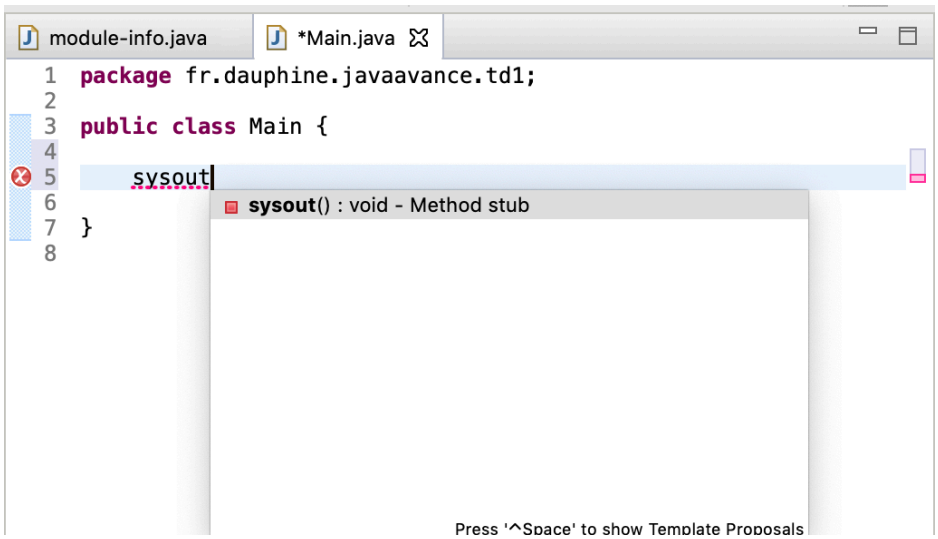
jeudi 30 septembre 2021

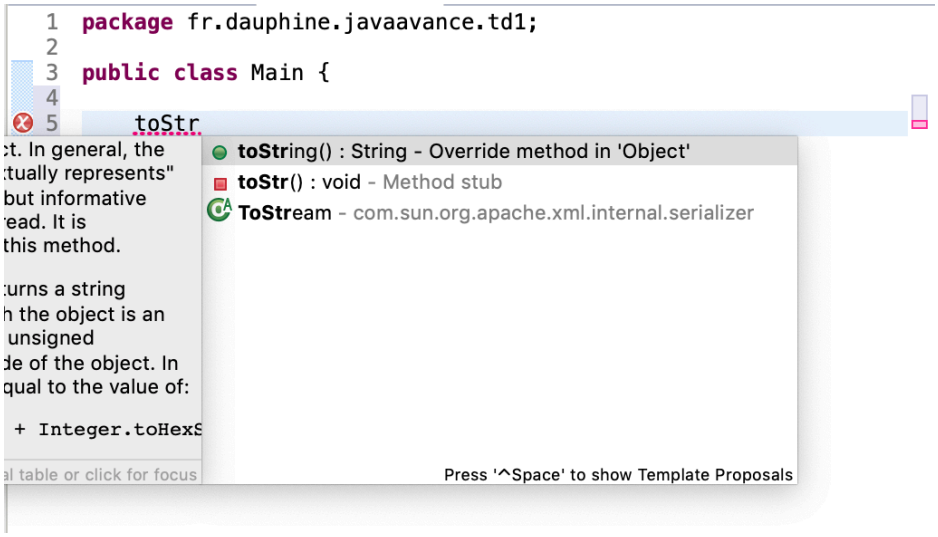
TD1

Exercice 1:

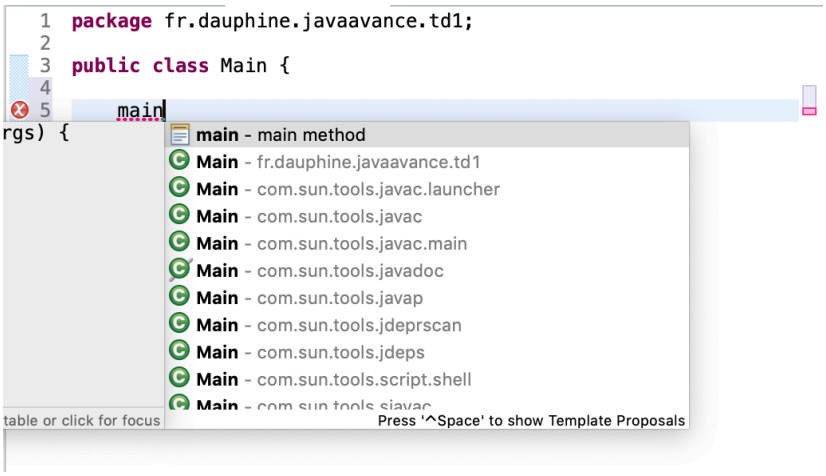
2) This is what happens when we type sysout and press Ctrl + space:



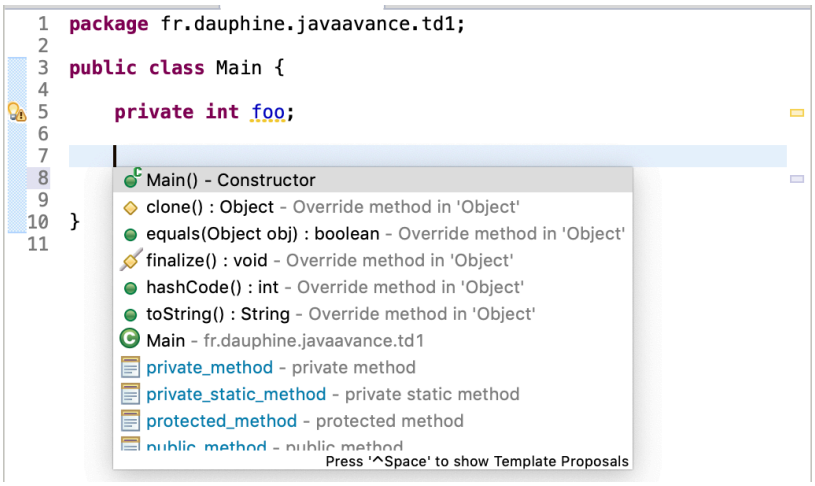
3) This is what happens when we type toStr and press Ctrl + space:



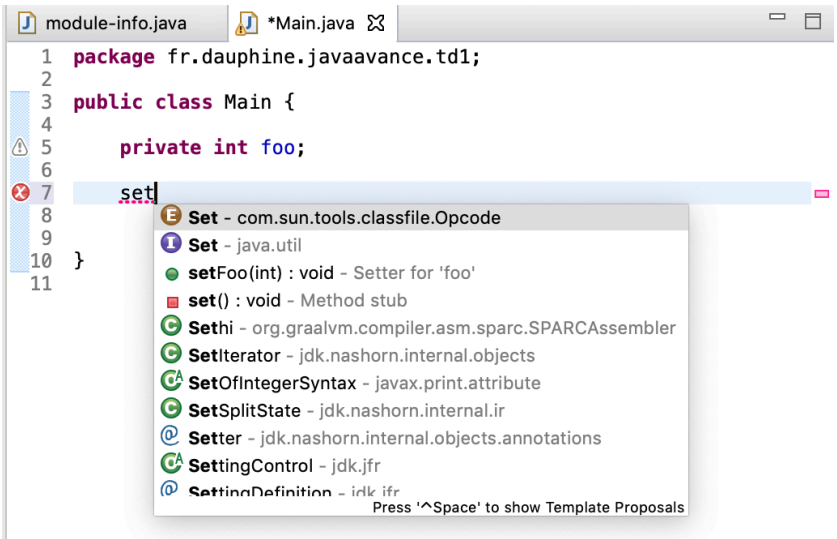
4) This is what happens when we type main and press Ctrl + space:



5) This is what happens when we type Ctrl + space inside the class after creating the field foo:



And this is what happens when we type set then press Ctrl + space:



6) When we type Alt+Shift+R (or option+command+R on mac) we can rename the class and the field.

Exercice 2:

1) It works because main has access to the private variables because it's a method of the class Point.

2) When creating a new class and using the same code as before, we have a problem with the variables x and y because they were previously defined as private on the class Point.

To delete the error message, we can either redefine x and y in the new class or create getters and setters in the class Point.

3) We set all fields visibility to private to secure our variables and so they can be independent of the variables of other classes. We use public/protected visibilities only when creating constants.

4) An accessor is a method that returns the value of a private field. Yes, we have to use getters here so we can use in TestPoint the variables x and y of the class Point.

5) When creating a constructor with two arguments (called px and py), we have an error on the following line of code: Point p = new Point();

Because 2 parameters are missing in the definition of the object p (because of the new constructor).

6) When modifying the parameters to x and y, we need to instantiate the variables.

7) To know how many points we created, we need to create a static variable count and increment it on the constructor and print it on the main method (check code).

8) The compiler relies on the signature of the method to call the right one. It all depends on the parameters used.

9) (check code)

Exercice 3:

1) This code prints true and then false.

True is printed because when writing « Point p2 = p1; », we gave to p2 the adress of p1. So when comparing the adress of p1 and the adress of p2, we find that they are the same and so the boolean expression p1==p2 is considered true.

False is printed because even though p1 and p3 have the same parameters, they are still 2 different objects with 2 different adresses. So the boolean expression p1==p3 is false.

2) (check code)

3)

Exercice 4:

1) (check code)

2) (check code)

3)

5) null pointer Exception occurs/is raised because we are trying to compare a point object to a null value

Exercice 5:

5) the problem is that p's value is modified

To avoid it we need to use getters on x and y and modify locally the value of these variables

6) erreur car on essaye de translater le centre or on l'a pas stocker dans une variable donc quand on va vouloir translater il va pas savoir quoi translater

Exercice 6:

1) we can use inheritance because a ring is made of 2 circles

2)