Week 15-16-17: Final individual assignment

This is the final individual assignment to give you better insight about your proficiency related to the covered concepts and the learning outcomes. Your lecturer will give you a formative indication and feedback to help you understand your proficiency.

The assignment consists of a description, example data, a list of requirements and an (incomplete) UML class diagram. You are required to implement an application based on these. The supplied class diagram is to give you a hint how the classes should be, but you will need to complete it with the actual class members (e.g., instance variables, methods, enums, etc.). That is, you get a class diagram completed till step 3.1 (following the steps numbering given in the lessons on design) and you pick it up from there.

Note that this assignment is individual and is used by your lecturers to evaluate if you reached the expected proficiency for this semester. For this reason, it must be your own code and you are not allowed to copy any part of it from your fellow peers. Your lecturer will give you feedback about what you submit and we assume that it is your own code and that you understand it fully.

Overview of assignments

Constraints

- You have to do one of the assignments and your lecturer(s) will tell you which one.
- You can use the supplied (incomplete) UML class diagram as a guideline for your solution; note that you may also come with your own, sensible, class design.
- Hand in your final version before the given deadline via Canvas.
- Your lecturer(s) will schedule, in week 18, a meeting with you. During this meeting, the focus will be on your explanation about the reasoning behind your implementation and on seeing if you can extend/modify your solution. You will receive feedback based on your performance.

Deliverables

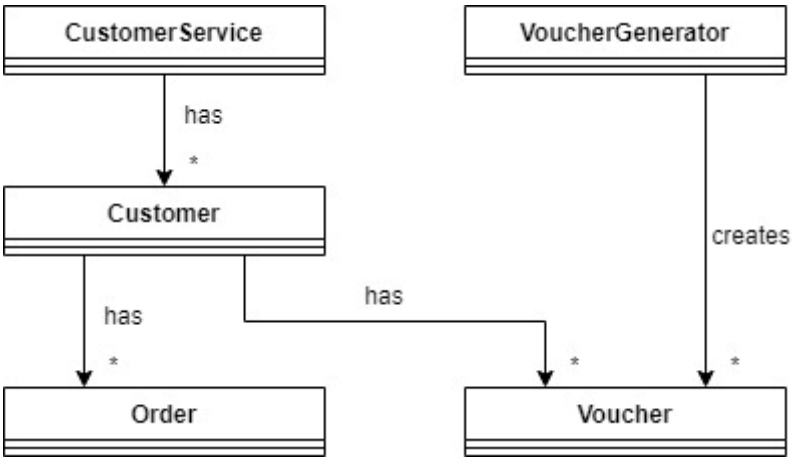
For this final individual assignment you are expected to submit in Canvas:

- Solution (i.e., source code) as a Windows Forms Application; make sure you submit everything so it can be opened in Visual Studio.
- UML class diagram.
- If present, database diagram; you can get this from MS SQL Management Studio and do make sure all data is included, i.e., include the tables as a ‘standard’-table view and the relationships.
- If needed, small document with instructions on how to use your application.

Coffee to Go

Company RoastedBeans™ sells coffee at various places where people pass by, like stations, parks, malls etc. In order to bind their customer base, they want an app to trace the number of coffees the customers drink, the type of coffee, and want to give them free coffee vouchers once they reach a certain number of orders.

To help you on your way, below you find a nice starting point UML diagram, which you can change or extend. It only shows the classes, so it is not complete.



Coffee details example 1	
Size	Medium
Type	Regular
Special	No
Extras	-
Price	€ 2,50

Coffee details example 2	
Size	Large
Type	Latte
Special	Yes
Extras	Marsh-mellows
Price	€ 4,50

Coffee details example 3	
Size	Medium
Type	Espresso
Special	No
Extras	-
Price	€ 2,00

This app should at least be able to:

- Add a customer to the customer base.
- Register an order to a customer.
- Remove a customer from the system.
    - Note: Removing a customer should also remove any outstanding vouchers.
    - Note: Orders should NOT be removed, since they provide a financial trail.
- Add a voucher generator, which will create a voucher after 10 coffees.
    - Note: The voucher cannot be used immediately in the same order.
- Use the voucher and mark it complete.
- Save and load all data to (a) file(s).
- Calculate the total amount of coffee sold per month.
- Calculate the total amount of coffee sold per customer.
- Find the top 10 customers, and manually give them a voucher.

Optional:

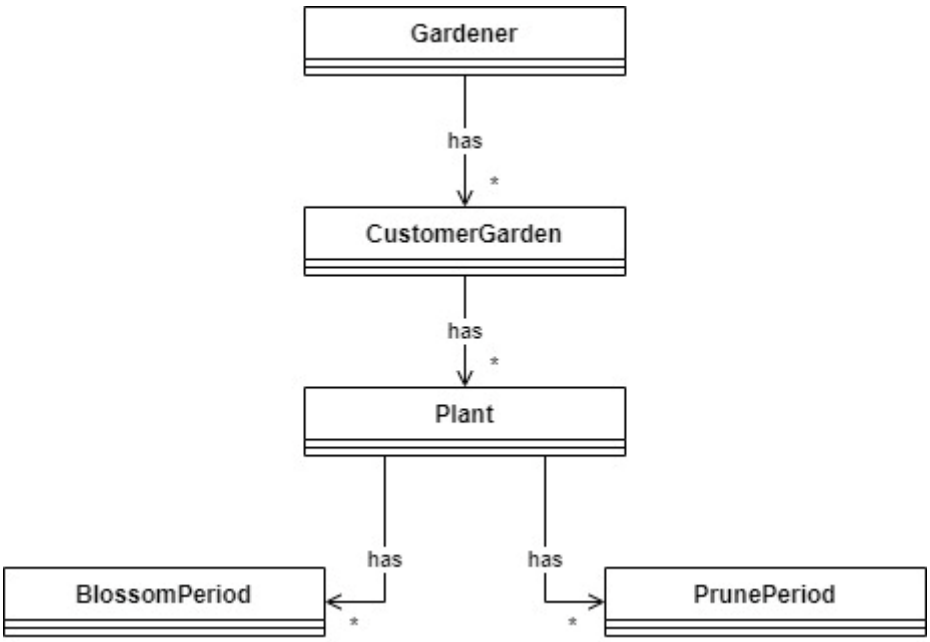
- In addition to files, save and load all data from a database.
- Create a system of bronze, silver, and gold vouchers.
    - Bronze: Given when buying 5 coffees
        - Free regular coffee
    - Silver: Given when buying 8 coffees, which were not used up for a bronze voucher
        - Free special coffee
    - Gold: Given when buying 10 coffees, which were not used up for a bronze or silver voucher
        - Free large special coffee

- End of Coffee to Go assignment -

Gardener

The gardening company EverGreen™ wants to keep track of the gardens of their customers. By keeping a record of all the different types of plants, including their blossom and prune periods, the best time to do maintenance in the garden can be calculated.

To help you on your way, there is an example UML diagram show below. This diagram consists of classes only, so it is not complete.



Garden details example	
Customer	Piet Jansen
Plants	Rhododendron Oak tree Palm tree Palm tree Fern

Plant details example	
Plant	Rhododendron
Color	Red
Evergreen	Yes
Type	Bush
Blossom periods	List of periods
Prune periods	List of periods

Blossom period example	
Start	April
End	May

Prune period example

Start	August
End	September

This app should at least be able to do:

- Add a garden to the system.
- Remove a garden from the system.
- Register a plant into a garden.
- Add blossom and prune periods to the plant.
    - Take note that plants can have multiple blossom and prune periods.
- Remove a plant from the garden.
    - Take note that when removing a plant, its blossom and prune periods need to be removed too.
- Select the plants of a selected garden, that can be pruned during a given month/period.
- Save and load data to (a) file(s).
- Calculate when the garden is at its blossom peak. This is when the most plants of the garden are within their blossom period.
    - If multiple periods apply, it is OK to select the first one that matches the conditions.

Optional:

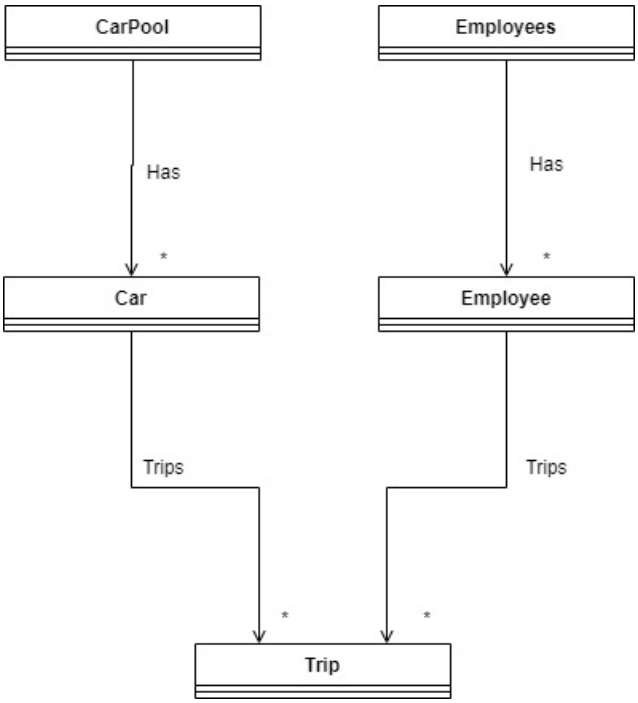
- In addition to files, save and load all data from a database.
- Search a garden which can be pruned at a given time.
    - When multiple gardens apply, it is OK to select the first one that matches the conditions.
- Generate a monthly top 3 of 'prettiest gardens', being the gardens, which have the most plants blossoming that month.

- End of Gardener assignment -

Company Car Sharing

Company TheGreenMile™ has a pool of electric cars, which employees of the company are allowed to use for business trips. However, it is important for both the government, and for the maintenance of the car, that the trips get logged. It is your task, to create an application that can log the trips of each employee for each car.

To help you on your way, below you find a nice starting point UML diagram, which you can change or extend. It only shows the classes, so it is not complete.



Car details example

Brand	Toyota	
Model	Prius	1.6 hybrid
End of life mileage	160000	km
Maintenance Interval	15000	km
Current Mileage	23560	Km
License plate number	1-FHICT-2	

Trip details example

Employee	Piet	Jansen
License plate number	1-FHICT-2	
Start Mileage	23490	Km
End Mileage	23560	Km
Private trip	No	

This app should at least be able to do:

- Add a car to the company's pool of cars.
    - Take note, that it should be impossible to add a car with the same license number twice.
- Search for a car by type and/or license plate number.
- Remove a car from the pool.
- Register an employee.
- Remove an employee from the system.
- Add a trip into the system, connected to a user and a car.
- Save and load data to (a) file(s).
- Automatically update the total mileage of the car when a trip is added.

Optional:

- In addition to files, save and load all data from a database.
- Calculate the total amount of mileage an employee has done.
- Calculate the percentage of private mileage the employee has done.

- End of Company Car Sharing assignment -