

# World of Robots

*Master 1 Génie Physiologique et Informatique*

## Del 2.4 : Synthesis



```
document.getElementById(div).innerHTML += '<div>';  
else if (i==2)  
{  
    var atpos=inputs[i].indexOf('@');  
    var dotpos=inputs[i].lastIndexOf('.');  
    if (atpos<1 || dotpos<atpos+1 || dotpos>inputs[i].length-1)  
        document.getElementById('errEmail').innerHTML += '<div>';  
    else  
        document.getElementById(div).innerHTML += '<div>';  
    dotpos=inputs[i].lastIndexOf('.');  
    atpos=inputs[i].indexOf('@');  
    if (atpos<1 || dotpos<atpos+1 || dotpos>inputs[i].length-1)  
        document.getElementById('errEmail').innerHTML += '<div>';  
    else  
        document.getElementById(div).innerHTML += '<div>';  
}  
if (i==5)  
    document.getElementById('errEmail').innerHTML += '<div>';
```

Group 14 : Laurène DONG, Marion DOSPITAL, Lauranne LAIR, Thibault NIGGEL, Florian PAILLAS, Ophélie SOCHARD

This synthesis is about our understanding of oriented-object design (OOD) and oriented-object programming (OOP) thanks to the practice in the “World of Robots Step 1” (WoR Step 1).

In the section “What is OOP/OOD?”, we learned about the perspective of the oriented-object applied in the IT field. We learned theory knowledge and now, we take our first step in the OOP by creating our first java programs.

All of these steps will help us to create our own interactive application of the “World of Robots”.

In the tutorial, we learn how to create a class with the attributes, setters, getters and constructor. We also define the limits and the implementation of the data associated and so on. This will help us to start with a good basis when starting the “World of Robot” application.

At first, we tried to understand the behavior of the “Ticket Machine Project”. We then learn with the “Ticket-machine project” how to define a class that stimulates behavior on objects.

With this project, we also learned to initialize the necessary data in a constructor.

We also got to use encapsulation by creating classes that group all the code that is concerned with a certain set of data in the same class.

We learned to use final and static class variables. A static variable means that all objects in the class can use or modify them while a final variable is a constant which cannot be modified once initialized.

Thanks to the different mini-projects that we had to do, we were able to understand the difference between class members and instance members. Class members are elements related to the class itself, not to a specific instance of that class. They are shared by all instances of the class. Examples: static variables, static methods. While instance members are elements specific to each instance of the class. Each body has its own values for its members. Examples: instance variable, instance methods.

The “AutomatedEvaluation” extension allowed us to check if the methods were two-step. If we had an error during the Current step, the latter provided us with the location of the error in order to help us correct it.

The second step was on the unit test, this step allowed us to check the correct functioning of the different requested functionalities.



As for the code itself, we learned to do different methods taking into account the different constraints such as constants, password security, functions to retrieve length, remove spaces too, or make all adjustments according to instructions. We can reuse these different methods as a basis for our project.