25/09/2024

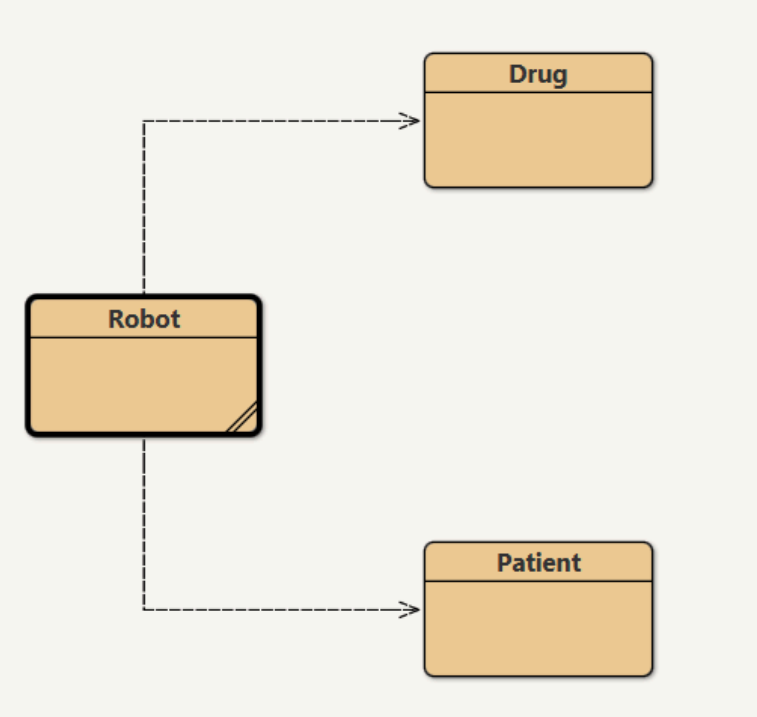
# Lab 4.4: Helpy, a first prototype

***Group n°5 :***

DEVOUCOUX Grégoire (Scribe) - LASFAR Nisrine - HACQUART Virgile - ROUIBAH Bilal (Group Leader) - ILEKTI Linda (Time Keeper)

---

**Class diagram :**




**Identification of the relationships :**

As we can see from the relationships between the three classes, the Robot class utilizes functionality provided by the Drug and Patient classes. There is a 'use' relationship between the Robot class and these classes because it directly calls and employs methods defined in them.

**Record of the functionalities :**

- **Pregnant contraindication :**

Here we create a Drug which is contraindicated. We will create a patient who is pregnant, and so a woman. First we create a drug with the contraindication “pregnant” and with 3 doses per day, as well as 0 drugs in stock.  Then, we create a patient, with

the constructor we ask for name, sexe, if the patient is asthmatic and if he/she is pregnant.
Finally, we create the robot linking the patient to the drug. We use the method boolean checkContraindication() and we want to have a “true” printed because the pregnant contraindication is true for the drug and the patient is a pregnant woman.

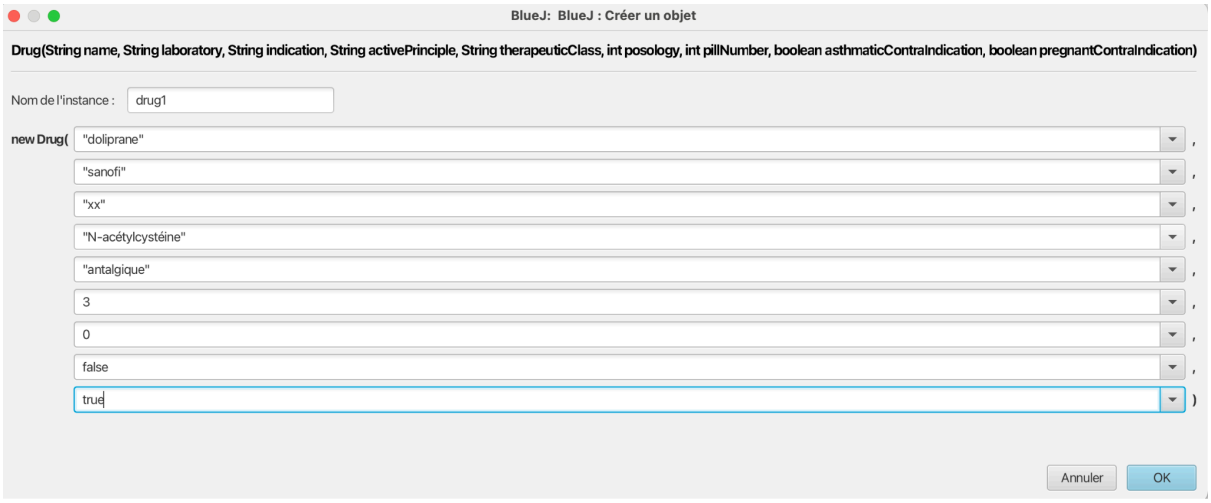



*Fig.1 : Creation of “Drug contraindication for pregnant” object.*

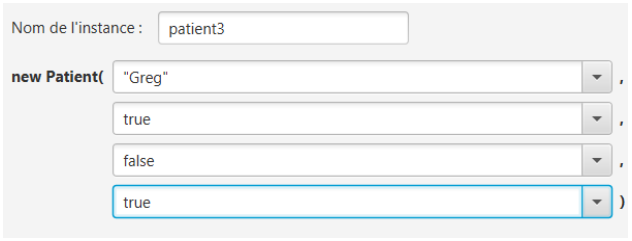



*Fig.2 : Creation of “pregnant patient” object.*

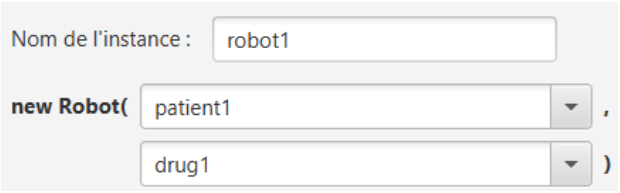



*Fig.3 : Creation of “Robot” object.*

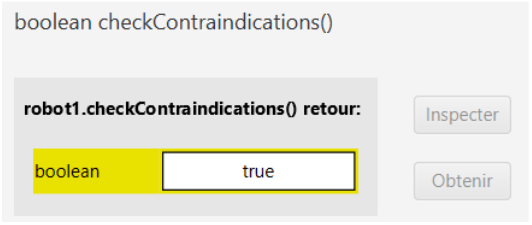



*Fig.4 : Print result of the result of the check of the presence of contraindication on the drug and patient. True means that the patient must not take the drug.*

- **Asthmatic contraindication :**

The process for checking if a drug is contraindicated for someone with asthma is the same, except that this time, when creating the "Drug" object in the "Drug" class, we set the "asthmatic" attribute to "true" and the “pregnant” attribute to “false”.

For the Patient we do the same for the contraindication we set it at false for pregnancy and true for asthma. We create a robot once again and we use the method boolean “checkContraindication()” and once again we want to have an answer true. Which is the case here.

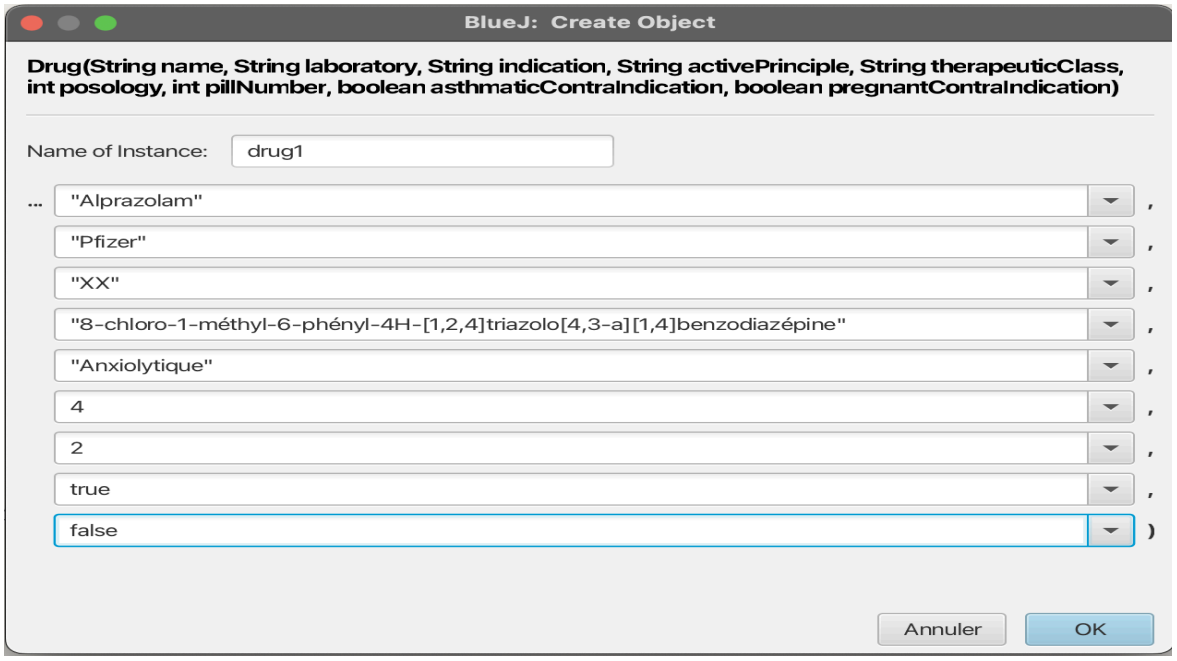



*Fig.5 : Creation of “Drug contraindication for asthmatic” object.*

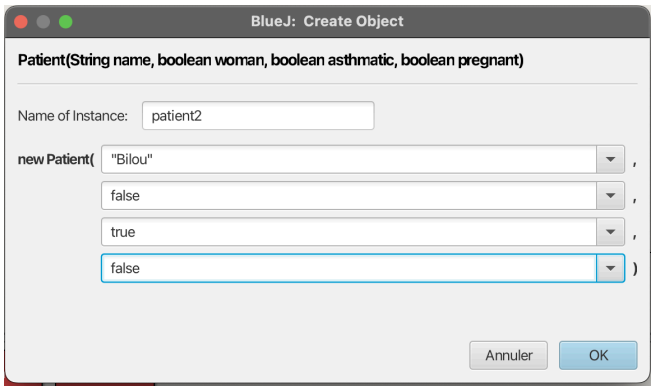



*Fig.6 : Creation of “Asthmatic patient” object.*

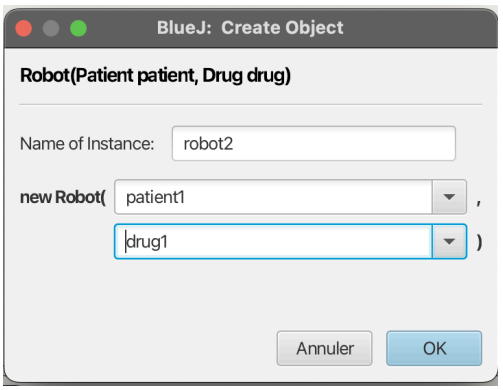



*Fig.7 : Creation of “Robot” object.*

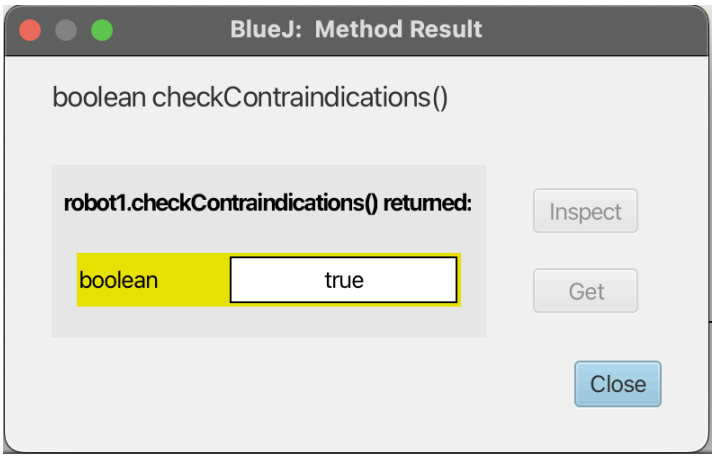



*Fig.8 : Results of asthmatic contraindication*

- **With no contraindication :**

We create a Drug with no contraindication and a patient with no contraindication too. And we use the same method once again. We want a result false because if there is not contraindication we coded it that way and that is what we obtain.

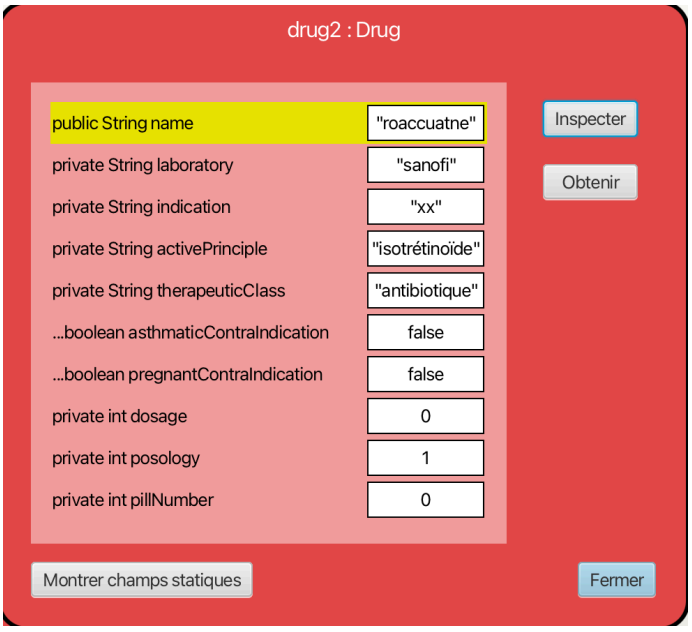



Fig: 10 : Creation of a drug without contraindication

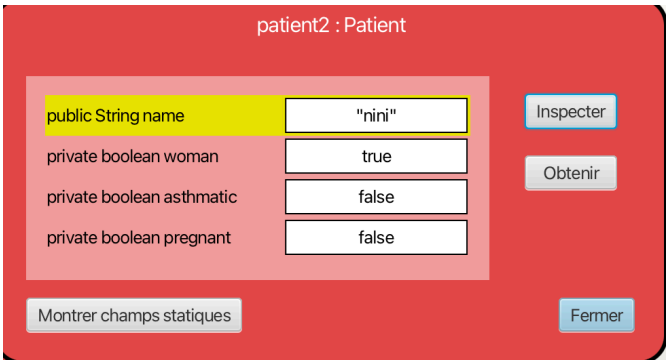



Fig 11: Creation of a patient without contraindication

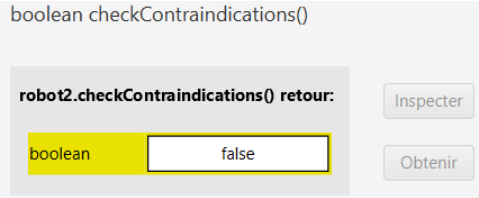



*Fig 12 : Result of the method checking the presence of a contraindication*

- **Control of the stocks :**

We setted the number of drugs at “0” for a patient, but the dosage per day of this drug is 3. As we did on top of here. We use the method void verifstockDrugs() in order to check the stock of the patient and if it is “0” we have a message “Acheter de nouveaux comprimés de xxx”.

```
void buyNewDrugs(int quantity)
boolean checkContraindications()
void deliverDrugs()
void deliverPrescription()
void verifstockDrugs()
```

*Fig 13 : Use of the method void verifstockDrugs() in the object “Robot”*

```
Acheter de nouveaux comprimés de xxx.
```

*Fig. 14 : Result of the method showed in Fig.13.*

- **Buy new drugs**

Additionally, we set the number of pills available in the patient's stock for the same medication to '1'. We know that we need more drugs because of the test before. We use the method void buyNewDrugs( int quantity). We buy 5 drugs and the stock on the drug object stepped up to 6.

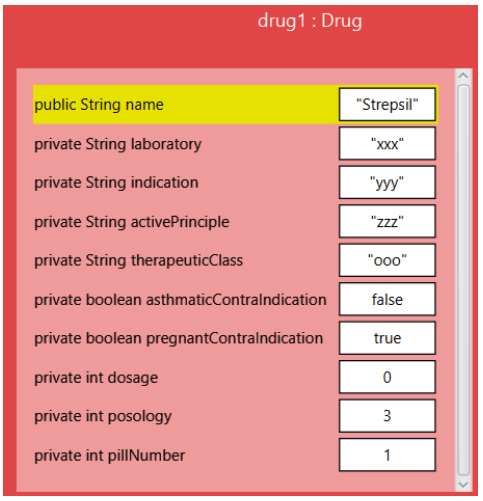



*Fig.15 : Stock of pills before buying it.*

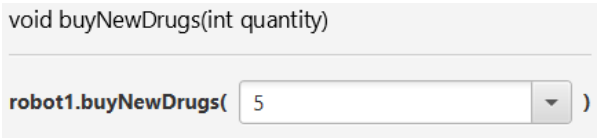



*Fig 16 : Buy 5 pills.*

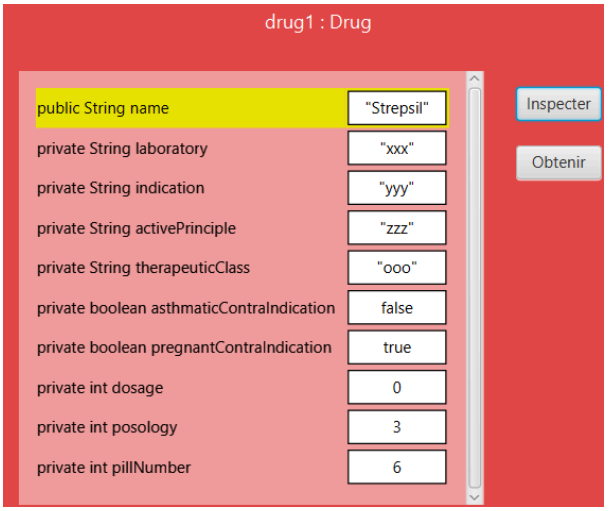



*Fig.17 : Stock with 5 more pills after buying 5.*