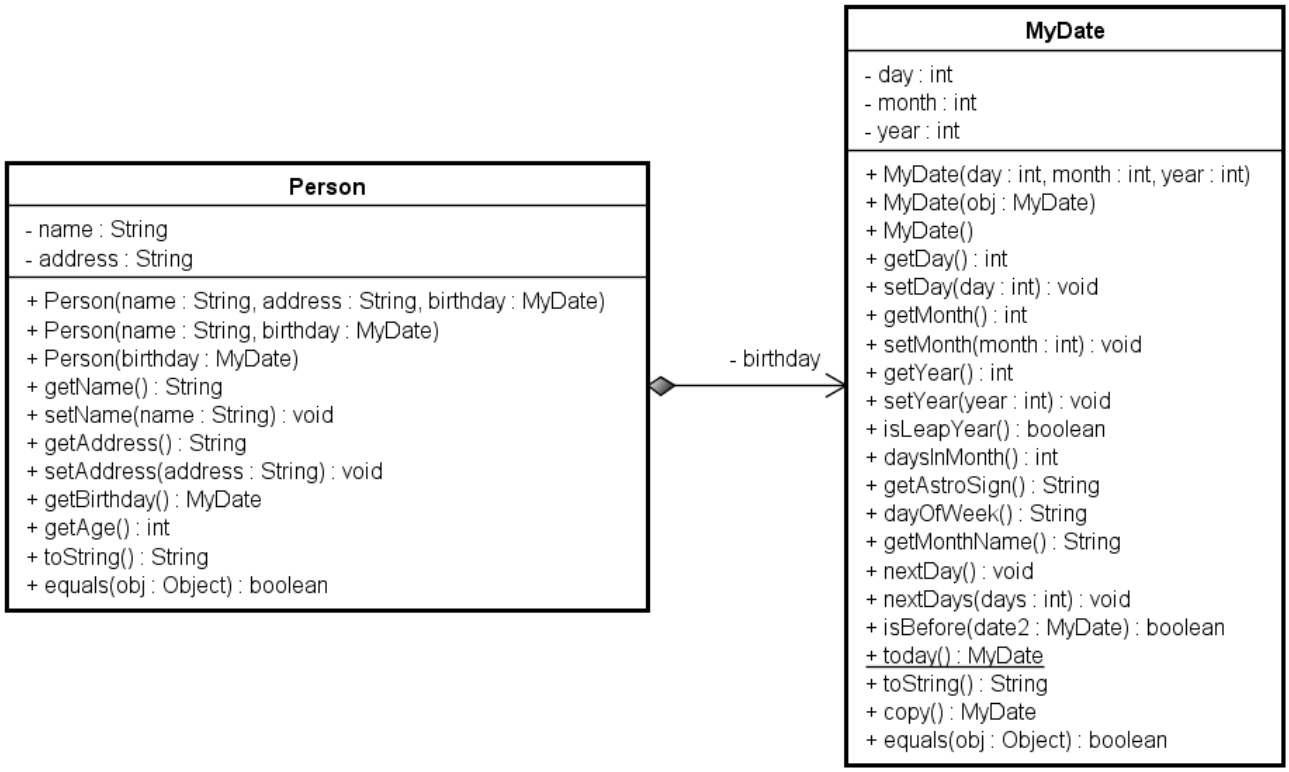
Programming Exercises - PRO1 - Session 13

Exercise 13.01

Those of you who have made all the exercises so far should now have a `MyDate` class containing the methods shown in the class diagram below. Some of the methods might be useful in this exercise, so if you have not made all of them yet, then this could also be a good time to do that.

Implement a class `Person` with the fields, constructors, and methods exactly as shown in the class diagram below.

When you are done then make a test class to test the functionality.



Exercise 13.02

Implement a class `Garage` representing a garage that can hold up to two cars. It should have:

a) 2 instance variables of type `Car` (from Exercise 12.02), representing the two cars parked in the garage.
b) A no-argument constructor. Set the two `Car`-object references to something representing no cars in the garage (`null`).
c) A boolean method `isParkingAreaTaken(int position)` that returns true if a car is parked at the position given by the argument passed to the method (position can only be 1 or 2).
d) A void method `park(Car car, int position)` that parks a car in the position given by the parameter (position can only be 1 or 2). If there is already a car parked in that position the car cannot be parked there.
e) A method `leaveGarage(int position)` that simulates driving a car from the parking area leaving the position empty. The method should return the reference to the Car-object that was on the position given by the parameter. If there is no car parked in that position the method return `null`.
f) A toString method returning all information in a string.
g) An equals method returning `true` if the object passed as argument is a `Garage` object with the same set of cars parked there, otherwise false. Note: references to objects could be `null`.

Exercise 13.03

Implement a test class with a main method and test your solution for `Garage` and `Car`.

Exercise 13.04

Draw a UML Class Diagram of the classes `Car` and `Garage`. What should the relationship between the two classes be? Association, aggregation, or composition?