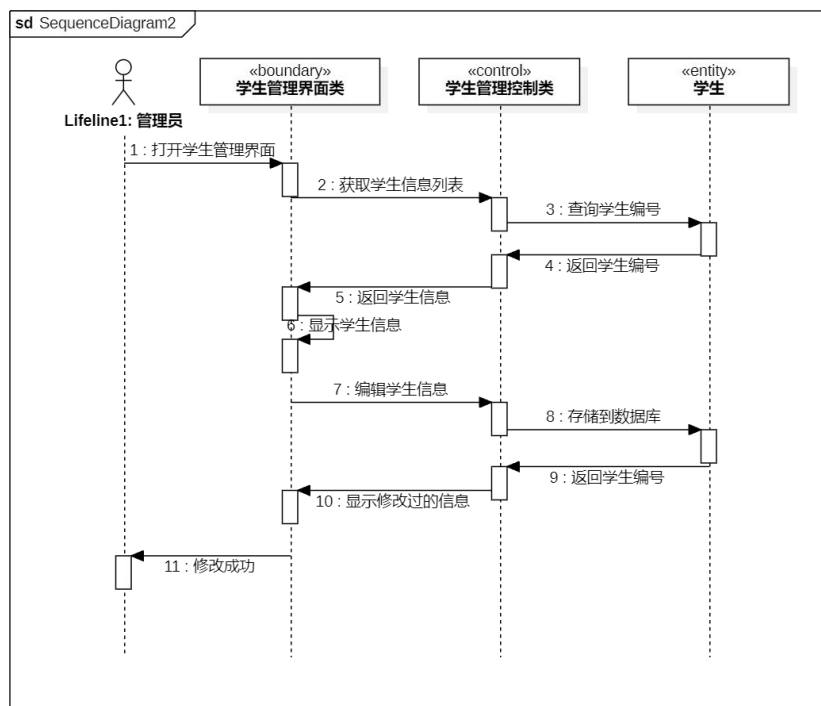


5 时序模型

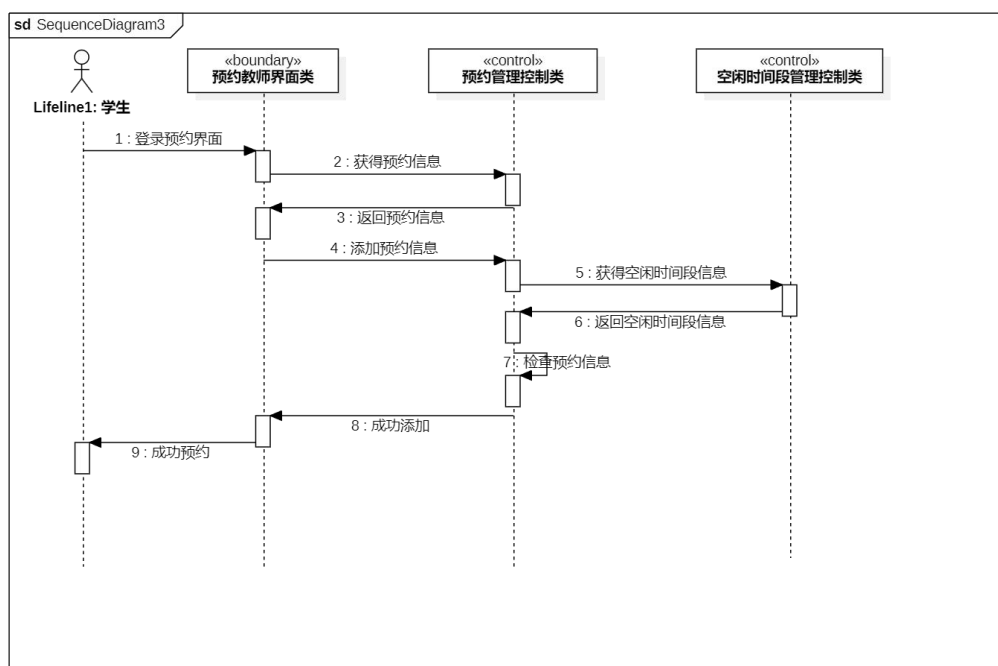
5.1 用例 1 的时序模型



5.2 用例 2 的时序模型



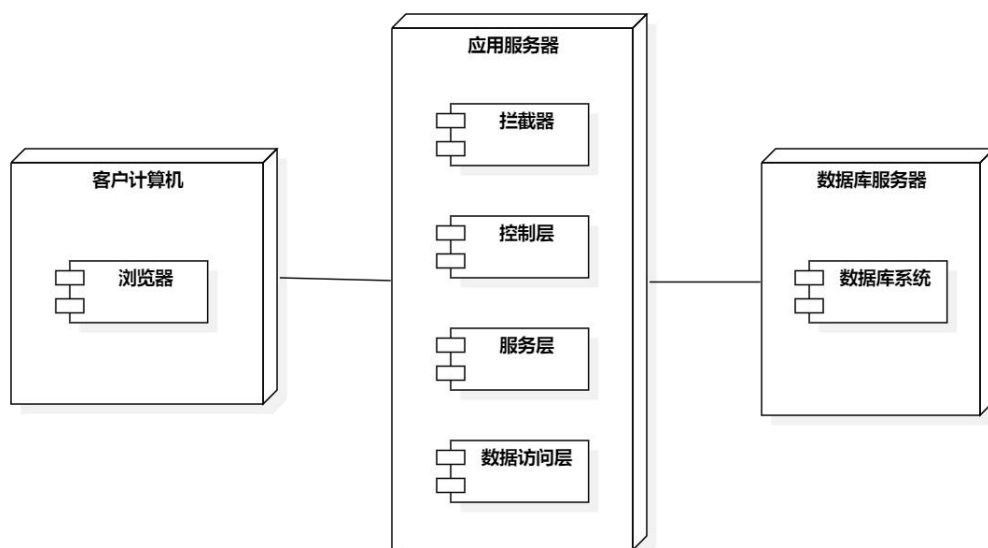
5.3 用例 3 的时序模型



6 部署模型

建立本组的项目的最终部署架构（需要哪些 server、哪些客户机、网络连接关系如何、等等），使用 UML 部署模型加以描述。

一般情况下，遵循 B/S 架构，即系统部署在云端或服务端，用户通过手机端或者 PC 端加以访问。但根据你的系统的具体特征和性能要求，可酌情引入新的物理设施。



7 小结

- (1) 建模是否有必要？直接去写代码呗！边写代码边思考，岂不是省了建模的大量时间？

建模在软件开发中是非常必要的。尽管直接编写代码可能看起来更快速，但建模能够帮助我们提前预见潜在的问题、减少不必要的返工、提高代码质量，并有助于团队成员之间更好地沟通和协作。建模是软件开发过程中的一种“蓝图”，它可以帮助我们更好地理解系统的整体结构和功能，从而更有效地进行开发。

- (2) 建模时如果能思考得很细节，其实就相当于写代码时的思考工作。如何能逼着自己在建模时想得更细节，你们是否总结出什么方法？

首先应该明确项目的需求，将需求转化为具体的模型元素，对需求有深入的理解。其次应该使用合适的工具，选择支持细节建模的工具可以有助于更具体的表达模型。另外应该积极团队讨论，与团队成员分享模型，听取团队成员的反馈和意见，有助于发现可能遗漏的细节。最后要通过多次的项目迭代和量化，逐步添加更多的细节。

- (3) 你对“面向对象”这套软件分析与设计的方法，怎么看？它与传统的以“算法+数据结构”为单位的结构化分析方法相比有什么好的地方和不好的地方？

面向对象的分析与设计方法（OOAD）是现代软件开发中非常流行的方法之一。与传统的“算法+数据结构”方法相比，OOAD 有以下优点：

- （1）可重用性：对象可以被封装和复用，减少了代码的冗余。
- （2）可扩展性：通过继承和接口，可以方便地扩展系统的功能。
- （3）可维护性：对象的封装性使得修改一个对象的功能时，对其他对象的影响较小。

然而，OOAD 也有其局限性，例如在某些情况下，使用简单的数据结构和算法可能更加高效。此外，面向对象的设计可能会增加系统的复杂性，需要更多的学习和实践才能掌握。

- (4) 你对 UML 这种建模语言怎么看？它给出了这么多的 diagram，各 diagram 是否的确起到了帮助开发者的作用？你们对各个 diagram 的作用怎么看？

UML（统一建模语言）是一种广泛使用的建模语言，它提供了一套丰富的图形符号和规则，用于描述系统的结构、行为和交互。UML 的各种 diagram 确实起到了帮助开发者的作用，它们可以帮助我们更直观地理解系统的各个方面。

- (5) 在以后的开发中，你们是否愿意先建模、开发团队再针对模型进行讨论修改、然后再进入到开发环节？

在以后的开发中，我们应该尽量遵循先建模、再讨论修改、然后进入开发环节的过

程。这样做可以帮助我们提前预见潜在的问题、减少不必要的返工、提高代码质量，并有助于团队成员之间更好地沟通和协作。