

Premiers pas en programmation orientée objet

Un guide efficace proposé par

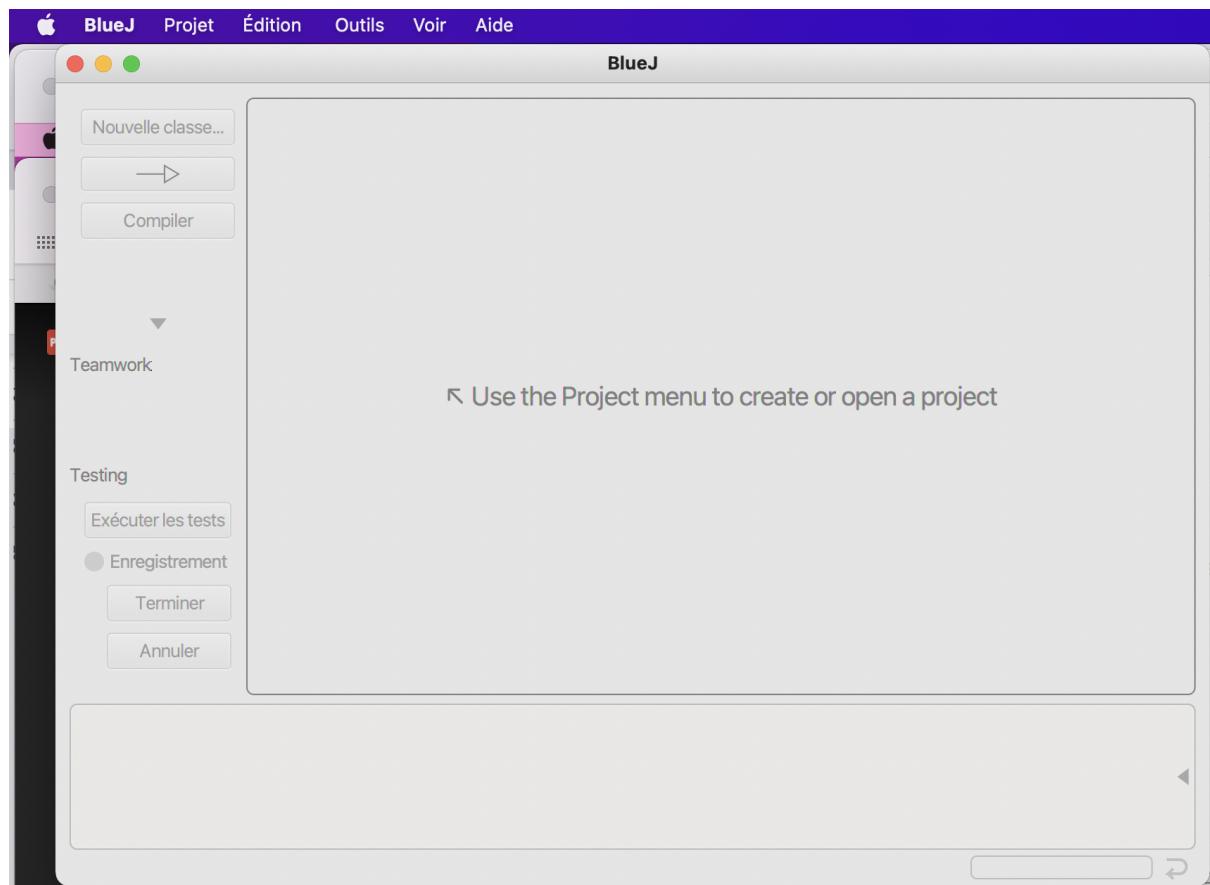
**Jérémy SELLEM
Mohamed BOUNAIME**

Prérequis :

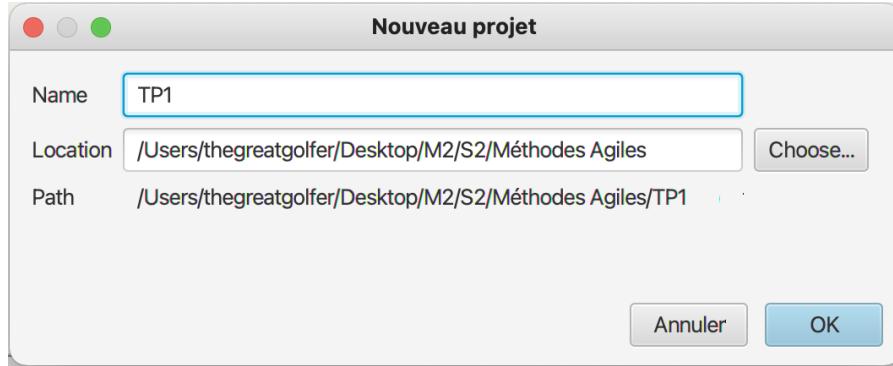
- Télécharger et installer *BlueJ* disponible via [ce lien](#)
- Télécharger et installer *Eclipse* disponible via [ce lien](#)

Tutorial :

- 1) Lancer BlueJ, une fenêtre apparaît

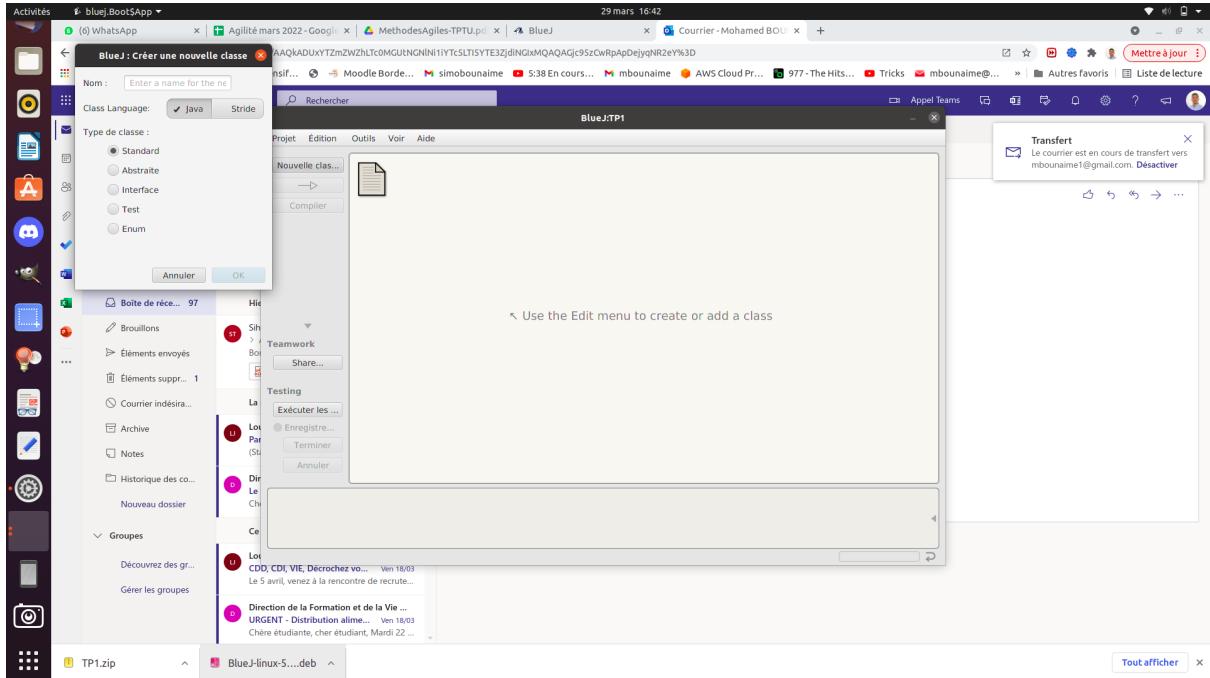


- 2) En haut à gauche de la fenêtre, sélectionner **Projet > Nouveau Projet...**
- 3) Une boîte de dialogue apparaît, renseigner le nom de votre projet, et choisir un dossier où le ranger. Dans l'exemple, nous appelons le projet **TP1**.
Cliquer sur OK

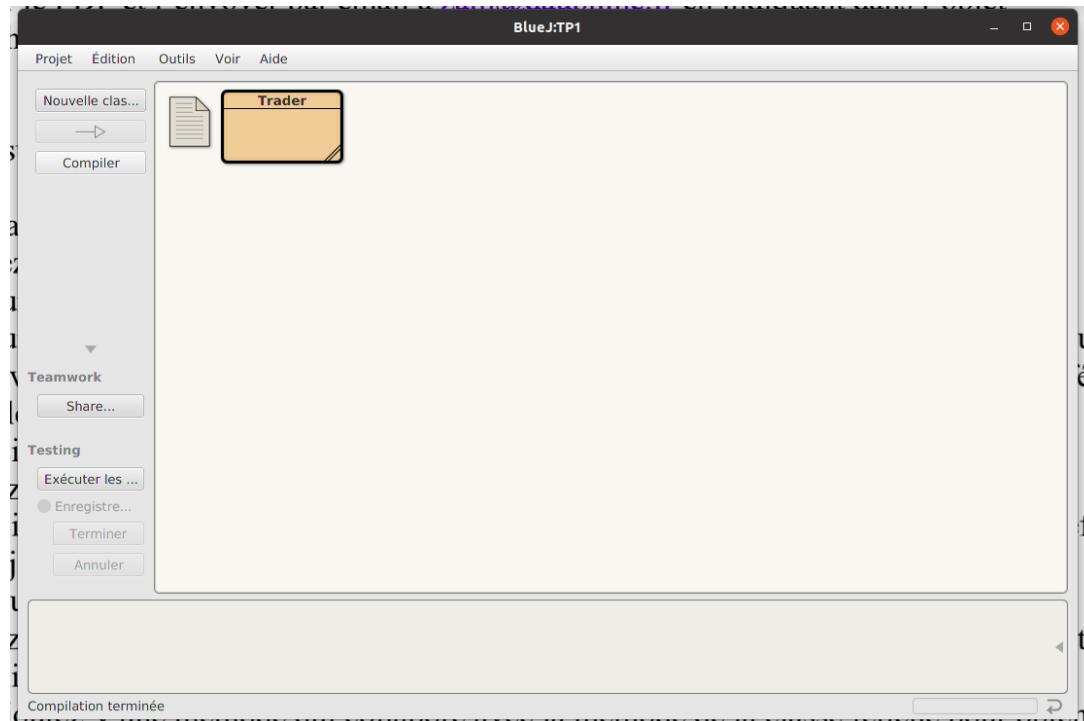


👉 Bravo, votre premier projet est lancé. Prêt à découvrir la suite ?

- 4) Nous allons ajouter une classe à votre projet. Une classe définit un type d'objet, par exemple un Trader.
Cliquer sur Nouvelle Classe...
Renseigner le nom de votre classe, dans l'exemple nous l'appellerons **Trader**
Sélectionner **Java** puis **Standard**. Cliquer sur **OK**



5) Pause, vérifier que vous voyez bien apparaître sur l'aire centrale votre classe

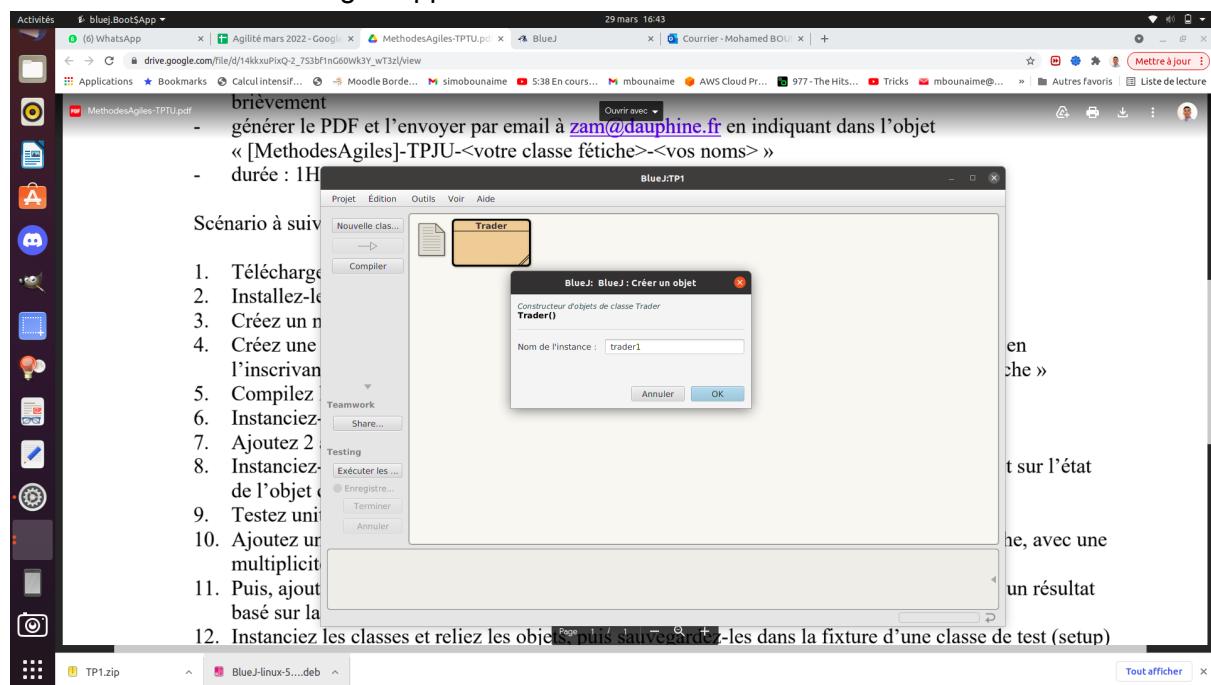


6) Il est temps de créer votre premier Trader !

Clique droit sur le rectangle **Trader**

Sélectionner **new Trader()**

Une boîte de dialogue apparaît



7) Donner un nom au trader, nous vous proposons de manière très originale : **trader1**
Cliquer sur **OK**

- 8) Parfait, votre trader apparaît sur la ligne en bas de votre fenêtre, cette ligne contient les instances de vos classes, c'est-à-dire les objets créés. Vous avez actuellement un objet **Trader** créé.



- 9) Nous allons maintenant ajouter des paramètres à notre classe Trader, pour cela :

Double cliquez sur **Trader**

Le code s'ouvre, écrivez les code ci-dessous

Cliquer sur **Compiler**

```

brevement
- générer le PDF et
  « [MéthodesAgile
- durée : 1H30

Scénario à suivre (copier/coller)
1. Téléchargez BlueJ
2. Installez-le sur votre ordinateur
3. Créez un nouveau projet
4. Créez une première classe nommée "Trader" l'inscrivant sur la ligne d'objets
5. Compilez la classe
6. Instancez-là
7. Ajoutez 2 attributs
8. Instancez-là de nouveau et affichez la valeur de l'objet créé.
9. Testez unitairement l'ajout d'un attribut
10. Ajoutez une seconde méthode add() avec une multiplicité 0..1 à 0..1.
11. Puis, ajoutez-y une méthode add() avec une multiplicité 0..1 à 0..1 basé sur la contribution de deux attributs.
12. Instancez les classes et relancez la méthode add() dans la méthode main() de la classe de test (Setup).

Trader - TP1
Classe Édition Outils Options
Compiler Défaire Couper Copier Coller Chercher Fermer Implémentation
private int y;

/*
 * Constructeur d'objets de classe Trader
 */
public Trader()
{
    // initialisation des variables d'instance
    x = 0;
    y = 0;
}

/*
 * Un exemple de méthode - remplacez ce commentaire par le vôtre
 *
 * @param y    le paramètre de la méthode
 * @return     la somme de x et de y
 */
public void add(int b)
{
    // Insérez votre code ici
    x = x + b;
}
enregistré

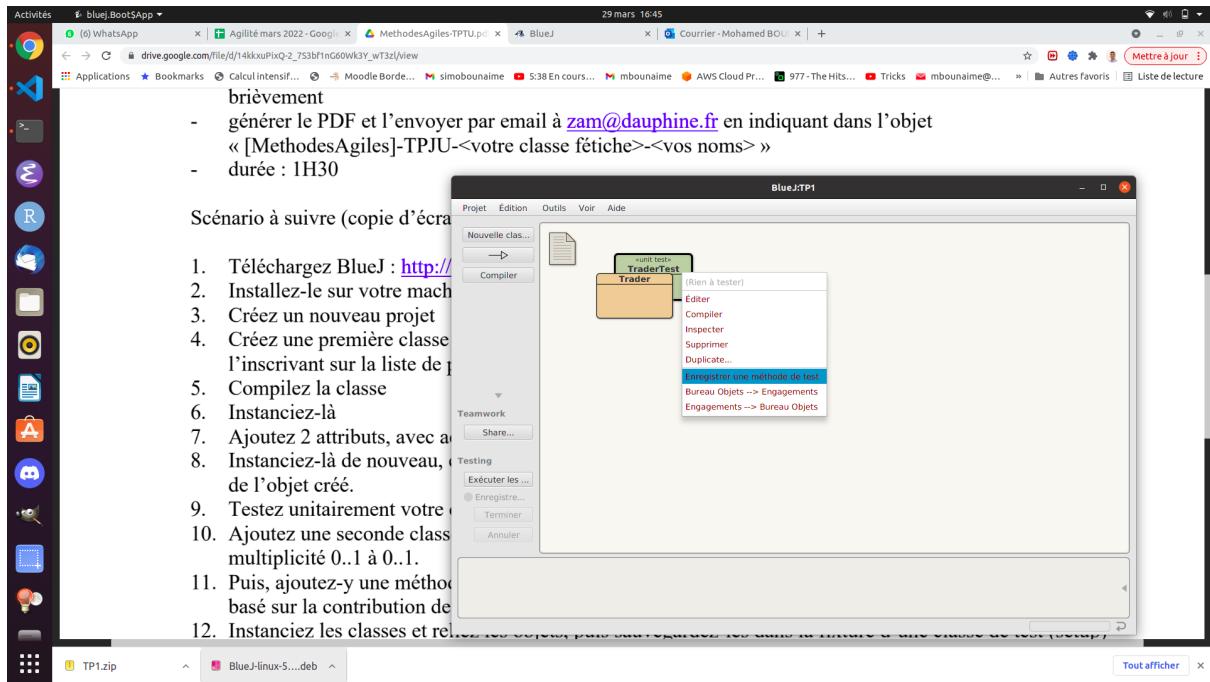
```

10) Nous avons créé une fonction, nous allons devoir la tester

Clique droit sur **Trader**

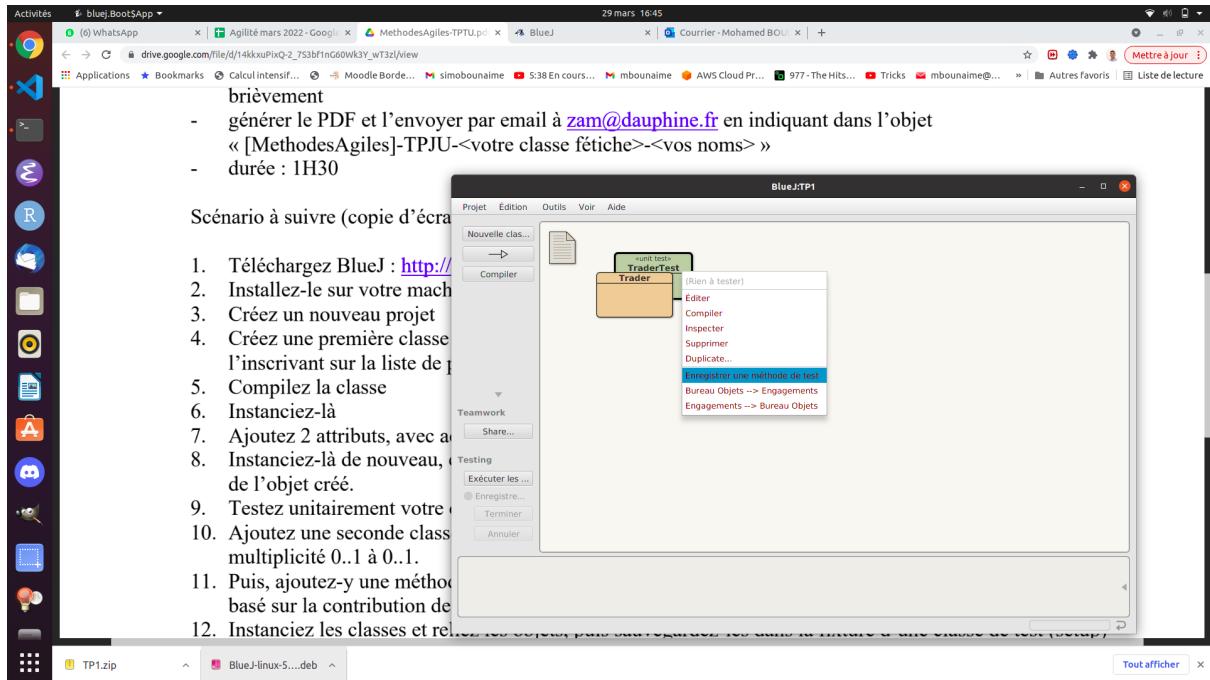
Cliquer sur **Créer classe Test**

Un rectangle vert apparaît derrière notre classe **Trader**

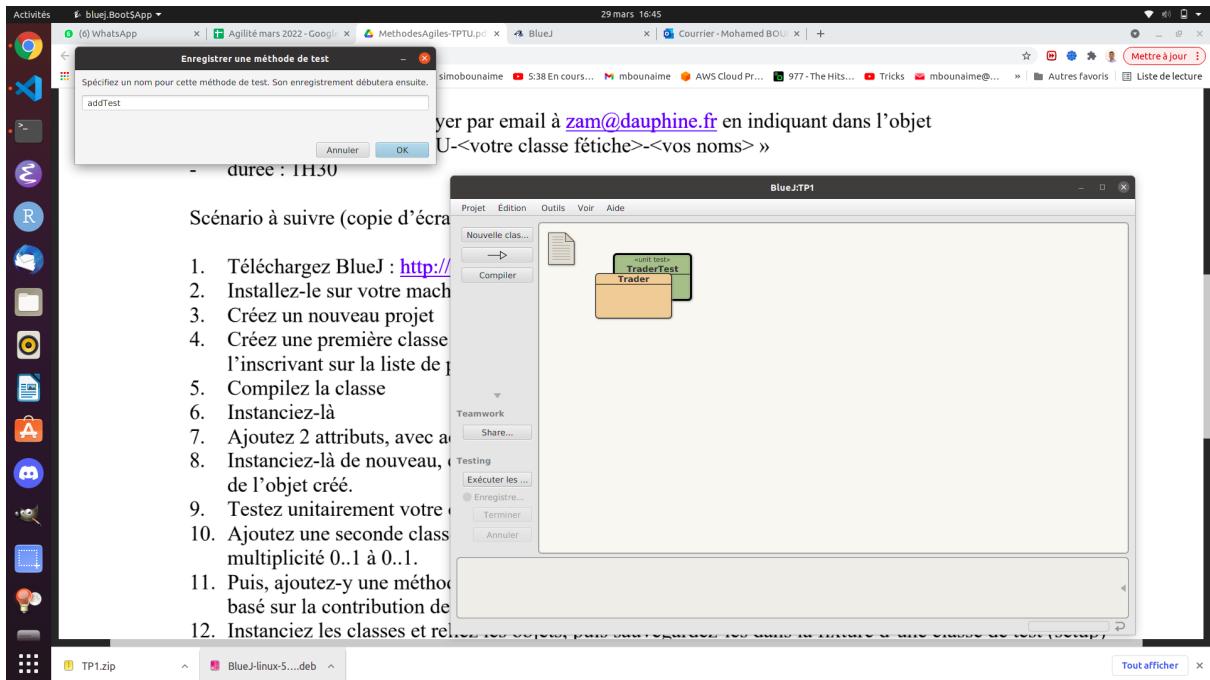


11) Clique droit sur le rectangle vert

Cliquer sur **Enregistrer une méthode de test**

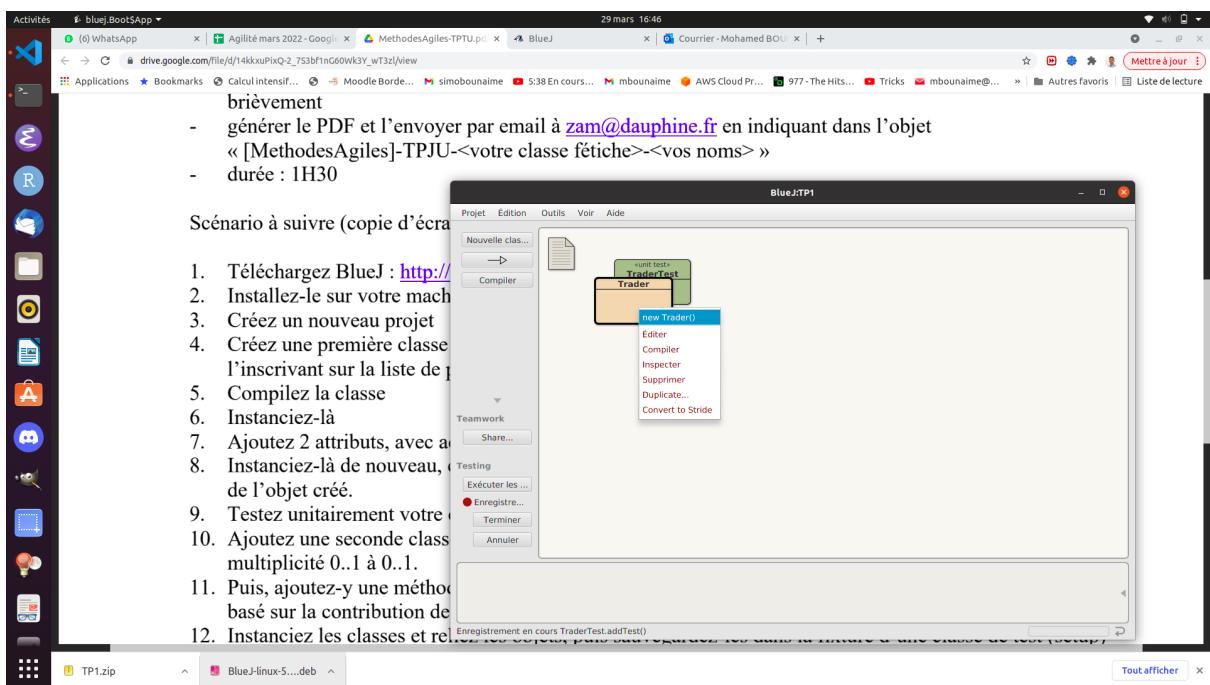


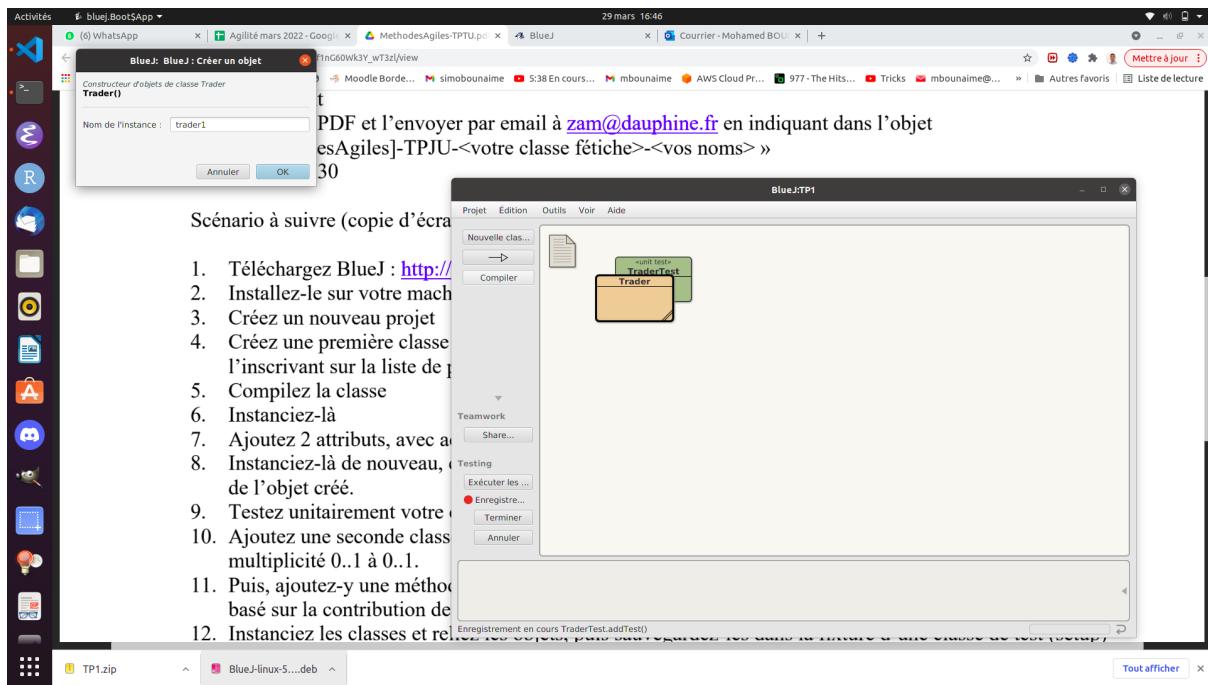
12) Appeler cette méthode **addTest** puisqu'on va tester la méthode **add**



13) L'enregistrement démarre (icône rouge en bas à gauche)

Créer un nouvel objet **Trader**



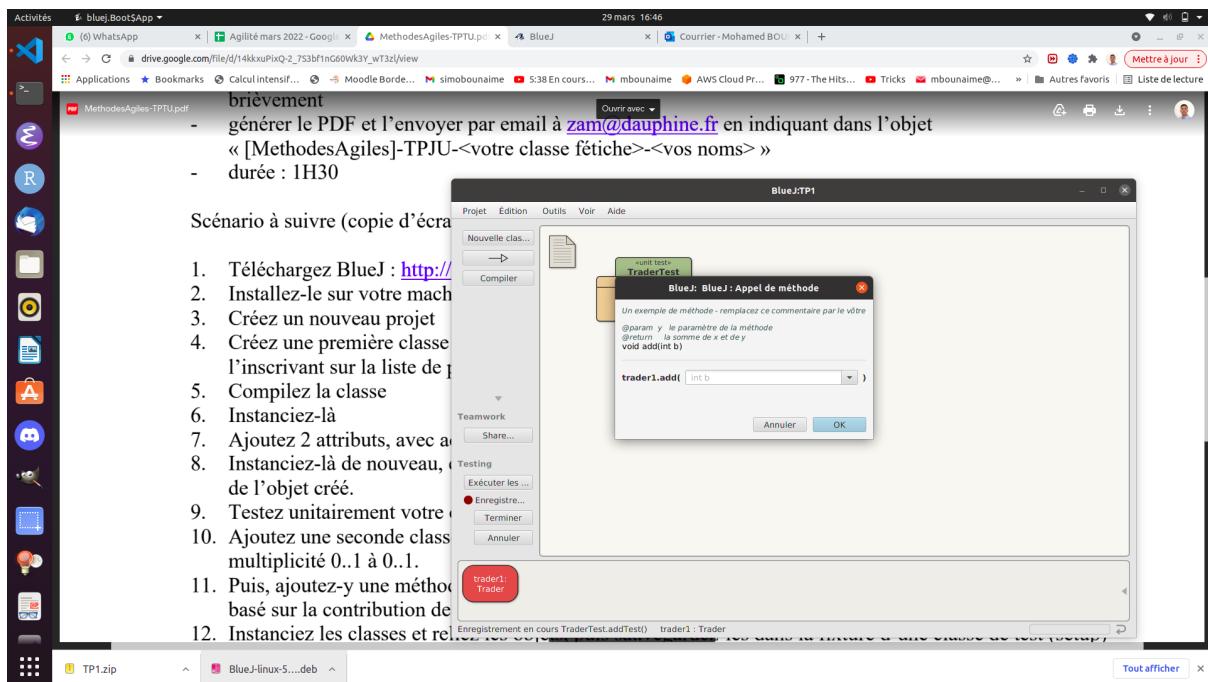


14) Clique droit sur **trader2** (bouton rouge)

Cliquer sur **void add(int b)**

Renseigner une valeur numérique, ex : 5

Cliquer sur **OK**

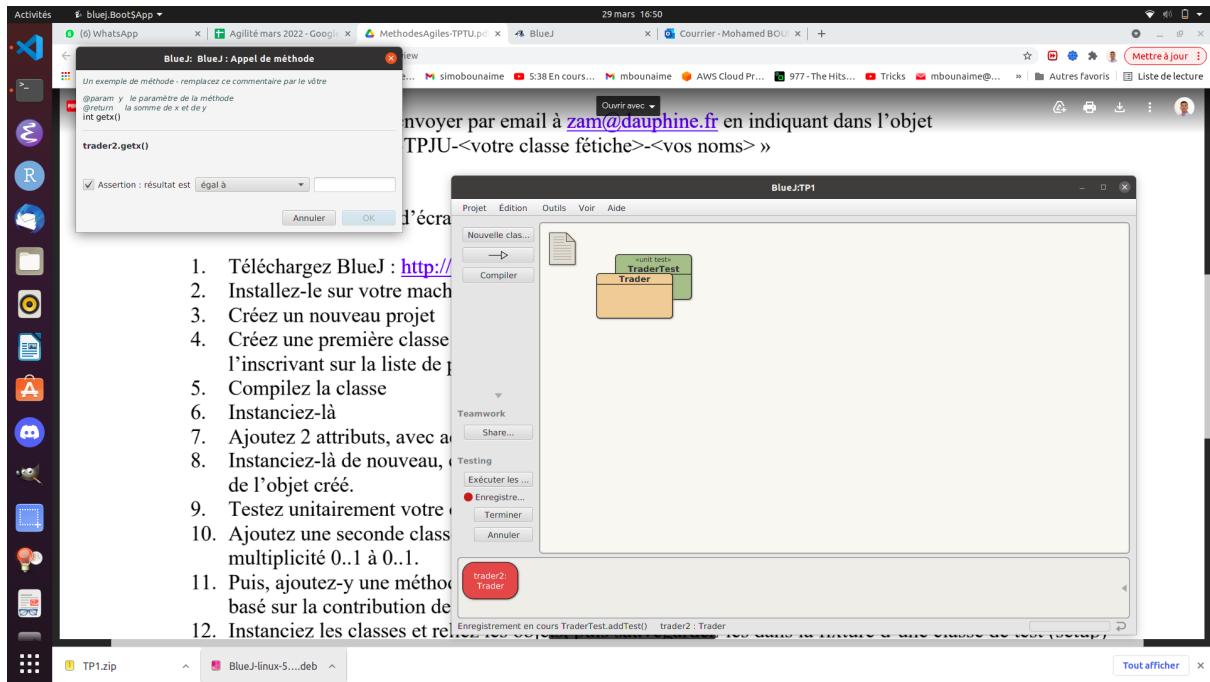


15) Clique droit sur trader2

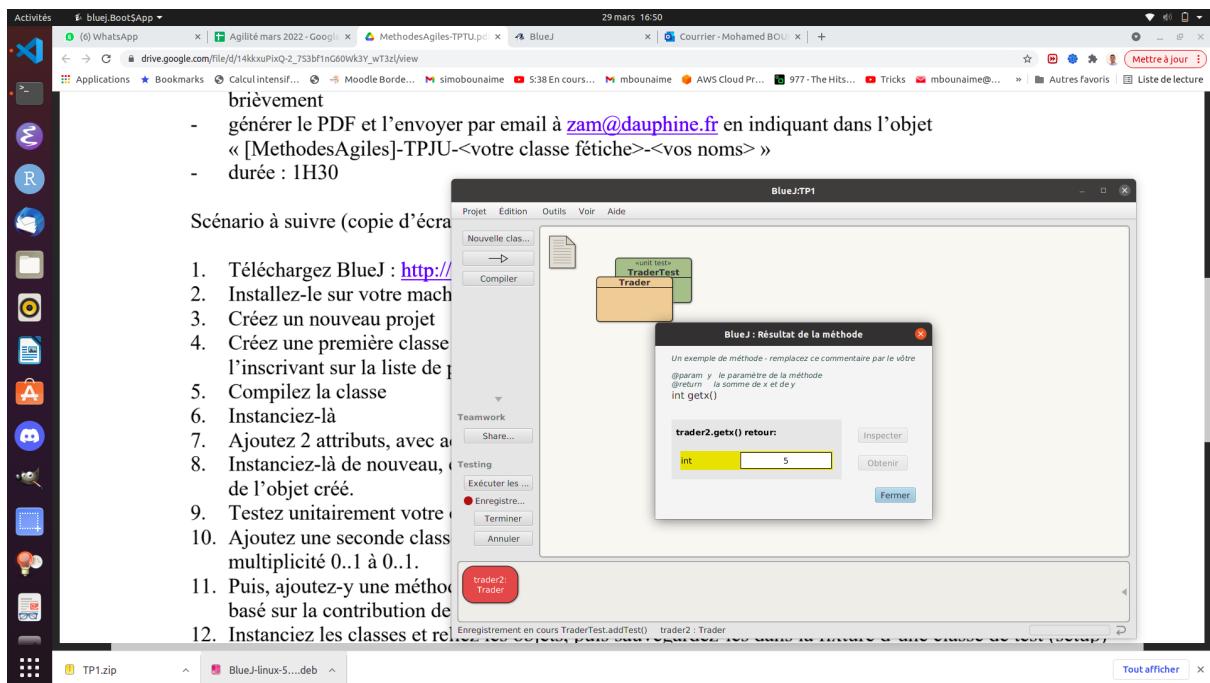
Cliquer sur int getx()

Renseigner la même valeur dans la fenêtre suivante

Cliquer sur OK

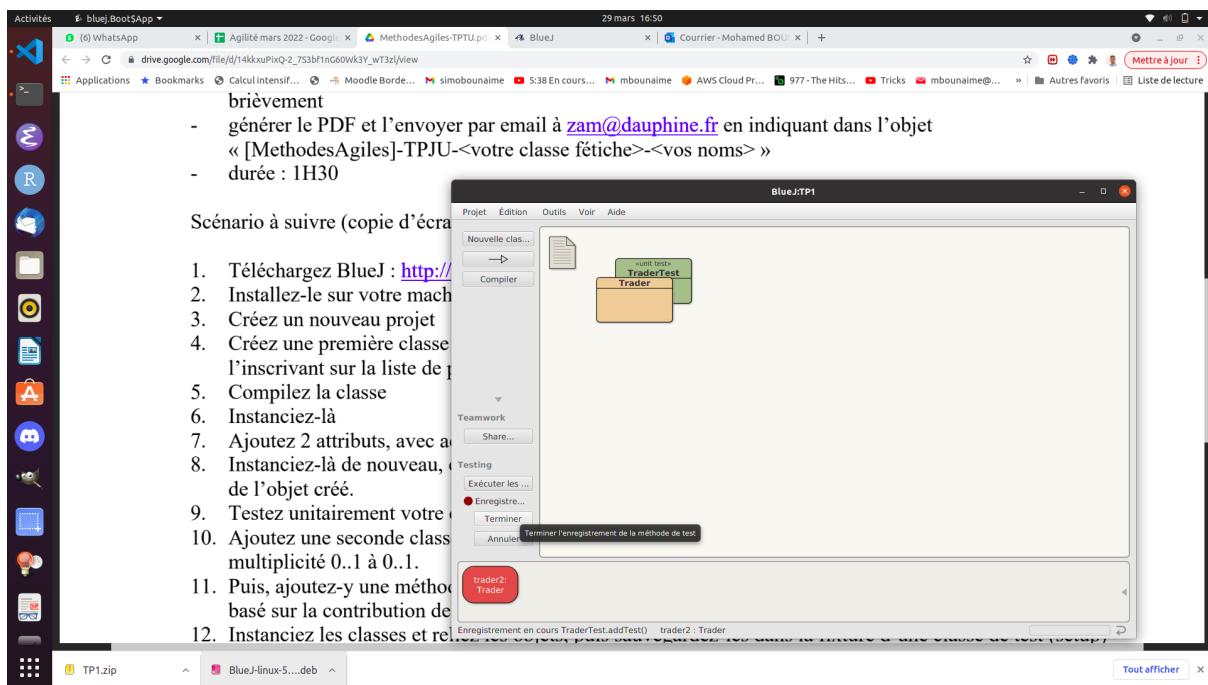


16) La valeur devrait s'afficher



17) Cliquer sur Fermer

Cliquer sur Terminer



18) Notre méthode de test est écrite (comme par magie !)

Vous pouvez constater que le code est rédigé en double cliquant sur le rectangle vert

19) Le but va être de tester réellement cette méthode, pour cela :

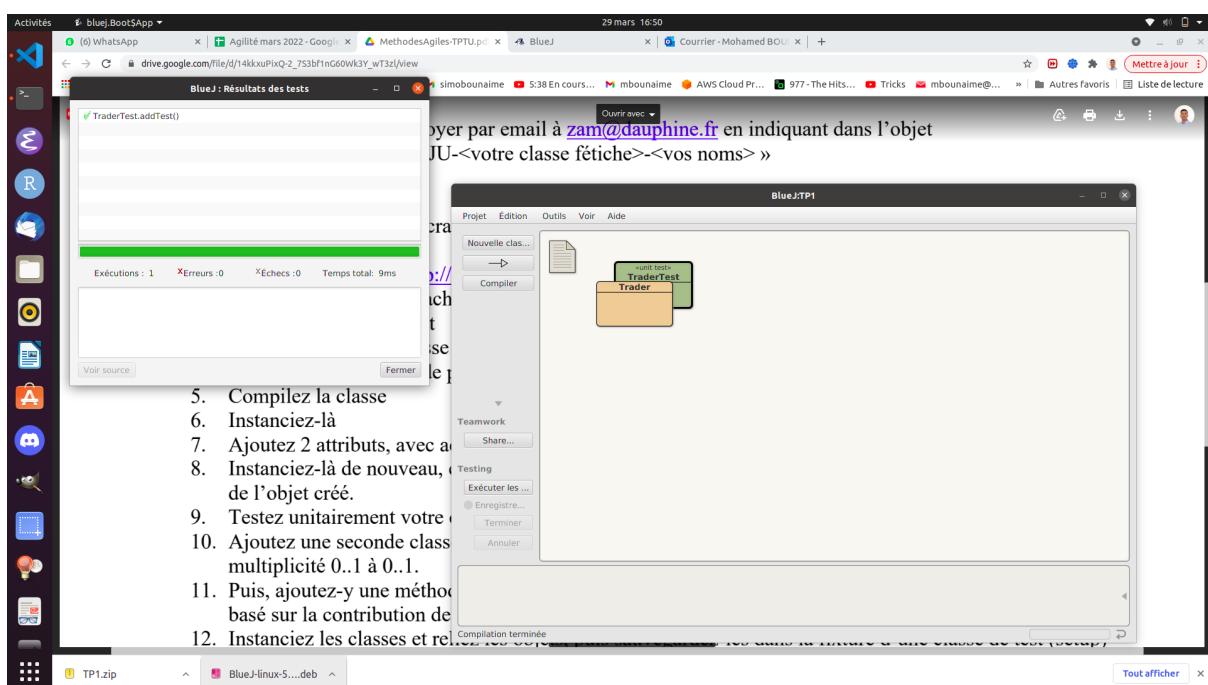
Clique droit sur le rectangle vert

Cliquer sur void **addTest()**

Une fenêtre devrait apparaître avec le résultat du test

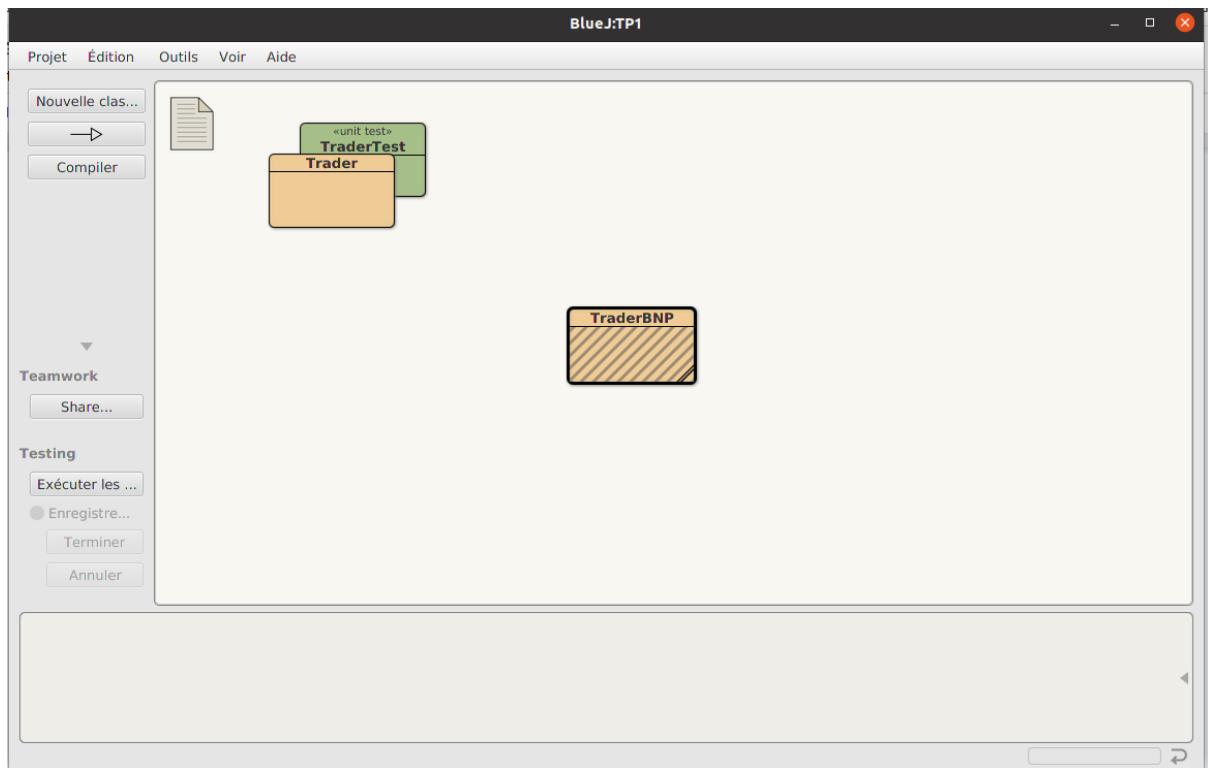
Si la barre est verte : Bravo ! Mission accomplie

Si la barre est rouge: recommencer le processus pour détecter les erreurs éventuelles



20) Maintenant que vous savez créer une classe

Créer la classe **TraderBNP**



21) Ouvrir le code de cette classe et insérer les lignes suivantes

```
TraderBNP - TP1
Classe Édition Outils Options
TraderBNP X
Compiler Défaire Couper Copier Coller Chercher Fermer Implémentation
/*
 * Décrivez votre classe TraderBNP ici.
 *
 * @author (votre nom)
 * @version (un numéro de version ou une date)
 */
public class TraderBNP
{
    // variables d'instance - remplacez l'exemple qui suit par le vôtre
    private int x;
    private Trader t;

    /**
     * Constructeur d'objets de classe TraderBNP
     */
    public TraderBNP()
    {
        // initialisation des variables d'instance
        x = 0;
        t = new Trader();
    }

    /**
     * Un exemple de méthode - remplacez ce commentaire par le vôtre
     *
     * @param y le paramètre de la méthode
     */
}
```

The screenshot shows the code editor for the 'TraderBNP' class. The code is as follows:

```
/*
 * Décrivez votre classe TraderBNP ici.
 *
 * @author (votre nom)
 * @version (un numéro de version ou une date)
 */
public class TraderBNP
{
    // variables d'instance - remplacez l'exemple qui suit par le vôtre
    private int x;
    private Trader t;

    /**
     * Constructeur d'objets de classe TraderBNP
     */
    public TraderBNP()
    {
        // initialisation des variables d'instance
        x = 0;
        t = new Trader();
    }

    /**
     * Un exemple de méthode - remplacez ce commentaire par le vôtre
     *
     * @param y le paramètre de la méthode
     */
}
```

TraderBNP - TP1

Classe Édition Outils Options

TraderBNP X Compiler Défaire Couper Copier Coller Chercher Fermer Implémentation

```
{
    // initialisation des variables d'instance
    x = 0;
    t = new Trader();
}

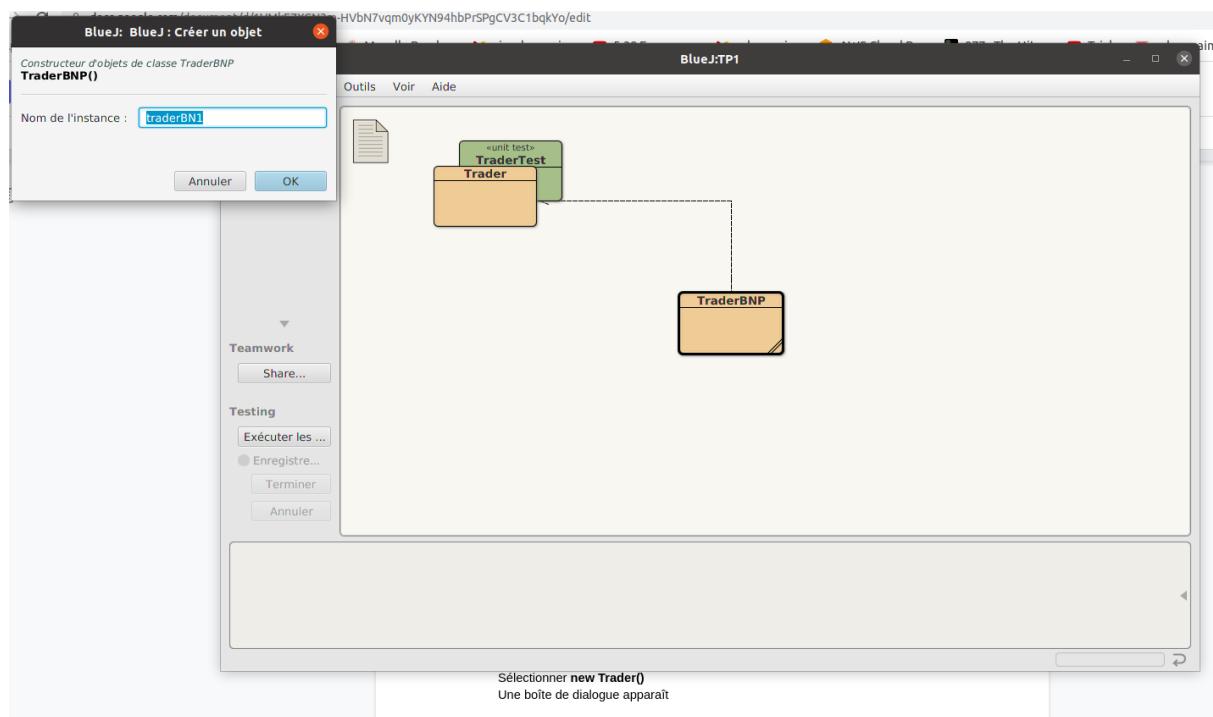
/**
 * Un exemple de méthode - remplacez ce commentaire par le vôtre
 *
 * @param y    le paramètre de la méthode
 * @return     la somme de x et de y
 */
public int sampleMethod(int y)
{
    // Insérez votre code ici
    return x + y;
}

public void get_x_trader()
{
    x = t.getx();
}
}
```

Classe compilée - pas d'erreur de syntaxe enregistré

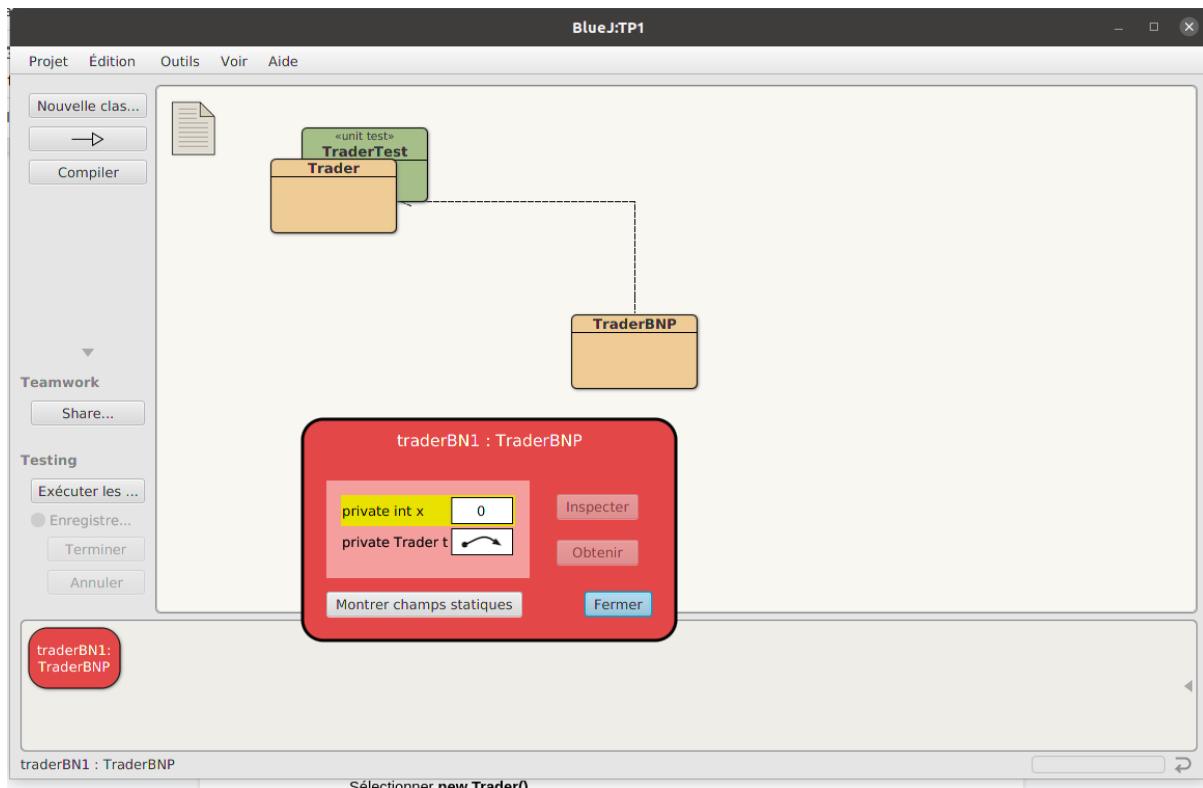
Nous avons lié nos deux classes, en effet, un **TraderBNP** est avant tout un **Trader**!

22) Créer un objet de type **TraderBNP**



23) Instancier la classe **TraderBNP**

Voir la relation entre les deux classes



- 24) **Défi :** vous pouvez à présent réaliser par vous-même le test sur la méthode de la classe **TraderBNP**
Suivez les étapes 13 à 19 sur cette nouvelle classe pour interactivement arriver à un test réussi

À bientôt 🙌