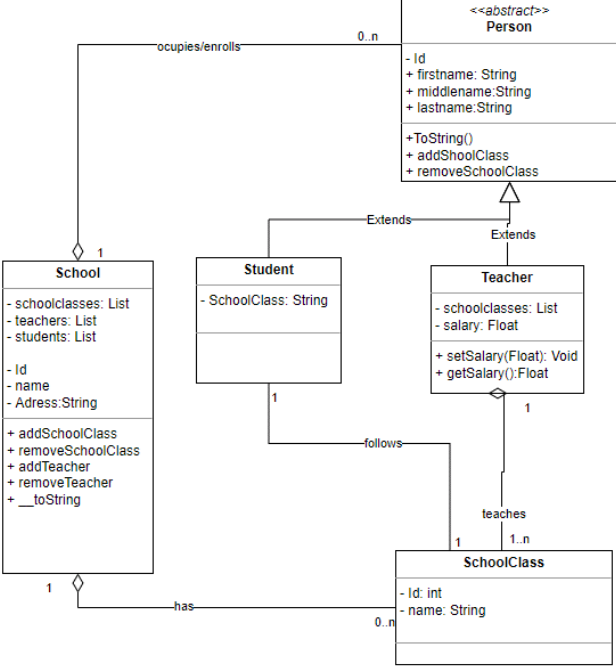


## Opgave: Testing a Model Class

<p>Koden som vi forlod den tirsdag d. 5/4 implementerer en model som vist her.</p>	 <pre> classDiagram     class Person {         &lt;&lt;abstract&gt;&gt;         -Id         +firstname: String         +middlename: String         +lastname: String         +ToString()         +addShoolClass         +removeSchoolClass     }     class School {         -schoolclasses: List         -teachers: List         -students: List         -Id         -name         -Adress: String         +addSchoolClass         +removeSchoolClass         +addTeacher         +removeTeacher         +__toString     }     class Student {         -SchoolClass: String     }     class Teacher {         -schoolclasses: List         -salary: Float         +setSalary(Float): Void         +getSalary(): Float     }     class SchoolClass {         -Id: int         -name: String     }      Person &lt; -- Student     Person &lt; -- Teacher     School "1" *-- "0..n" SchoolClass : occupies/enrolls     School "1" *-- "1" Student     Student "1" -- "1" Teacher : follows     Teacher "1" *-- "1..n" SchoolClass : teaches     </pre>
<p>Koden kan downloades fra dette link</p>	<p><a href="https://github.com/kimo1ucl/PHP-OOP-Opgave-Testing-A-Model-Class">https://github.com/kimo1ucl/PHP-OOP-Opgave-Testing-A-Model-Class</a></p>
<p>"Arkitektur"</p>	<p>I nuværende form findes abstraktions klassen <b>Person</b>, der findes også 2 afledte klasser: <b>Student</b> og <b>Teacher</b>.</p> <p>Vores <b>School</b> klasse er blevet en slags manager for systemet, her finder vi lister af students, teachers, og schoolclasses.</p> <ul style="list-style-type: none"> <li>• Alle objekter oprettes via metoder i <b>School</b> klassen som er den eneste php klasse fil vi behøver reference til, for at teste hele systemet.</li> <li>• Metoder i <b>School</b> klassen returnerer om metoden lykkedes med at gennemføre det den blev sat til.</li> <li>• Alle lister kan leveres af metoder i <b>School</b> klassen, det benytter vi i test sammenhæng, ved at udlæse en liste af objekter vi ønsker skal anvendes i test.</li> <li>• Hvis vi feks. ønsker at teste teatchers og metoder for at assigne schoolclasses, henter vi alle objekterne i 2 lister, og vælger vores testgrundlag ved at hente objektet via reference i testscriptet.</li> </ul>
<p>Opgave</p>	<p>Vi ønsker at udvide vores "framework", sådan at problemområdet udvides på en måde så ansatte beskæftiget som feks. Uddannelseschef (Bo Stærke) også skal "håndteres" i systemet.</p> <p>I diagrammet har vi kaldt ledere som uddannelseschefer <b>Manager</b>. Managers kan være et lidt upræcist ord, hvis vi tog alle ledelses stillinger på UCL, vil det næppe om dem alle kunne siges, at de leder en mængde skoleklasser og en mængde undervisere. Men for nu lader vi videre spekulationer ligge.</p>
<p>Krav</p>	<p>Man skal kunne teste managers på samme måde som vi tester teachers og students. Den nye klasser skal afledes af super klassen Person</p>

## Opgave: Testing a Model Class

Opdateret model

Modellen er ikke i alle detaljer ajourført.

feks. er der mange steder implementeret en getId(), og det fremgår ikke af modellen.

