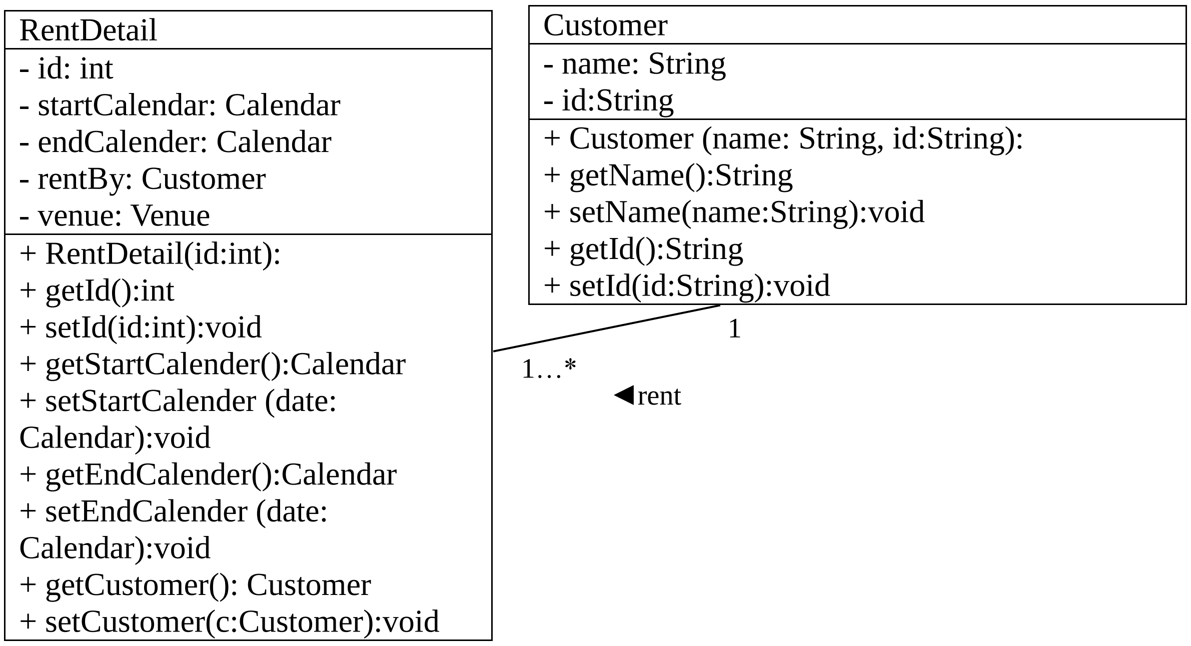
PROG 2114 Advanced Java Programming

<u>Case study:</u>

Answer the following questions based on the UML class diagram below:


RentDetail
- id: int
- startCalendar: Calendar
- endCalender: Calendar
- rentBy: Customer
- venue: Venue
+ RentDetail(id:int):
+ getId():int
+ setId(id:int):void
+ getStartCalender():Calendar
+ setStartCalender (date: Calendar):void
+ getEndCalender():Calendar
+ setEndCalender (date: Calendar):void
+ getCustomer(): Customer
+ setCustomer(c:Customer):void
Customer
- name: String
- id:String
+ Customer (name: String, id:String):
+ getName():String
+ setName(name:String):void
+ getId():String
+ setId(id:String):void
1
1…*
◀rent

The Venue Rental Company hires you to work on their online venue renting system. Customer can rent a venue by specify the date and time. The renting request will be accepted by the system if the request does not overlap with other existing accepted request.

Complete the following tasks:

a) Create a class named Customer by follow all the specification shown in UML class diagram above.

b) Create a class named RentDetail (as shown in UML class diagram above) and fulfill the following requirements:
   i. Include all the data fields, constructor method, accessor and mutator methods as shown in UML class diagram.
   ii. Implement the Comparable interface and compare the startCalendar and endCalendar of two renting request.

   For example:
   Renting request 1: Start Calendar = 1/1/2021 8:00 and End Calendar = 1/1/2021 10:00

Renting request 2: Start Calendar = 1/1/2021 11:00 and End Calendar = 1/1/2021 12:00
After compare the start and end calendar, request 1 is earlier (*smaller*) than request 2.

Renting request 1: Start Calendar = 1/1/2021 12:00 and End Calendar = 1/1/2021 1:00
Renting request 2: Start Calendar = 1/1/2021 11:00 and End Calendar = 1/1/2021 12:00
After compare the start and end calendar, request 1 is late (*bigger*) than request 2.

Renting request 1: Start Calendar = 1/1/2021 8:00 and End Calendar = 1/1/2021 10:00
Renting request 2: Start Calendar = 1/1/2021 9:00 and End Calendar = 1/10/2021 11:00
After compare the start and end calendar, request 2 is overlap with request 1. The overlap request is not allowed in the system.

iii. Override the method toString() to return a string description of the successfully renting details. Following is the sample output:

```
Rent by: Alan (D12345)
Rent ID: 1001
Rent Venue: Meeting Room 125
Start Date & Time: Fri Jan 1 08:00:00 SGT 2021
End Date & Time: Fri Jan 1 09:00:00 SGT 2021
```

c) Modify the Customer class based on the requirements below:

i. Add a private data field named RentList that holds a list of RentDetail objects.

ii. Write an accessor method to return the list of RentDetail objects.

iii. Add a new method called addRentalDetail(RentDetail) to add a renting request into the RentList. The method will add the rent detail object into the list only if the start and end date are not overlap with other RentDetail object in the list. Otherwise, nothing will be added.