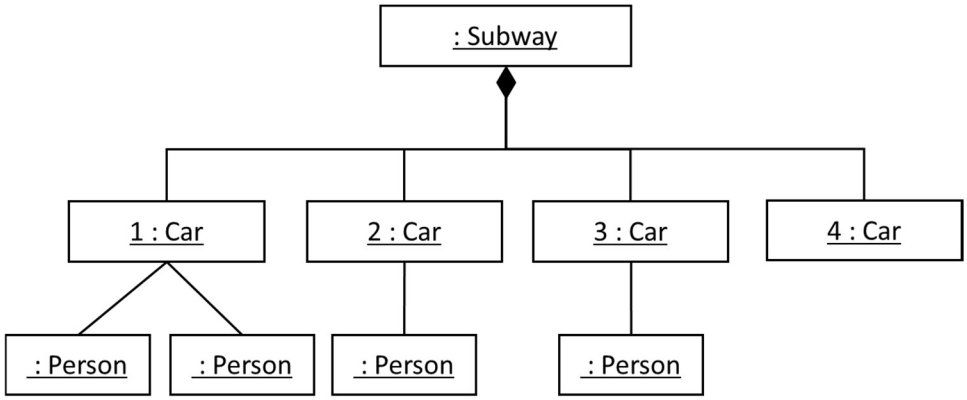
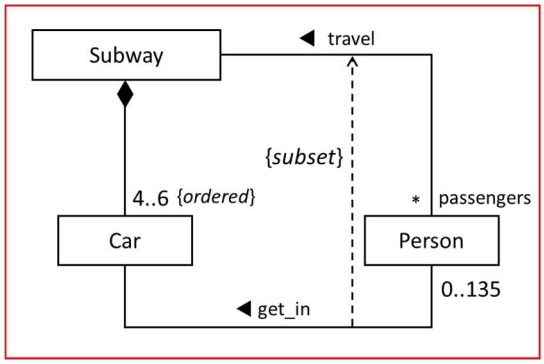
8th class: Association, aggregation, and composition

Task: Give the class diagram of the following problems. Use association, aggregation, and composition. Imagine how to populate some of the tasks and draw the object diagram for them.

1. Some people travel of subways. One subway consists of 4, 5, or 6 cars. At last 135 persons can get on a car.



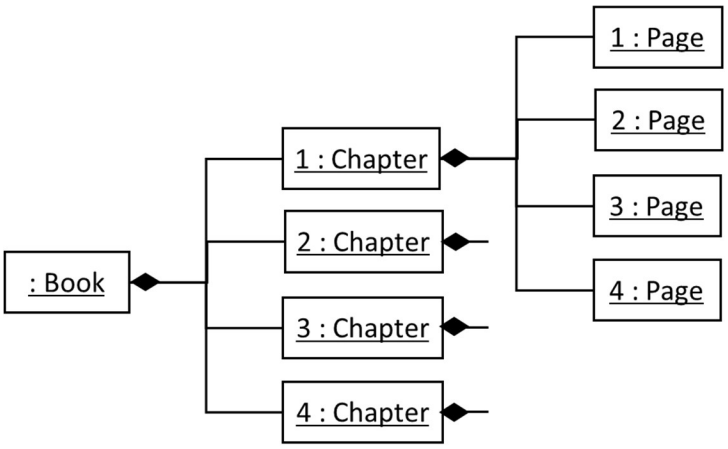
The object diagram shows one example, a 4-car subway with 4 persons on it: the first two persons in the first car, the third person in the second car, and tha last person in the third car. The last car is empty.



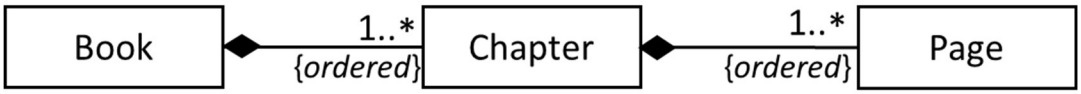
The cars belong to the subways, there is a special association between them, a composition. The composition is a special aggregation, where the container (subway) does not exist without the contained elements (cars) and e.g. one car can belong to only one subway.

The people on the subway are referred as passengers by the subway class. If we want the Subway class to access effectively its passengers, then we should create a collection attribute in the Subway class storing maximum 135 Persons. We could also indicate that two same persons cannot travel on the subway by using the {unique} keyword next to the role name *passengers*. We could give a role name for the Person class in the get_on association, too to show that we store the persons getting on a concrete car.

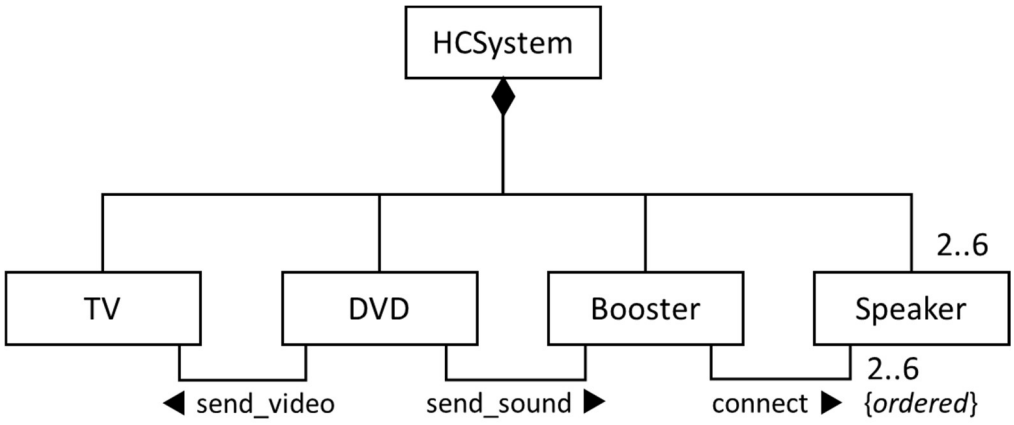
2. One book consits of at least one chapter, one chapter contains minimum one page. Give a class- and a sample object diagram for the task!



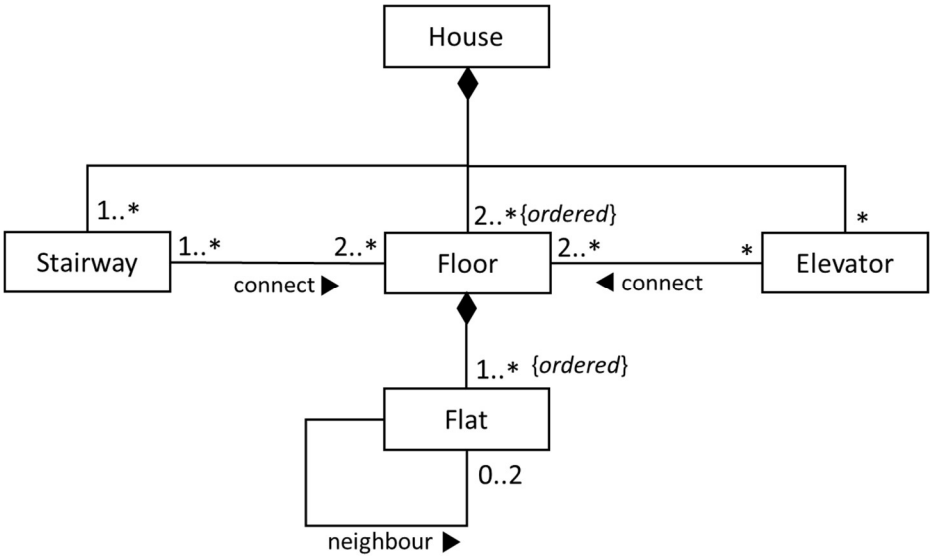
The class diagram is much simpler, instead of a tree-view, we can use multiplicities:



3. Prepare the class diagram of a home cinema system. The system consists of a TV, a DVD player, a booster, and some speakers. There are from 2 to 6 speakers. The DVD player sends video signal to the TV and sound signal to the speakers. The booster connects the speakers.

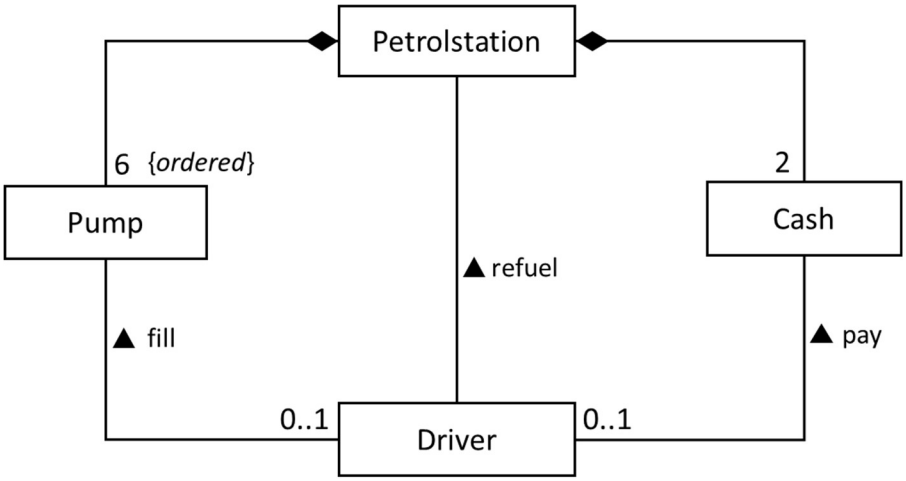


4. A terraced house consists of at least two floors, there is at least one flat on each floor. Floors are connected by stairways and elevators. Every floor is connected by at least one stairway. Every flat has at last two neighbours.

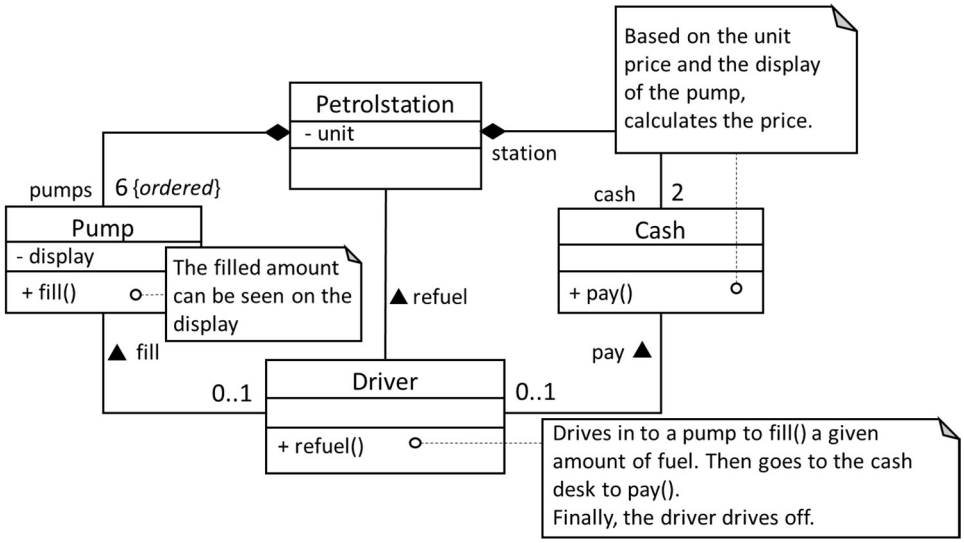


5. On a petrol station, there are 6 pumps and 2 cash desks. When a driver drives in, decides at which pump to fill (it is a message sent to the station, which checks if the given pump exists). After getting to the pump, the driver fills the vehicle with a given amount of fuel, then goes to pay at the cash desk (the cash desk checks the amount of fuel filled by the driver, calculates the price, and clears the display of the pump). Finally, the driver drives off. One driver can use only one pump and only one cash desk. This task is going to be solved again on a later lecture with parallel blocks and awaits.

   First, let us see the main structure of the program:

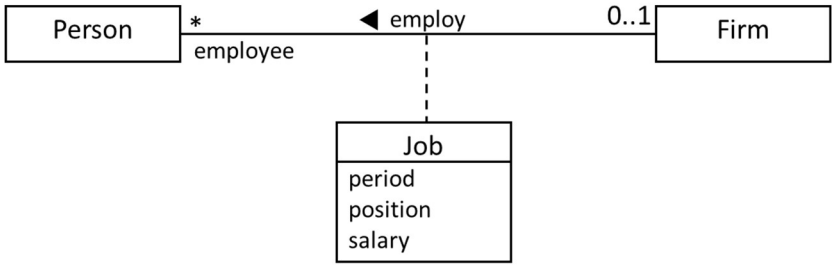


Then, let us consider what procedure refuel() should do. For that, let us introduce some new attributes and methods:

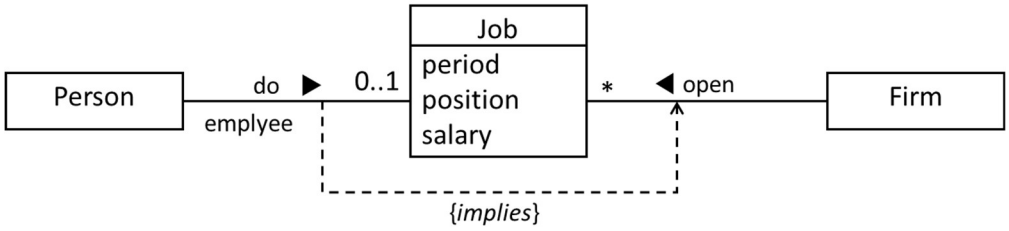


Homework: decide the visibility and the owner of the role names, define some getters and setters, and introduce some new methods (e.g. to access a pump of a given index from the Cash_desk class so that its fill() method could be called).

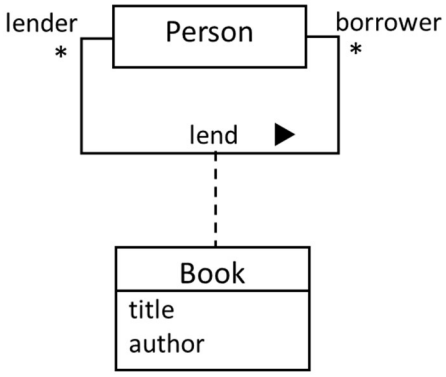
6. A firm employs people in different positions. One person can be emplyed in at last one firm. The job description contains the starting (and finishing) date of the job, the position of the employee, and the salary.

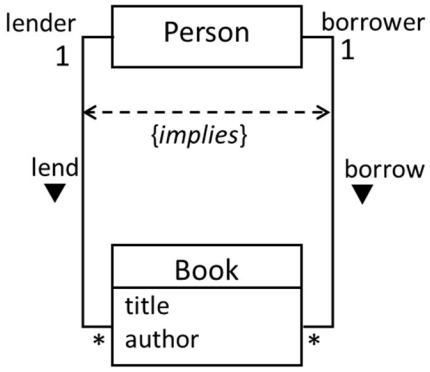


In the above model a so-called association class is used. One instance of the association class (Position) shows the relation between one person and one firm. In the model below the association class is replaced by a normal class.



7. A person lends a book to another person. The title and the author of the book is known (with and without association class).

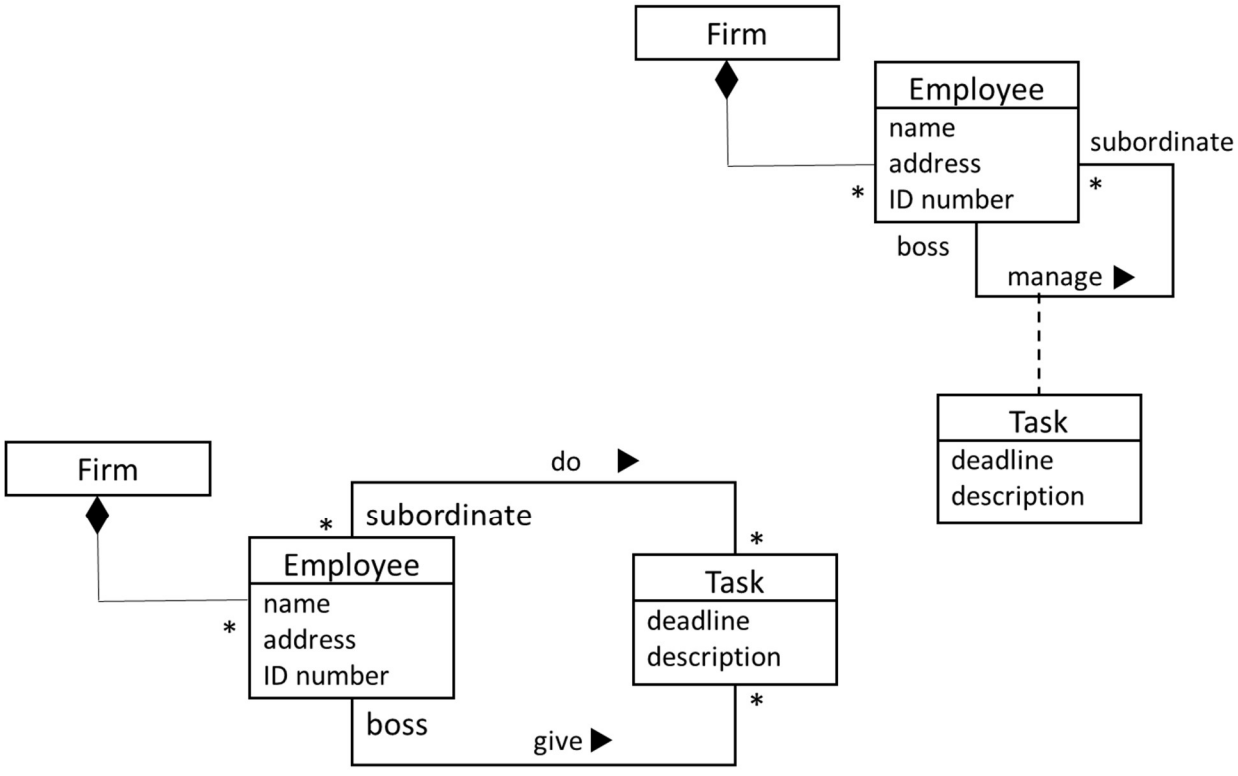




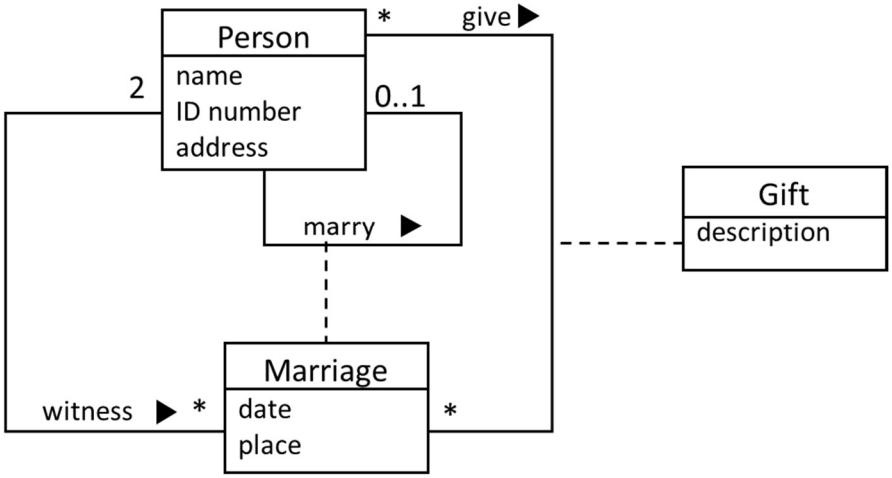
An association class can be replaced by a normal class according to certain rules: the original association is deleted, but the new normal class which replaces the association class gets into an association with the classes of the original association. The multiplicities of the original association are kept and used in the new associations at the side of the new class. The remaining multiplicities are set to 1.

If the association class is an n-ary class, it can be replaced by a normal class in the same way.

8. Employees of a firm (with name, address, and ID number) consist of a boss and subordinates. The boss manages the subordinates. In managing, we mean finishing a task before a deadline.



9. A possible relation between two persons (with name, ID number, and address) can be a marriage. One person can be married to maximum one person. Attributes of the marriage are the place and the date. One marriage has 2 witnesses (2 persons). At the marriage, persons give wedding presents. An attribute of a wedding present is its description.



Without assoiation classes: