第2次作业: 类图

教材第73至74页,习题第8题,第10题,第18题至21题,共6题.

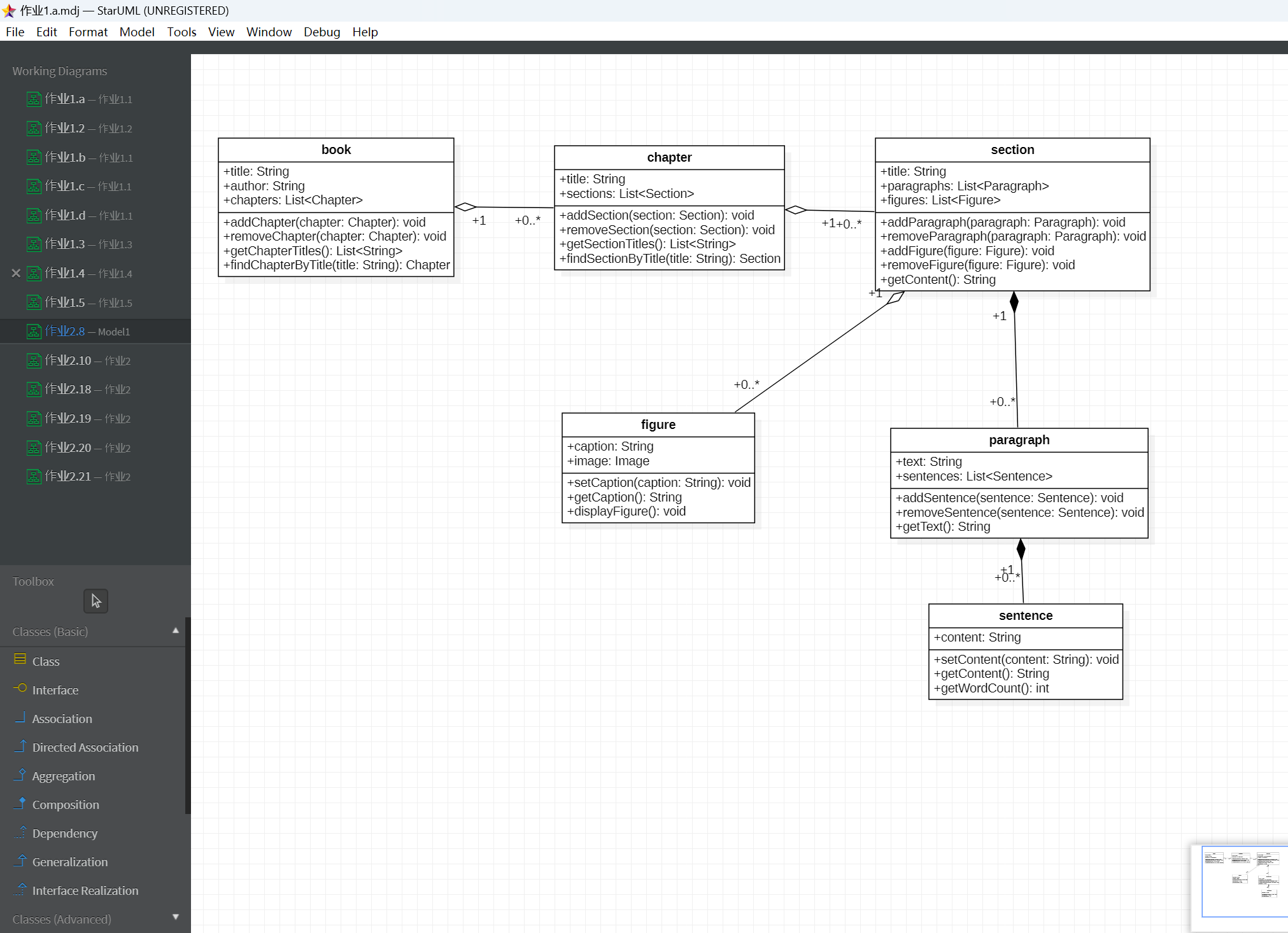
8. 一本书含有若干章,一章有若干节,一节由一些段落和图组成,一个段落由一些句子组成.请给出一个描述上述事物以及它们之间关系的类图.

### 类图说明：

1. **Book类**：表示一本书，包含书名、作者信息以及若干章节。
2. **Chapter类**：表示书中的一章，包含章节标题和若干节。
3. **Section类**：表示一节，包含节标题、段落列表以及图列表。
4. **Paragraph类**：表示一个段落，包含段落文本和句子列表。
5. **Figure类**：表示一幅图，包含图的标题和具体图像数据。
6. **Sentence类**：表示一个句子，包含句子内容。

### 关系说明：

* **Book**包含多个**Chapter**。
* **Chapter**包含多个**Section**。
* **Section**可以包含多个**Paragraph**和**Figure**。
* **Paragraph**由多个**Sentence**组成。



1. 有的房间是立方体的,有的房间是圆柱体的.根据这样的说法,给出一个类图.

关系说明

Room **类**：

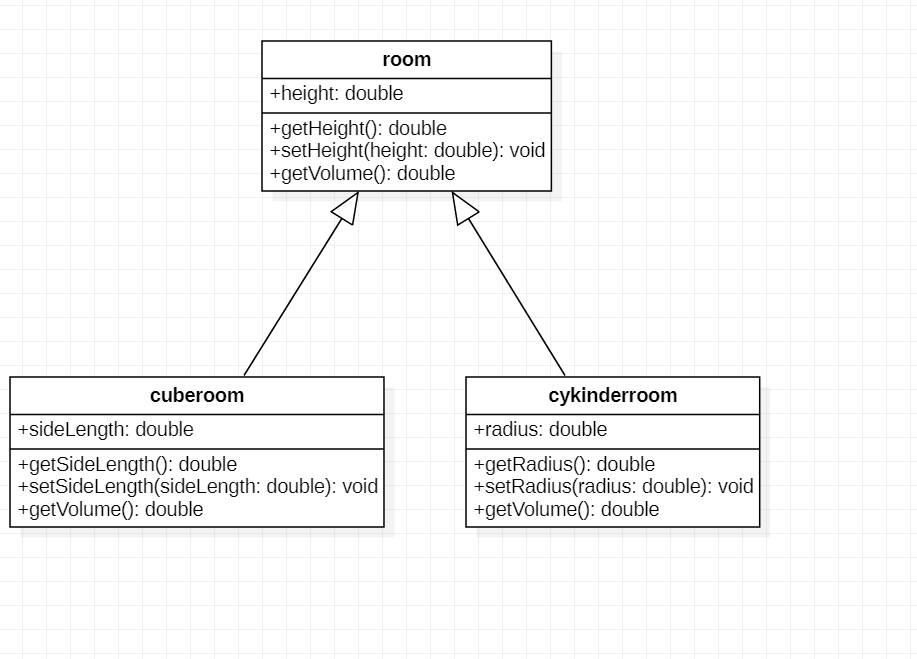
* 表示房间的抽象概念，定义公共属性和操作（如高度和体积计算）。
* 属性：height：房间的高度。
* 方法：getVolume()：计算房间体积（抽象方法或虚方法）。

CubeRoom **类**：

* 表示立方体房间，继承自 Room。
* 属性：sideLength：立方体边长。
* 方法：重写 getVolume() 方法：体积 = 边长³

CylinderRoom **类**：

* 表示圆柱体房间，继承自 Room。
* 属性：radius：圆柱体半径。
* 方法：重写 getVolume() 方法：体积 = π × 半径² × 高度。



1. 为火车票订票系统建立类图.具体的需求为:预订某一车次的车票,包括具体的时间和座位;在预订后,顾客必须在一定的时间内购票,否则预订无效;旅行社和火车售票处均可以进行预订业务.

### ****类图设计****

#### ****主要类****

1. **Train**：表示火车和车次相关信息。
2. **Ticket**：表示具体的车票信息。
3. **Reservation**：表示预订相关信息，包括预订时间、座位等。
4. **Customer**：表示预订车票的顾客。
5. **Agency**：表示可以执行预订业务的旅行社。
6. **TicketOffice**：表示可以执行预订业务的火车售票处。

### ****类之间的关系****

**Train - Ticket**

* 1. **关系**: 关联
  2. **描述**: 一个火车包含多个车票。
  3. **多重性**: 1..\* (一辆火车对应多个车票)

**Train - Reservation**

* 1. **关系**: 关联
  2. **描述**: 一个火车对应多个预订信息。
  3. **多重性**: 1..\* (一辆火车对应多个预订)

**Reservation - Customer**

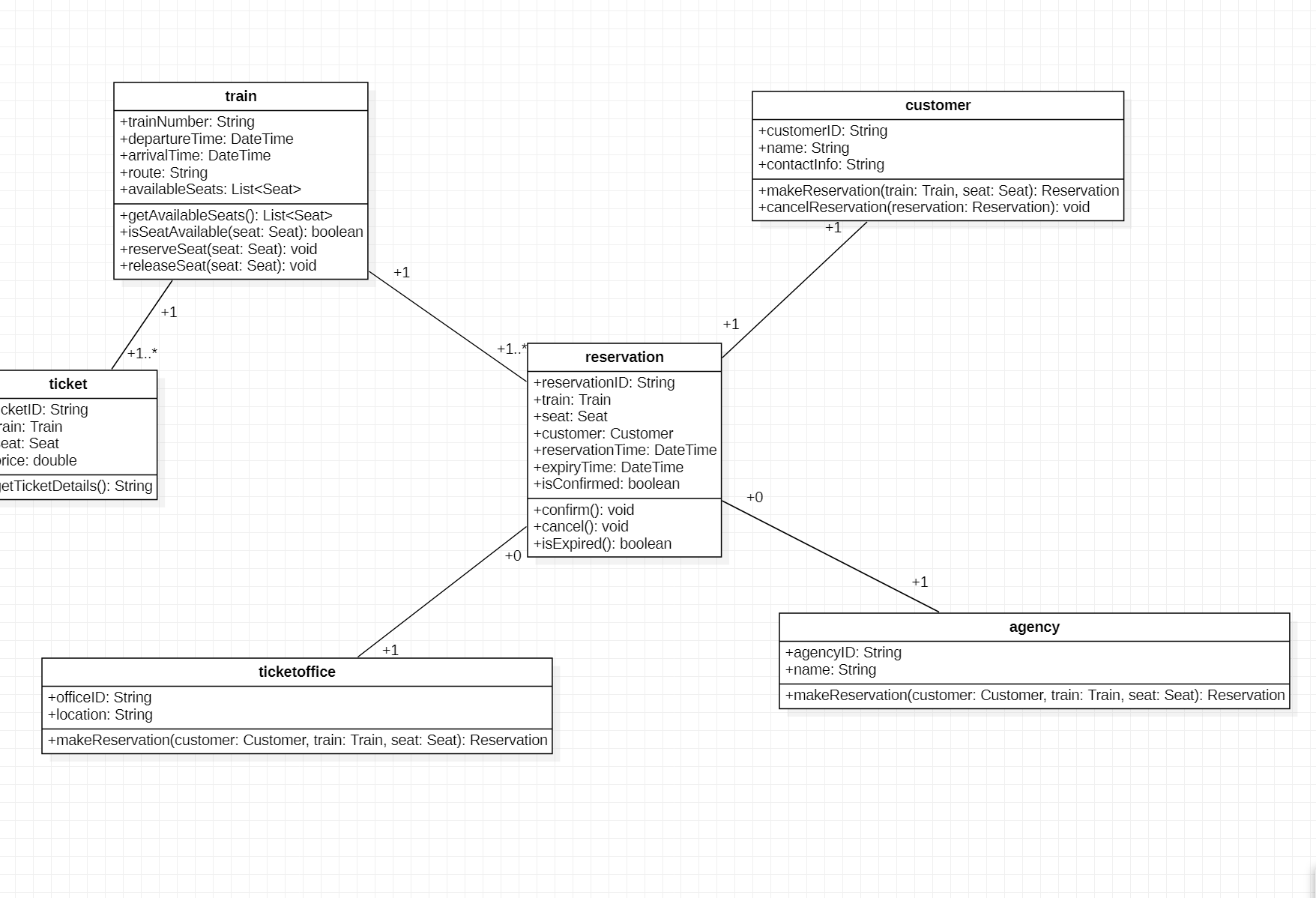
* 1. **关系**: 关联
  2. **描述**: 一个预订由一个顾客发起。
  3. **多重性**: 1..1 (一个预订只能由一个顾客发起)

**Reservation - Agency**

* 1. **关系**: 关联
  2. **描述**: 一个预订可能由旅行社代办。
  3. **多重性**: 0..1 (一个预订可以由旅行社处理，也可以由售票处处理)

**Reservation - TicketOffice**

* 1. **关系**: 关联
  2. **描述**: 一个预订可能由售票处代办。
  3. **多重性**: 0..1 (一个预订可以由售票处处理，也可以由旅行社处理)



1. 针对自行车(及其主要部件),建立一个简单的类图.

### 自行车类图设计

自行车的主要组成部件包括：**车架**、**车轮**、**车座** 和 **车把**。以下是一个简单的类图，表示自行车和这些部件的关系，以及每个类的属性和操作。

类之间的关系

**Bicycle - Frame**

**关系**: 组合（Composition）

**描述**: 自行车必须有车架，车架是自行车的重要组成部分。

**UML符号**: 实心菱形指向 Frame。

**Bicycle - Wheel**

* **关系**: 组合（Composition）
* **描述**: 自行车必须有车轮，通常有两个车轮。
* **多重性**: 2..2 (两个车轮)

**Bicycle - Seat**

**关系**: 组合（Composition）

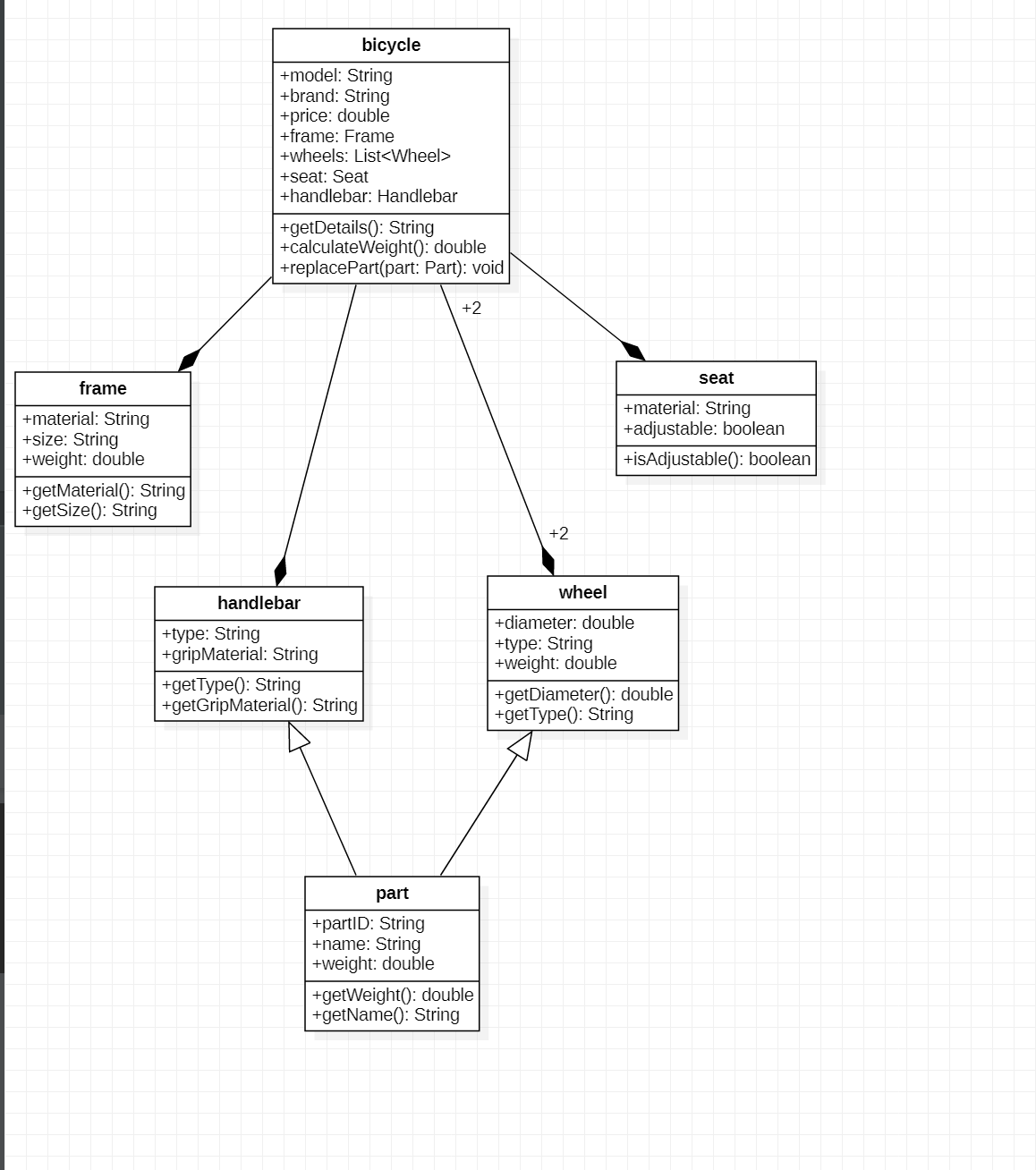
**描述**: 自行车必须有车座，车座是组成部分。

**Bicycle - Handlebar**

* **关系**: 组合（Composition）
* **描述**: 自行车必须有车把，车把是组成部分。

**Part（抽象类）**

* **关系**: 通用关系
* **描述**: Frame、Wheel、Seat 和 Handlebar 都继承自 Part，表示零部件的公共属性和操作。



20 针对下述问题,建立一个类图:有两种顾客,一种是常客,享受公司的一些优惠待遇;另一种是散户.

### 类图设计：两种顾客（常客和散户）

根据问题描述，我们可以通过继承关系表示 **常客** 和 **散户** 两种顾客的特性。一个基类 Customer 表示顾客的通用信息和操作，RegularCustomer 和 CasualCustomer 继承自 Customer，分别实现各自的特性。

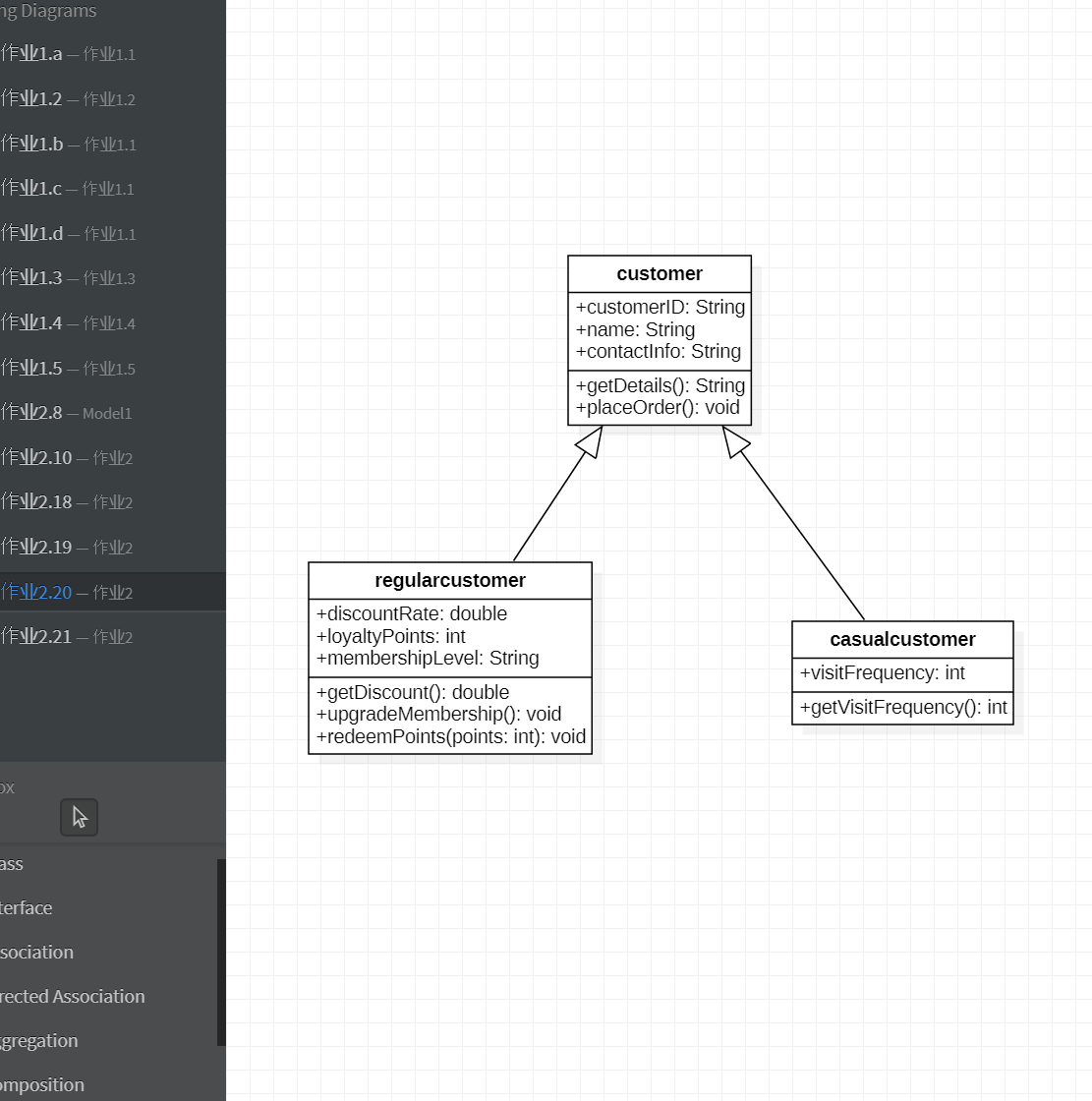
类之间的关系

**Customer - RegularCustomer**

* **关系**: 继承（Generalization）
* **描述**: 常客是顾客的一种类型，继承顾客的基本属性和操作。
* **UML符号**: 实线箭头指向基类 Customer。

**Customer - CasualCustomer**

* **关系**: 继承（Generalization）
* **描述**: 散户是顾客的另一种类型，继承顾客的基本属性和操作。
* **UML符号**: 实线箭头指向基类 Customer



1. 对于你所学过的课程,建立类图.课程所属于的科目是不同的,而且有些课程需要在某些先修课程之后开设.

以下是类图的关键设计思路：

1. **课程类 (**Course**)**:

属性包括：课程名称、课程类型（必修或选修）、学分等。

1. **科目类 (**Subject**)**:

每个课程属于某个科目，如数学、计算机科学等。

1. **先修关系 (**Prerequisite**)**:

课程之间的依赖关系由该类定义。

类之间的关系

**Subject 与 Course**:

* 一个科目可以包含多个课程（1..\*）。
* 一个课程必须属于一个科目（1）。

**Course 与 Prerequisite**:

* 一个课程可以有多个先修课程（0..\*）。
* 一个先修关系关联两个课程（1 和 1）

**Course 自身的关联**:

通过 prerequisites 属性实现课程之间的先修依赖。

具体关系示例

**Subject（科目）**

* 数学科目：微积分、线性代数
* 计算机科学科目：C语言、人工智能、数据科学、数字图像、计算机图形学

**必修课程**:

* C语言 是所有选修课的先修课程。

**选修课程**:

* 人工智能 是最后的选修课程，它依赖于 数据科学 和 计算机图形学。

