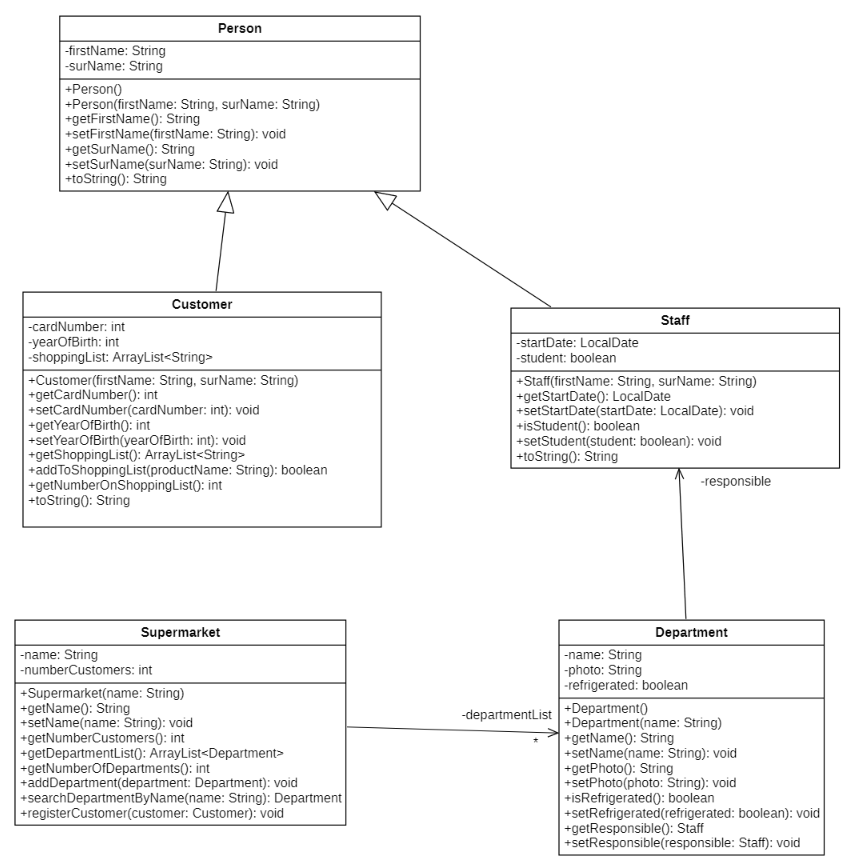
**Project supermarket – part 1**

We are making a web project related to a supermarket.

**General**

* Download the zip-file *Start folder project\_Supermarket.zip* on your local disk
* Unzip and rename the folder of the project (in Explorer) using your own name so e.g:  
  Broos\_Jan\_r0670812\_Supermarket
* Make your code readable, maintainable and efficient
* Create the classes in the package *model* according to the given UML Diagram. No more and no less. **You are not allowed to create additional or other attributes or methods than the one you find in the UML Diagram**...
* Make a **copy / backup** of your project on a regular basis if you have finished a part of this assignment, so that you can fall back on it if something has gone wrong...
* **Add your name and student number in each file in comments.**
* **Please note****:** we know that a lot of information/code is exchanged between students to help each other. This is absolutely no problem when doing the exercises. However, the project has PE points, so must be made **INDIVIDUALLY/PERSONALLY**.
* So if you use/copy someone else's code or you help someone with the code, we can no longer assess someone’s knowledge/capabilities individually. This is a form of fraud: both for the person copying and for the person who has copied. See OER (Education and Examination Regulations). Make sure that you make your project individually and do not "share" your code with other students. So do not put your code on "social media/shared cloud" to help each other and do not use it there. The Java teachers are there to help everyone...

**The class diagram:**



In the package **fact.it.supermarket.model,** create the classes **Person,** **Customer,** **Staff, Department** and **Supermarket** as shown in the above class diagram. Don't forget to add your name and student number to each of these classes in comments.

Unit tests are provided for all classes, use these to test if your code complies with what is asked for.

Additional information for the class **Person**:

* The **toString** methode returns the following String for the person with first name Juul and surname Mertens (surname in capitals):
  + MERTENS Juul

Additional information for the class **Customer**:

* + - You can initialise the attribute *shoppingList*when declaring the attribute. You do this as follows:

*private ArrayList<String> shoppingList**= new ArrayList<> ();*

* + - The *card number* of the customer is always set to -1 when a new customer is created.  
      * The shopping list may contain a maximum of 5 product names. The method *addToShoppingList*only adds the product name to the shopping list if it does not yet contain **5** product names.
        + If the product name could be added, the method returns *true***.**
        + If it could not be added, the method returns *false*.
    - The method *getNumberOnShoppingList*shows the number of products stored in the ArrayList.
    - The *toString*method returns for the customer with first name Ans, surname Verelst and customer card number 3 the next String:

Customer VERELST Ans with card number 3

For the implementation of this functionality use the *toString* method of the superclass **Person**!

Additional information for the **Staff** class:

* + - The attribute *startDate*is the date of creation of the staff member.
    - The boolean *student* indicates whether this employee is a working student or not.
    - The *toString*method returns this information:

Staff member LAMBRECHTS Jef (working student) is employed since 01/02/2022.

or

Staff member VOLDERS Bie is employed since 16/03/2022.

For the implementation of this functionality use the *toString* method of the superclass **Person**!

There is no additional information needed for the **Department** class.

Additional information for the **Supermarket** class:

The association between *Supermarket*and *Department*indicates that a Supermarket may contain several Departments. We model this by keeping an ArrayList of Departments in the supermarket. You initialise this ArrayList when declaring the attribute as follows:

*private ArrayList<Department> departmentList = new ArrayList<> ();*

* + - The method *getNumberOfDepartments*shows the number of departments stored in the ArrayList.
    - Using the *addDepartment*method, you can add an department to the ArrayList.
    - Using the *searchDepartmentByName*method, you have to compare the parameter *name* with the name of every department in the attribute *departmentList*. If there is a match, you have to return that department-object. If there is no department with the specified name, ***null*** will be returned. (be careful: Strings are compared with *“.equals(…)*”).
    - The method *registerCustomer*registers the given customer at a certain supermarket. With this registration, the attribute *numberCustomers*isincreased by 1. The cardnumber of that customer is also set to this number of customers and thus indicates the number of registered customers she/he is at this supermarket.

**Run the tests.**