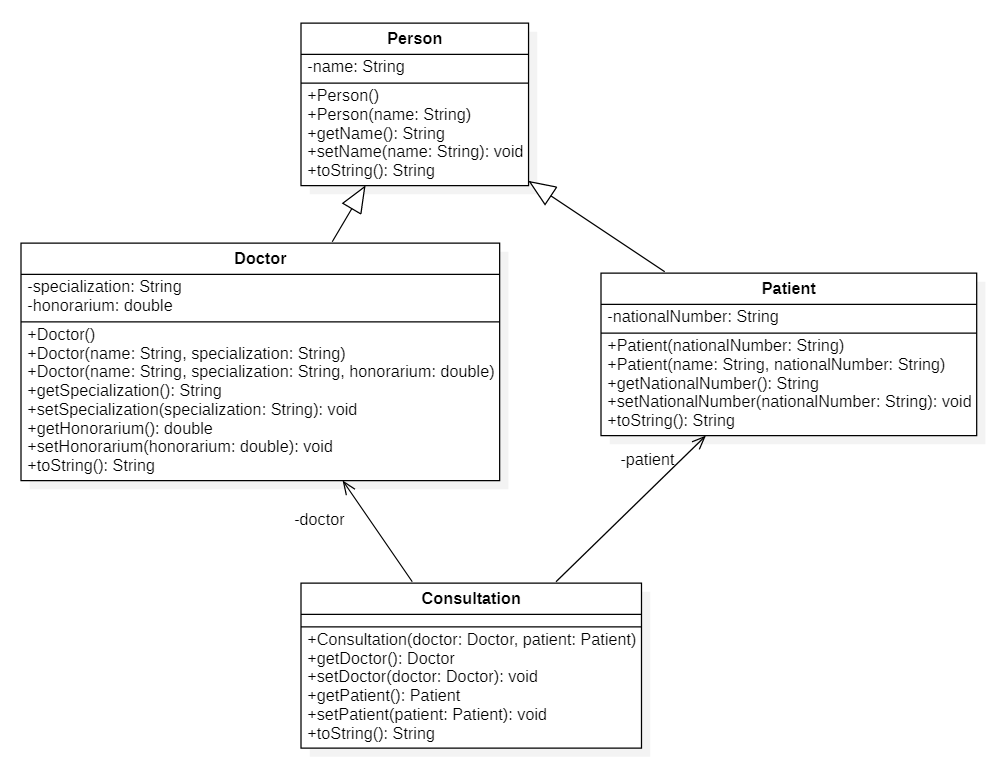
**Exercises Debugging Part 2**

**For all exercises:**

* Use the start folders on Canvas
* Debug the application projects without looking at their previous solutions. These exercises are here to improve your debugging skills, so please treat them in that way.

**Exercise 1**

Someone tried to build the following solution yet made quite a number of mistakes. Open the “**exercise-patient**”-project and fix the errors to make sure it adheres to the instructions below:  
  
Create the following classes and provide them with the required attributes and methods.



Below are the results of the *toString*() methods in the different classes. Use the *toString*() method of the superclass if possible.

Person: Name: Miet Adriaens

Patient: Name: Peter Boonen (national number: 850412 204M14)

Doctor: Dr. Myriam Dierick, Surgeon

Consultation: Doctor: Dr. Leo Lebon, Cardiologist

Patient: Name: Tom Verlaet (national number: 750510 105M16)

Consultation: 2021-01-07

For the abovementioned *toString* you should use *the current date*

**Make sure that all tests are working successful**

**Exercise 2**

Someone tried to build the following solution yet made quite a number of mistakes once again. Open the “**exercise-renovation**”-project and fix the errors to make sure it adheres to the instructions below:

For your information:

The method *getAmountOfPaintInLitres*() indicates how many litres of paint you need.   
  
Tips:

* + area of all walls in the room = 2 \* (length + width) \* height
  + ceiling area = length \* width
  + per 10m² you need one litre of paint

**Make sure that all tests are working successful**

**Afbeelding met tekst

Automatisch gegenereerde beschrijvingNow run the web application.**

The start of the web application should look like this and if you enter not only your name but also all the information needed to calculate how much paint you will need.

Afbeelding met tekst

Automatisch gegenereerde beschrijving  
this should be the result...

The *Back* hyperlink takes you back to the homepage **index.html.**

if you check also the paint ceiling…:

Afbeelding met tekst

Automatisch gegenereerde beschrijving

this should be the result...

Afbeelding met tekst

Automatisch gegenereerde beschrijving

**Exercise 3**

Someone tried to build the following solution yet made quite a number of mistakes once again. Open the “**exercise-course**”-project and fix the errors to make sure it adheres to the instructions below (all tests should run already…):

After starting the web application, you should be able to fill in 4 text boxes and indicate whether the course is a weekly course, to then press the *Register a new course* button to submit the form.

Afbeelding met tekst

Automatisch gegenereerde beschrijving

On **courseinfo.html** the following result should get displayed:

Afbeelding met tekst

Automatisch gegenereerde beschrijving

The *Back* hyperlink takes you back to the homepage **index.html.**

**Exercise 4**

Someone tried to build the following solution yet made quite a number of mistakes once again. Open the “**exercise-refuelling**”-project and fix the errors to make sure it adheres to the instructions below (all tests should run already…):

Afbeelding met tekst

Automatisch gegenereerde beschrijvingWhen you start the web application, the index-page should look like this*:*

Afbeelding met logo

Automatisch gegenereerde beschrijvingIf you fill out the text boxes using the example values and press *Calculate consumption*, the next page should look like this…

The *Back* hyperlink takes you straight back to the homepage **index.html.**

**Exercise 5**

Someone tried to make a web application, **primedebug**, which could’ve been used to find all the prime numbers up to a specific value. ​(Read up on prime numbers here: [EN](https://thirdspacelearning.com/blog/what-is-a-prime-number/) - [NL](https://www.beterrekenen.nl/website/mob.php?pag=248)).

Afbeelding met tekst

Automatisch gegenereerde beschrijving

To support this webpage someone created a **IsPrime** method which takes a number as an argument and returns a boolean to show if that number is a prime number or not. This **IsPrime** method code checks:​

* whether the number is lower than or equal to 1 (in which case it is not a prime number)​
* whether it’s 2 (which is the only even prime number)​
* whether it’s even (in which case it’s not a prime number)​

So, when the number is not lower than or equal to 1, is not 2 or is not even, it’s obvious the number is an odd number greater than 2.

In this case, the method’s code simply loops over all the values between 3 and the square root of the number itself. This is sufficient to determine if the number is a prime number.​

Now this application still has problems, fix them step by step:​

* Solve Thymeleaf and HTTP errors at runtime. After this the app should at least run.
* Print out all the primes up to 100. Unfortunately, the code doesn’t seem to work correctly, it reports 9, 15, 25, etc. as prime numbers, even though they clearly are not.​ However there does not seem to be an obvious problem with the code. You must debug the application to check the IsPrime method and its loops.

A correctly working application should show the following prime numbers when one enters 15:

Afbeelding met tekst

Automatisch gegenereerde beschrijving Afbeelding met tekst

Automatisch gegenereerde beschrijving