

# Student System Part I

The assignment covers the following learning goals:

- You can apply inheritance in your implementation;
- You can recommend an application design and implementation by making use of multiple classes.

Question difficulty: ★★★★★

## Case description

This assignment consists of several parts, which will be implemented from this week till week 3.

You are going to develop an application for a school like Fontys. The persons, who are involved in the school are students, teachers, administrative employees, employees working in the catering, etc.

For now we will consider only 2 kinds of employee, but in future it should be easy to extend the application for more kinds.

For every person we would like to store some information, like their unique pcn, their name, age and how long the person has been at the school. Furthermore, there should be the possibility to register that a person celebrates his/her birthday and starts another schoolyear.

It will be your task to implement this application. Surprise, Surprise, you will be using inheritance.



Not all constructors, properties and methods are included in the UML Class Diagram. Decide for yourself what is really needed and include as you progress with this assignment.

The same applied for the access modifiers. You will notice that you will want to change the private access modifier to implement the requested features.

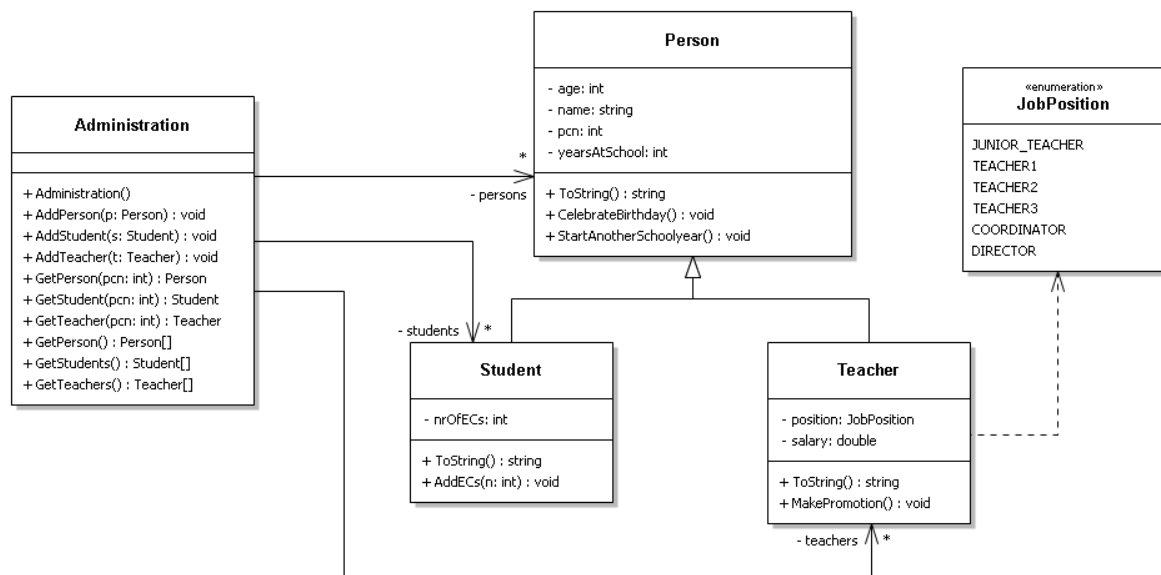
## User interaction

It should be possible for a user of the application to:

- Add a new person
  - The user provides the required information and confirms;
  - The system adds the person;
    - This should only be possible if no other person, student or teacher has the same PCN
- Show all persons
  - The system displays all persons;
- Show info of the person
  - The user provides a PCN and confirms;
  - The system displays found person;
    - When no person is found a message should be shown.

- Add a new student
    - The user provides the required information and confirms;
    - The system adds the student;
      - This should only be possible if no other person, student or teacher has the same PCN
  - Show all students
    - The system displays all students;
  - Show info of the student
    - The user provides a PCN and confirms;
    - The system displays found student;
      - When no student is found a message should be shown.
  - Start a new school year for the student
    - The user provides a PCN and confirms;
    - The system increases the found student's school year by one;
      - When no student is found a message should be shown.
  - Add EC's to a student
    - The user provides a PCN & ECs and confirms;
    - The system increases the found student's ECs;
      - When no student is found a message should be shown.
  
  - Add a new teacher
    - The user provides the required information and confirms;
    - The system adds the teacher;
      - This should only be possible if no other person, student or teacher has the same PCN.
  - Show all teachers
    - The system displays all teachers;
  - Show info of the teacher
    - The user provides a PCN and confirms;
    - The system displays found teacher;
      - When no teacher is found a message should be shown.
  - Start a new school year for the teacher
    - The user provides a PCN and confirms;
    - The system increases the found teacher's school year by one;
      - When no teacher is found a message should be shown.
  - Promote the teacher
    - The user provides a PCN and confirms;
    - The system promotes the found teacher's;
      - When no teacher is found a message should be shown.
-

## UML & Screenshots



UML CLASS DIAGRAM 1: INCOMPLETE ADMINISTRATION DIAGRAM

Administration	
Constructors & methods	Description
<b>AddPerson</b> <b>AddStudent</b> <b>AddTeacher</b>	Add the person, student or teacher to the correct collection. Note the requirement of an unique PCN.
<b>GetPerson</b> <b>GetStudent</b> <b>GetTeacher</b>	Return the person, student or teacher with the PCN, if there is none return <i>null</i> .

Person, Student & Teacher	
Constructors & methods	Description
<b>ToString</b>	Should all return a string with all the information of the person, student or teacher.
<b>CelebrateBirthday</b>	Increase the age by one
<b>StartAnotherSchoolyear</b>	Increase the <i>yearsAtSchool</i> by one.  Furthermore, after every 3 years a teacher will get a salary raise of 10%. So, after 3 years at school, a salary raise of 10%; after 6 years again a raise of 10%; after 9 years again, etc. . .

Student	
Constructors & methods	Description
<b>AddECs</b>	Increases the EC's by <i>n</i>

Teacher	
Constructors & methods	Description
<b>MakePromotion</b>	if possible, change the <i>position</i> of a teacher to the next one. The increasing order of position are junior-teacher, teacher1, teacher2, teacher3, coordinator and director.

The *position* does not change if the teacher is already a director.

The screenshot shows a window titled "Form1" with standard Windows window controls (minimize, maximize, close). The interface is divided into several sections:

- Add form section:** Contains input fields for "PCN:", "Name:", "Age:", and "Year at school:". To the right of "Year at school:" are fields for "EC's:" and "Salary:". Below these fields are three buttons: "New person", "New student", and "New teacher".
- Show all section:** Below the "Add form" section are three buttons: "Show all persons", "Show all students", and "Show all teachers".
- Person form section:** A sub-form with a "PCN:" field and a "Show info" button.
- Student form section:** A sub-form with a "PCN:" field, a "Show info" button, a "Start new schoolyear" button, a small empty input field, and an "Add ECs" button.
- Teacher form section:** A sub-form with a "PCN:" field, a "Show info" button, a "Start new schoolyear" button, and a "Promote" button.

A large empty rectangular area is located between the "Show all" buttons and the individual form sections.

FIGURE 1: ADMINISTRATION GUI

#### Additional features

Add and implement a celebrate button for a person