Testing and Debugging part 1

Debugging…

Introduction

- Testing and Debugging = locating and fixing errors
- 5 types:
  - opening and installation errors:
    - already visible during the opening of a project
  - compilation errors:
    - during programming
  - testing errors:
    - tests do not run
    - tests do not succeed
  - start-up errors:
    - when you want to run your programme, but it does not start
  - runtime errors: during the execution of your program:
    - your programme runs and crashes at the start or during execution
  - logical errors: the programme does not give the expected / logical result
- Debugger
  - can help you detect runtime errors and logical errors
    - **not** for installation errors, compilation errors and start-up errors
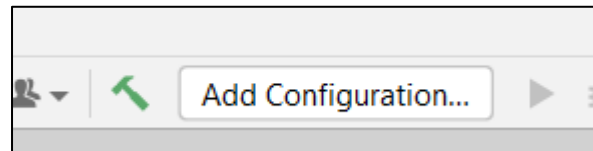
Opening error

- Opening error appears when opening a project that is not a project:
  - Open IntelliJ
  - Choose File>Open…
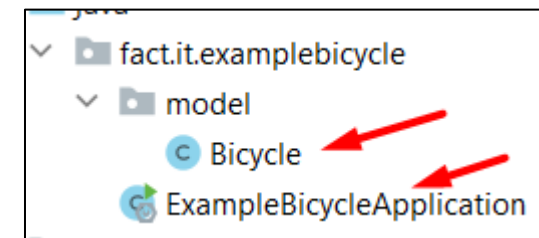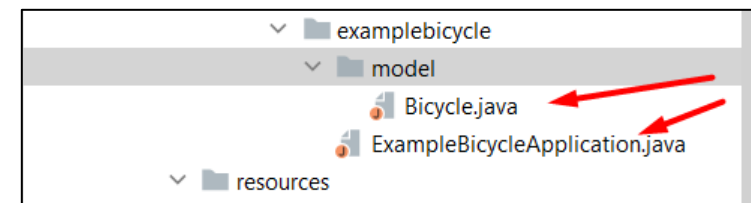 - Click on <u>a project</u> with the special icon:
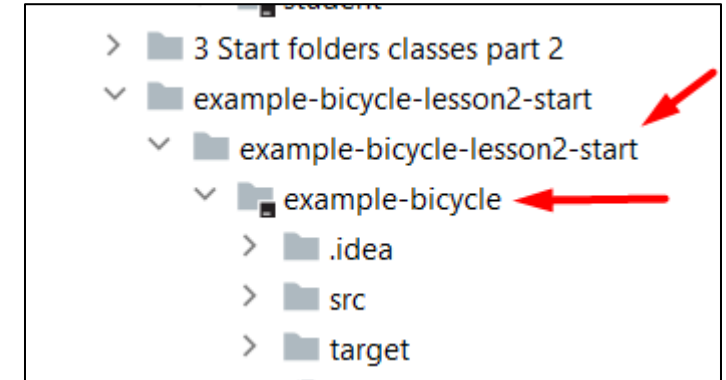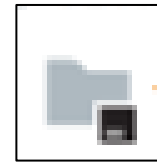
- If not clicked on
  - =>
  - in stead of…
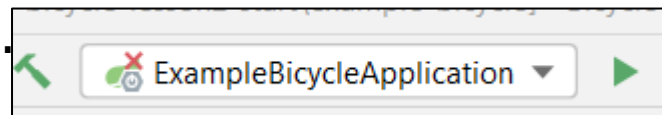
Installation error

- Following problems may appear when the project you want to open has a different JDK-version than the one of your IntelliJ-installation:
  - In case you get the following error, follow the steps below and on the next slide:
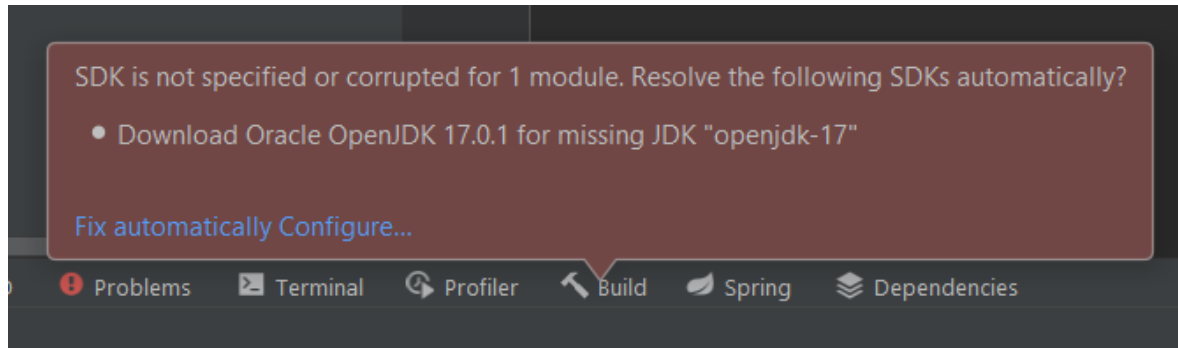
  - Click on 'Fix automatically'
  - Wait until all downloads have finished (this might take some time), you can follow the progress by looking at the progress bar in the bottom right corner

Installation error

- When the download of the newest Java SDK has finished, click on File > Invalidate Caches…, next click on 'Invalidate and Restart'
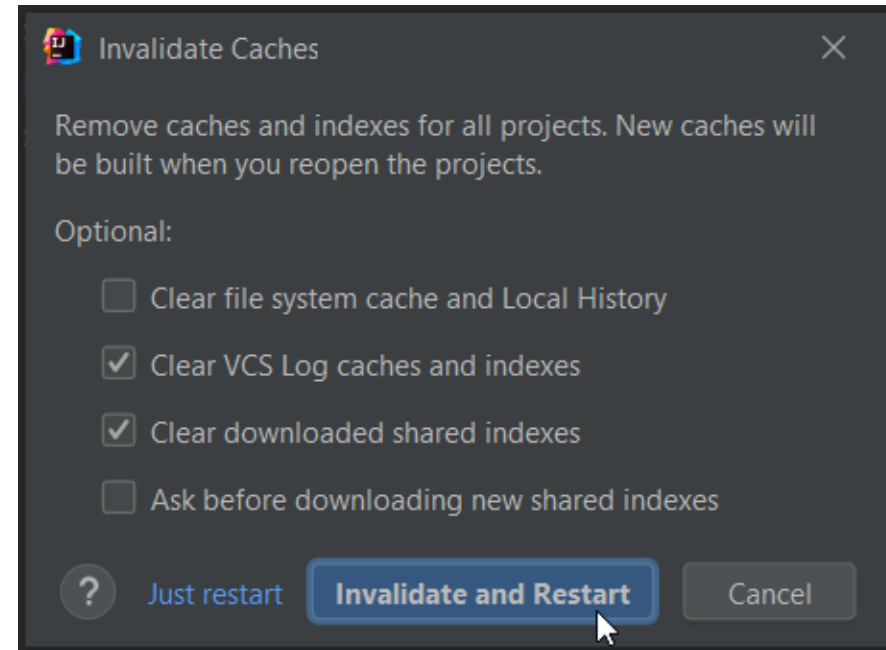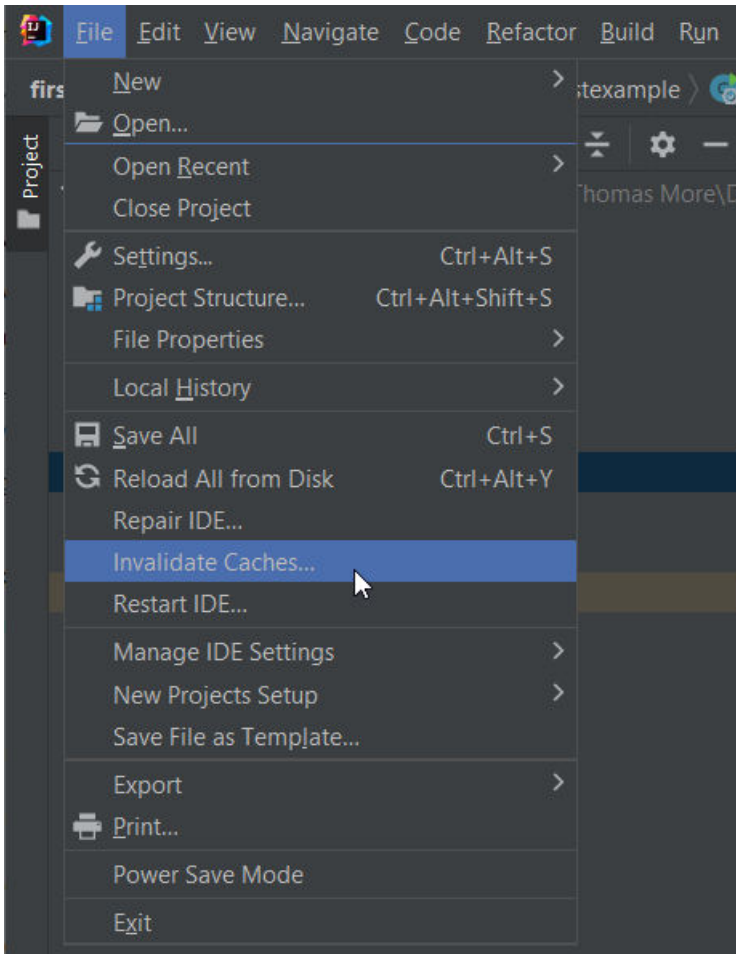
Installation error

- ## If 'Fix automatically' doesn't work then

  - Choose from menu *File* then *Project Structure*

  - At *Project SDK* choose *add SDK*.

Installation error

- If you get the error that 'jdk is missing' and 'Fix automatically' doesn't work then
  - Choose *Download JDK* and take the correct version (17!) and Vendor *Oracle OpenJDK 17*.

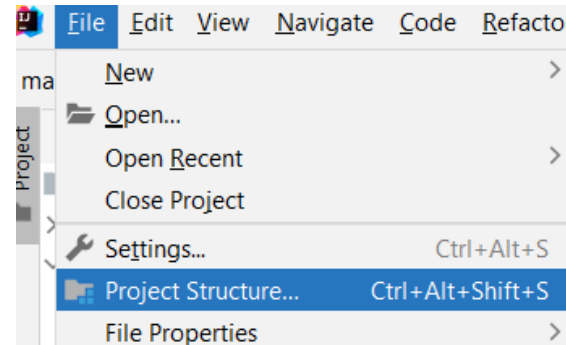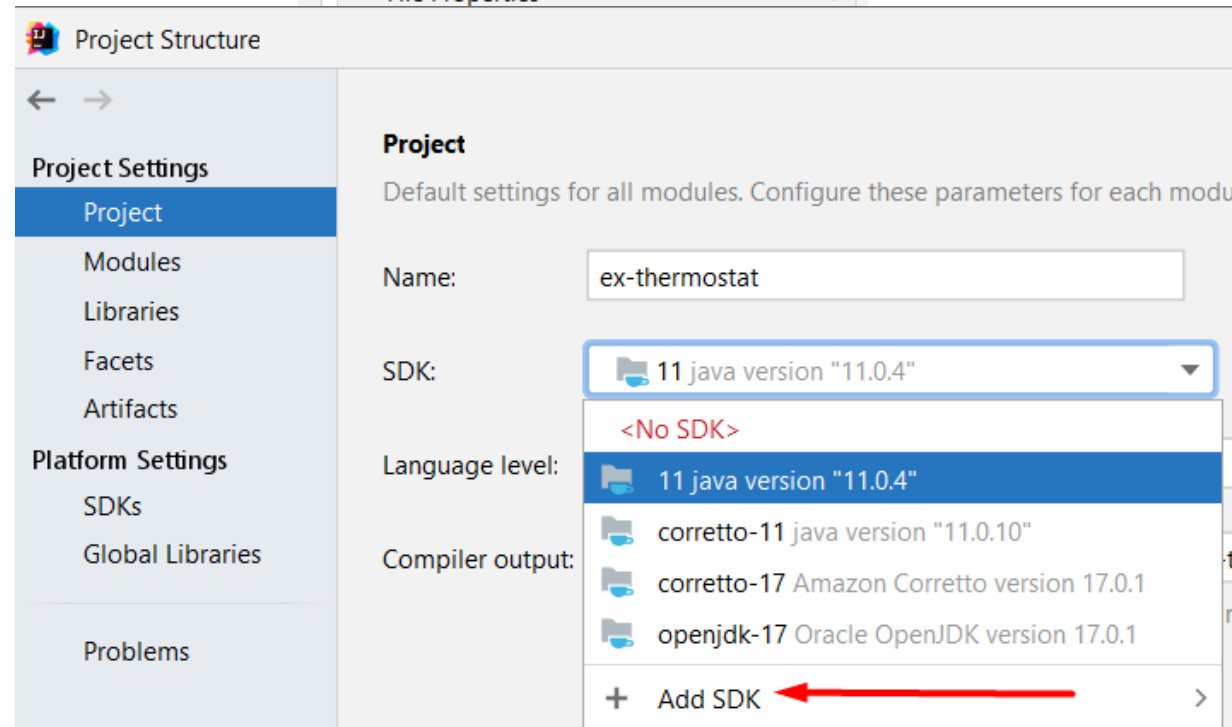 - Click *Download* and wait until installation is ready...
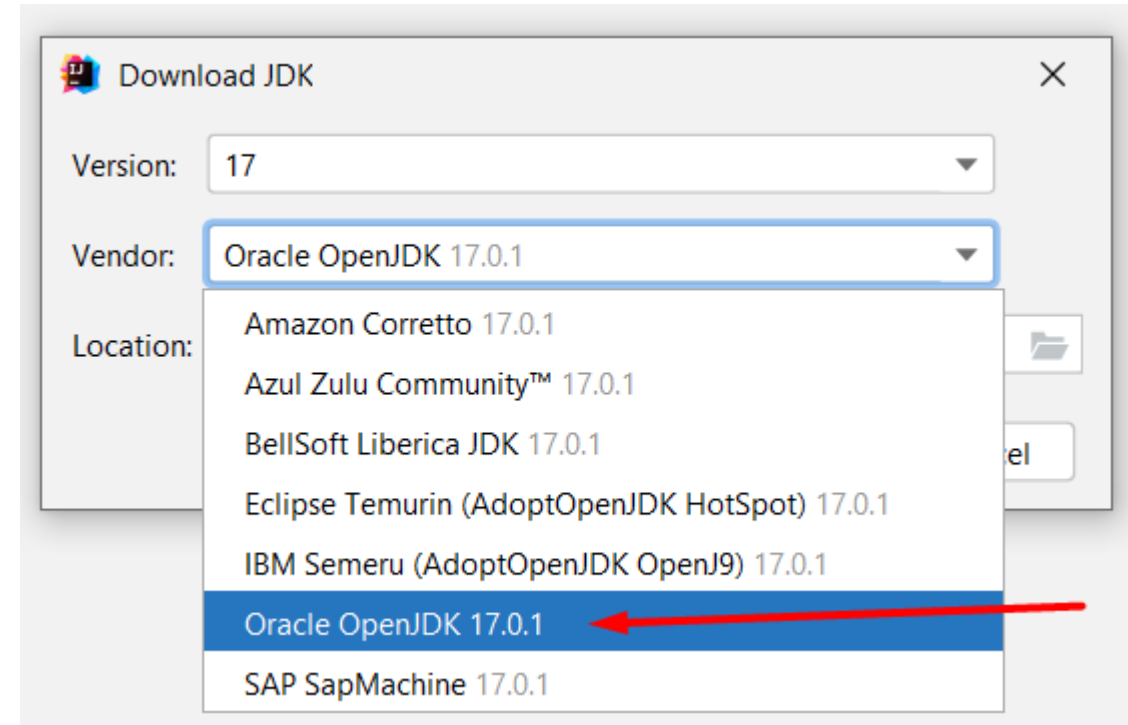
Compilation errors

- Knowing WHAT is wrong is often half the solution:

- READ the red text accompanying the error and try to understand it.

- <u>Do not blindly carry out the solutions suggested by IntelliJ</u>. This often causes more problems…

- Some common examples:

```
etPricePerYear() {
ePerYear = rentalPrice * 12
2014) {
rYear *= 0.95;
```

';' expected

Insert ; Alt+Shift+Enter   More actions… Alt+Enter

      • Solution: add a ";" at the end of the line…

Compilation errors

- Some common examples:


```
getInitials("RA") + ")";
```

'getInitials()' in 'fact.it.examplebicycledebugging.model.Student' cannot be applied to '(java.lang.String)'

Add 'String' as 1st parameter to method 'getInitials'   Alt+Shift+Enter   More actions...   Alt+Enter

- the method "getInitials()" does not have a parameter. 2 possible solutions:
    - => delete the parameter "RA" between the brackets
    - => change the method definition in the class so a parameter of the type String is required

    - Check the exercise to know what you have to do…

Compilation errors

- Some common examples:

```
Student student = new Student();
student.setForeigner("true");

System.out.println("Studen       Required type:    boolean
System.out.println("Full n
System.out.println("Email:       Provided:         String

                                 Wrap using 'Boolean.parseBoolean()'  Alt+Shift+Enter    More actions...  Alt+Enter
```

- the method requires a boolean but provides a String ("true"). Solution:
  - setForeigner(true)

Compilation errors

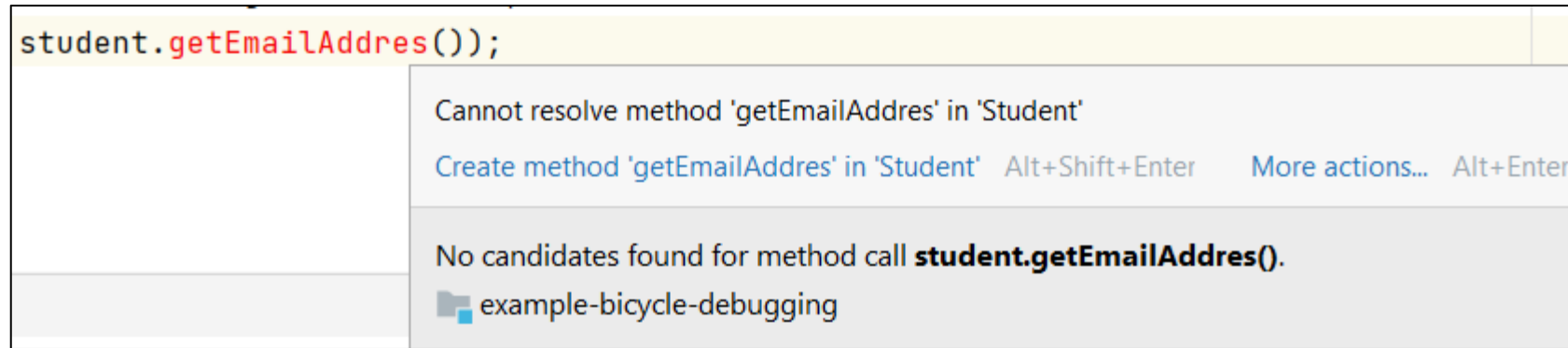- Some common examples:
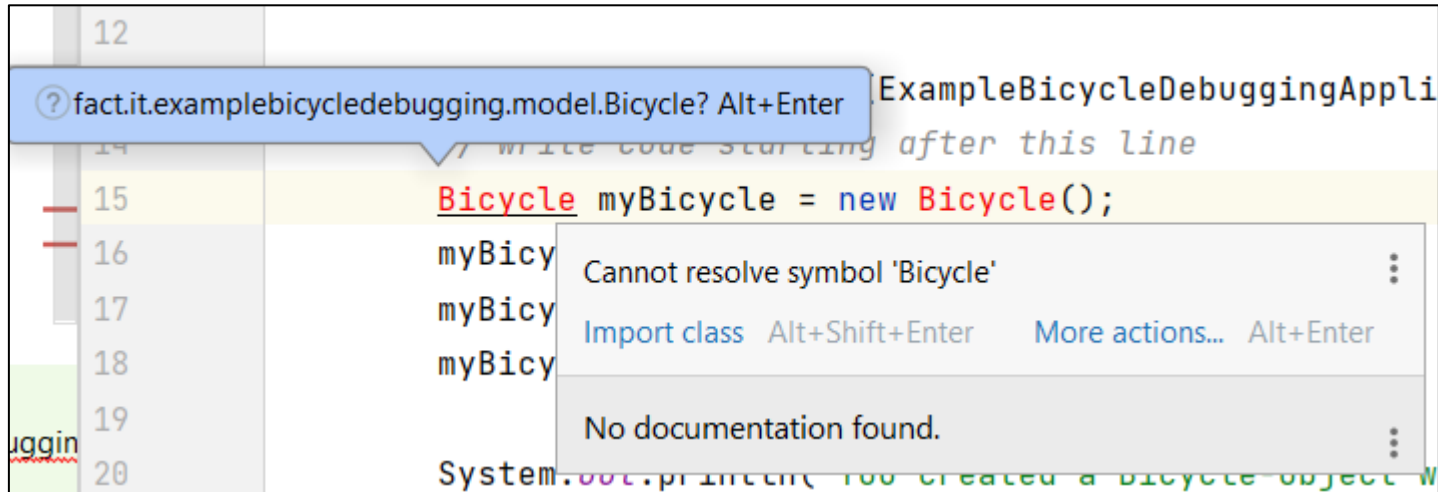
- the class "Bicycle" is not imported in the application
  Solution:
  - press Alt+Enter and the class will be imported by IntelliJ

Compilation errors

- Some common examples:


```
student.getEmailAddres());
```
Cannot resolve method 'getEmailAddres' in 'Student'

Create method 'getEmailAddres' in 'Student' Alt+Shift+Enter   More actions... Alt+Enter

No candidates found for method call **student.getEmailAddres()**.

example-bicycle-debugging

- the method "getEmailAddres()" does not exist. 2 possible solutions:
  - => check the name of the method. If it is the wrong name, correct it getEmailAddres**s**()
  - => create a new method with this name. You can let IntelliJ do this for you by pressing Alt+Shift+Enter BUT be careful! The compilation error will disappear but you possibly create errors later on...

Test errors – tests do not run

- If you haven't programmed all methods OR a wrong method's name is used somewhere in the program, the tests will not run.

- E.g.
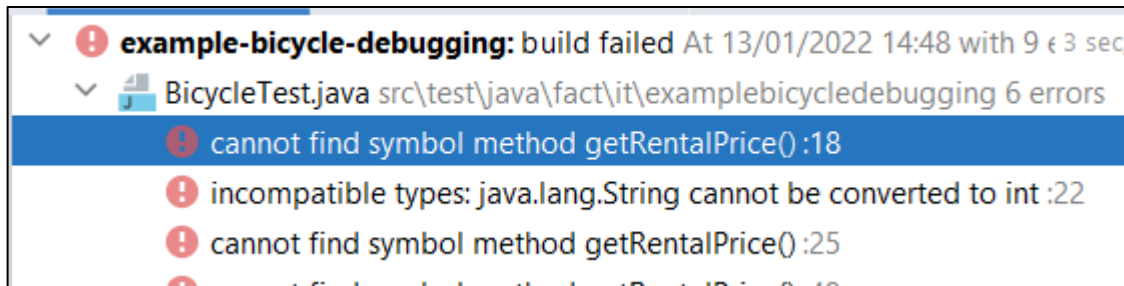
- Check the names of the methods you had to program:
  - do you have programmed ALL required methods?
  - do they have exactly the same name as required?
  - has the method you programmed the exact signature (type and sequence of the method's parameters) as required?
- NEVER change code in a test-file

Test errors – tests do not succeed

- If you coded all methods but you made a mistake in calculation or result:

- Correct the mistake by looking at the difference between what "Expected" and "Actual"

Starting error

- Application failed to start when you run the application:

```
****************************
APPLICATION FAILED TO START
****************************

Description:

Web server failed to start. Port 8080 was already in use.
```

- FIRST look at all open windows of IntelliJ. If one of these windows have a red icon at the top right: another web application is running. Click on the red square and try to run your application again…
- If this does not work => Change the port that your application uses in the application.properties file:

```
resources
> static.img
> templates
  application.properties
```

```
application.properties ×
1        server.port=8678
```

Runtime errors
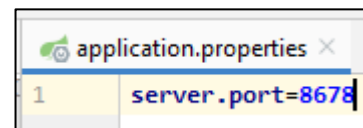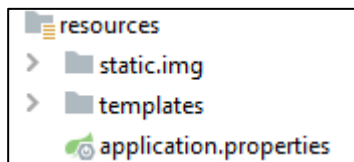
- A common example => NullPointerException:

```
Exception in thread "main" java.lang.NullPointerException Create breakpoint : Cannot invoke "String.toUpperCase()" because "this.surname" is null
    at fact.it.examplebicycledebugging.model.Student.getFullName(Student.java:53)
    at fact.it.examplebicycledebugging.ExampleBicycleDebuggingApplication.main(ExampleBicycleDebuggingApplication.java:36)
```

- NullPointerException occurs when a method is called from an object that is "null" (i.e., not instantiated)
- Solution:
  - Look/read which object = null and find the cause…
  - Make sure that the object is instantiated / gets a value… In this case "surname" is empty and needs to get a value.

Tips & tricks

- Googling "blindly" and doing what is suggested often causes more errors than before

- Only start "Googling" <u>AFTER</u>
  - you have read the error carefully and do not find the solution on that basis
  - you have not found the solution on the basis of the above

- "Smart" googling =
  - give **context** to the error:
    - IntelliJ
    - Java
    - Thymeleaf
    - MySQL
    - …
  - copy error message literally
    - links at the top
    - stackoverflow

Tips & tricks

- consulting fellow students / social media / ...
  - compared to the previous steps
    - increased risk of errors
    - less/no learning effect: this source cannot/should not be consulted during evaluation moments => risk of being judged as "fraud"
- official teaching channels (Canvas, Teams, Outlook Mail):
  - 100% sure of correct information
  - learning effect is greater