Tutoriel : Création et manipulation de classes en Java:

Introduction

Ce tutoriel vous guide pas à pas dans la découverte de la **programmation orientée objet (POO) en Java** à travers l'environnement de développement **BlueJ**.

Vous allez apprendre à :

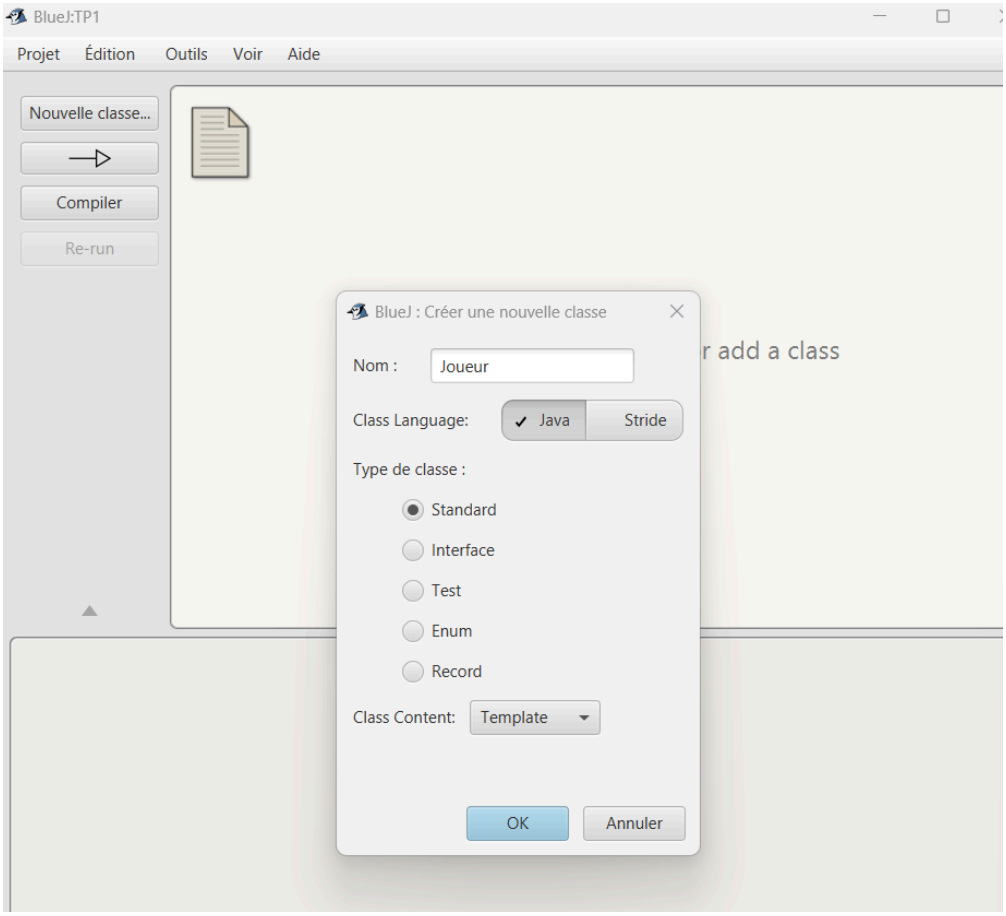
- **Créer et compiler** des classes Java
- **Instancier des objets** et manipuler leurs attributs
- Utiliser les **accesseurs (getters/setters)** pour encapsuler les données
- Implémenter des **méthodes** pour définir les comportements de vos objets
- Mettre en place des **tests unitaires** pour valider votre code
- Établir des **relations entre objets** (associations entre classes)

Le fil conducteur de ce tutoriel s'articule autour d'un **système de jeu simplifié** avec des joueurs, des camps et des actions comme le boost de mana. Cette approche pratique vous permettra de comprendre concrètement les concepts fondamentaux de la POO.

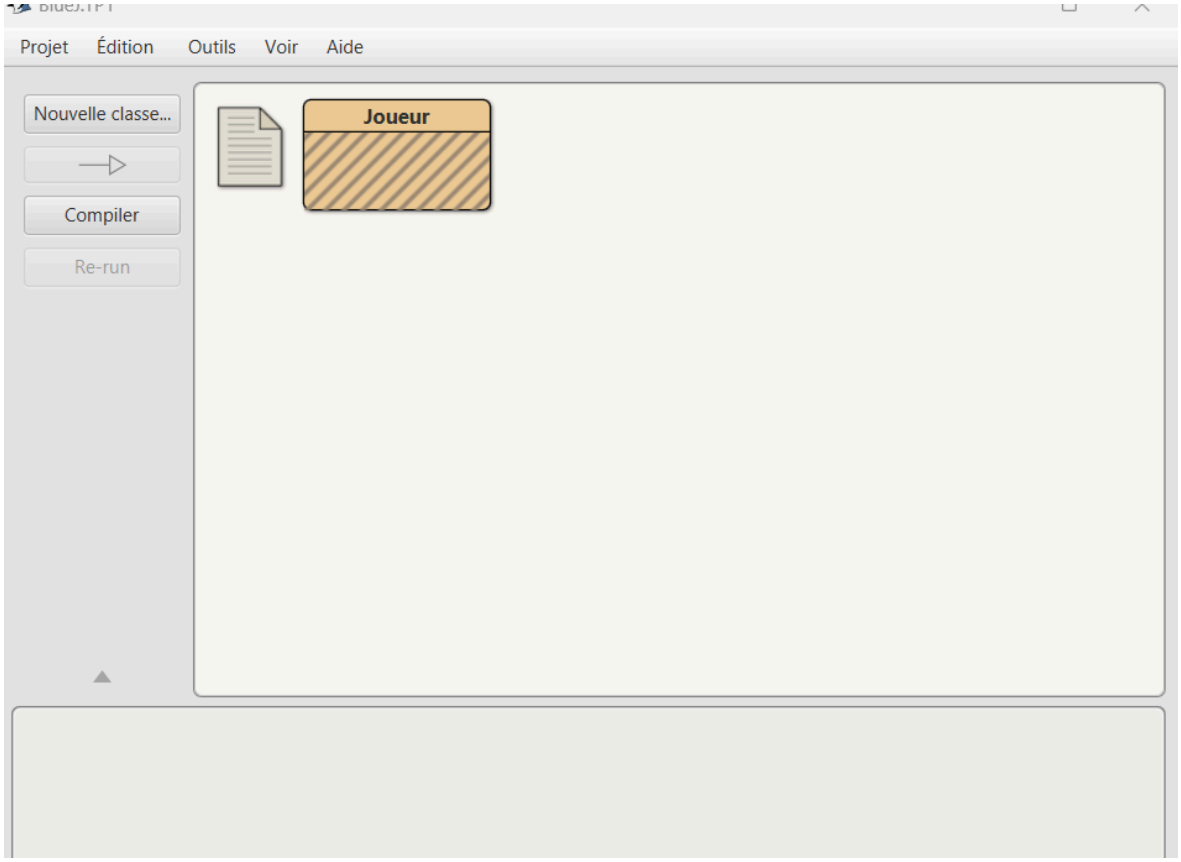
Prérequis : Avoir BlueJ installé et une compréhension basique de la syntaxe Java.

Durée estimée : 30-45 minutes

Étape 1 : Créer une nouvelle classe



Appuyez sur le bouton nouvelle classe

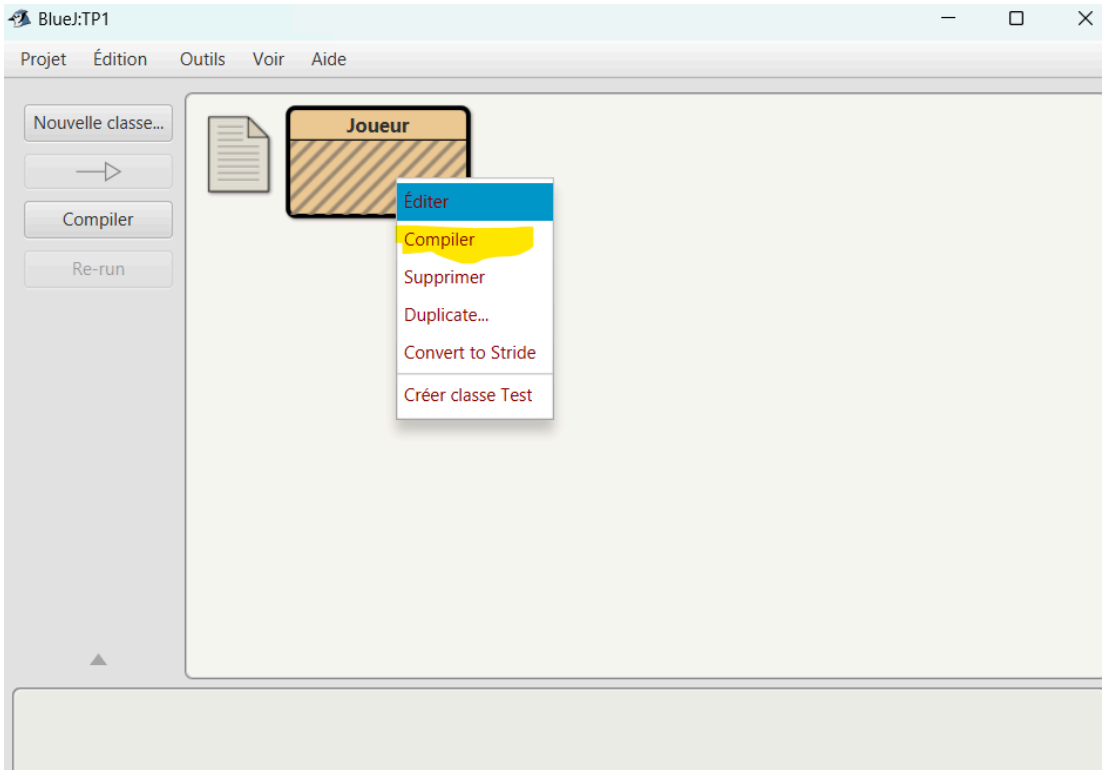


Observation : La classe apparaît grisée, indiquant qu'elle n'est pas encore compilée.

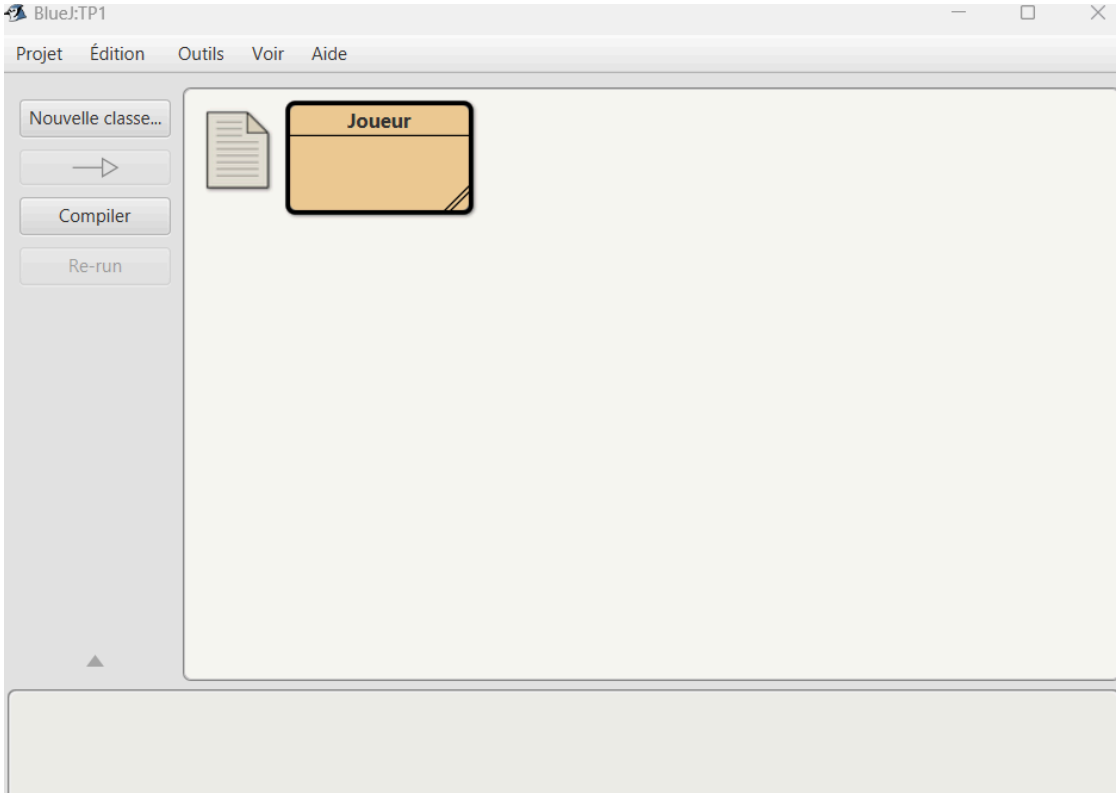
Étape 2 : Compiler la classe:

Effectuez un **clic droit** sur la classe créée

Sélectionnez l'option **Compiler.**

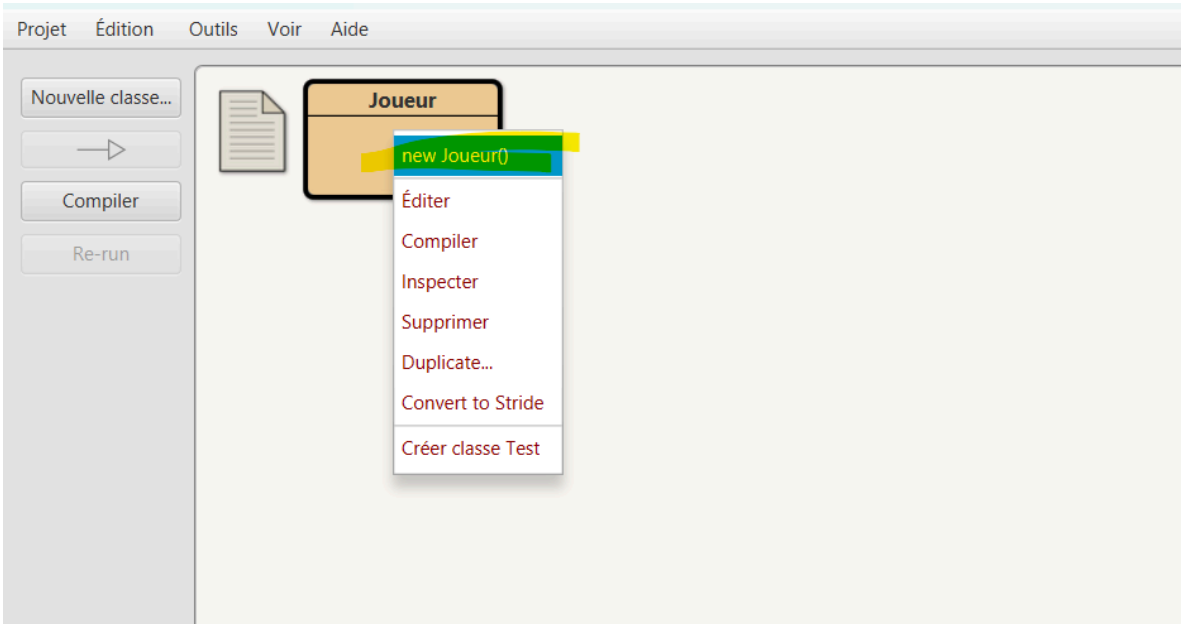


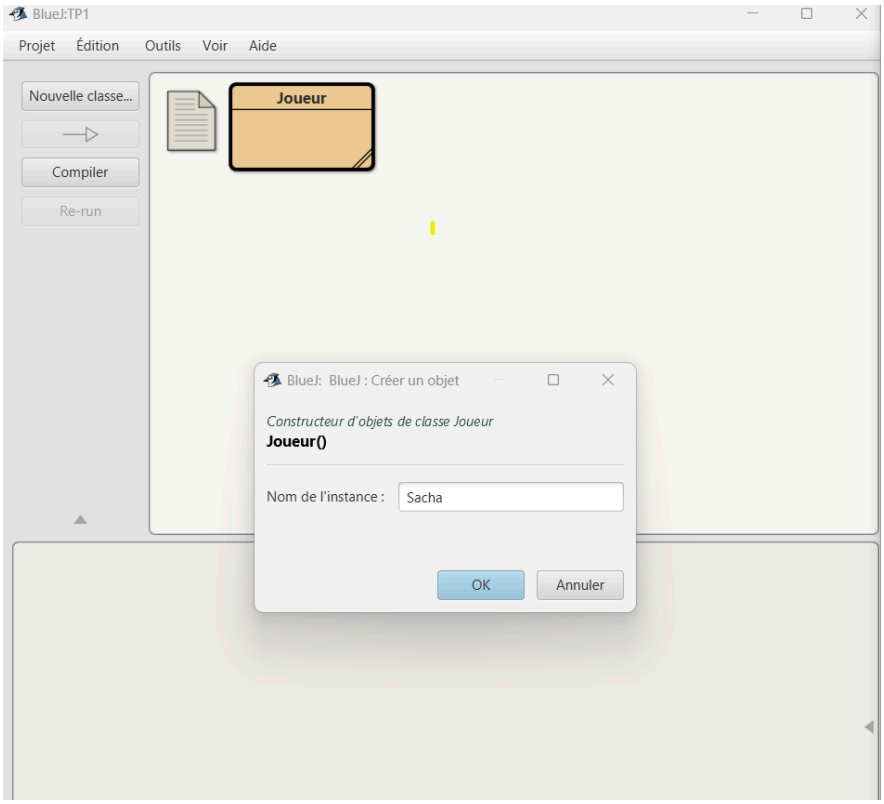
Résultat : La classe n'est plus grisée, elle est maintenant compilée et prête à l'emploi.



Étape 3 : Instancier un objet Joueur

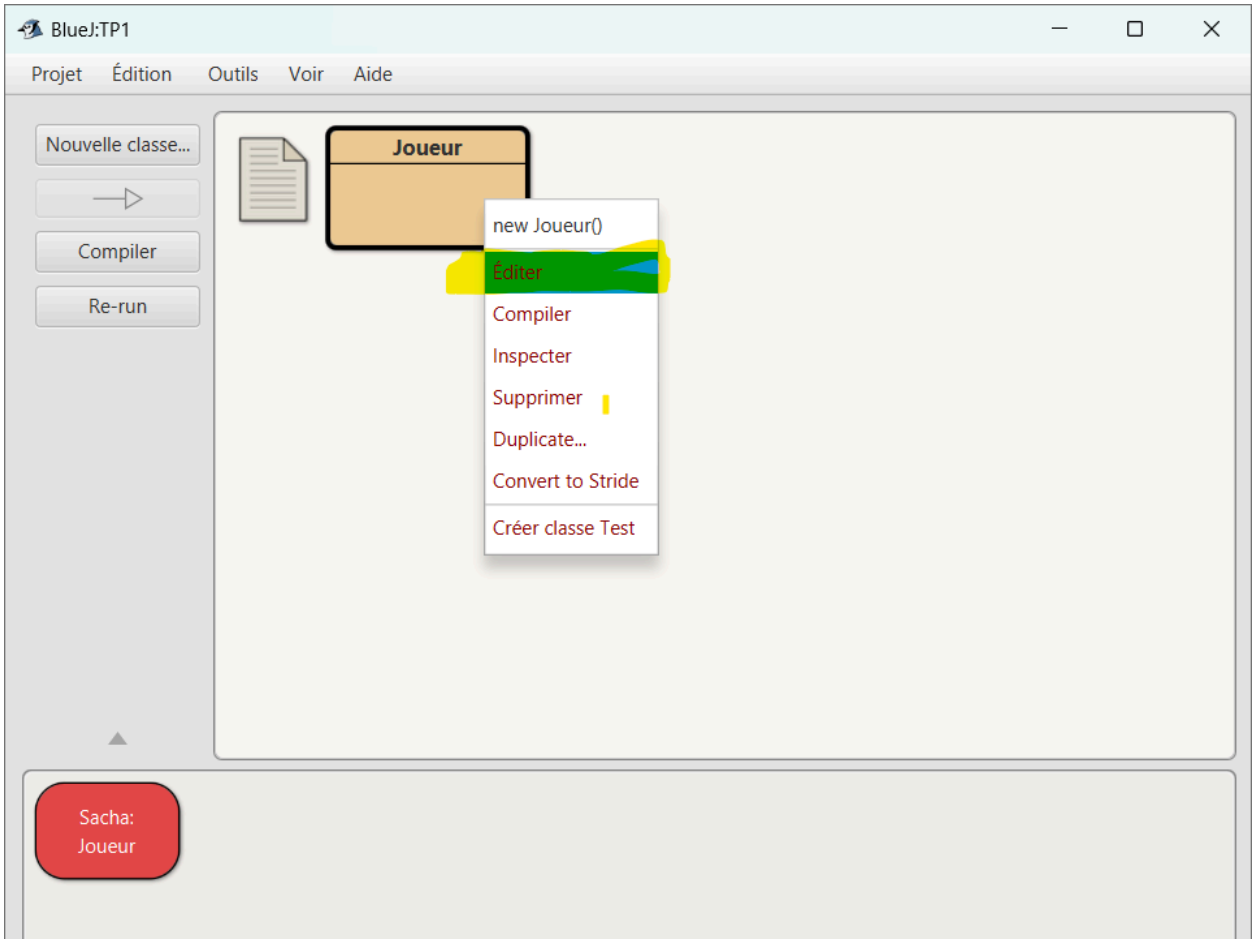
Créez une instance de la classe `Joueur` :





Résultat:

Étape 4 : Enrichir la classe



Ajoutez à votre classe `Joueur` :

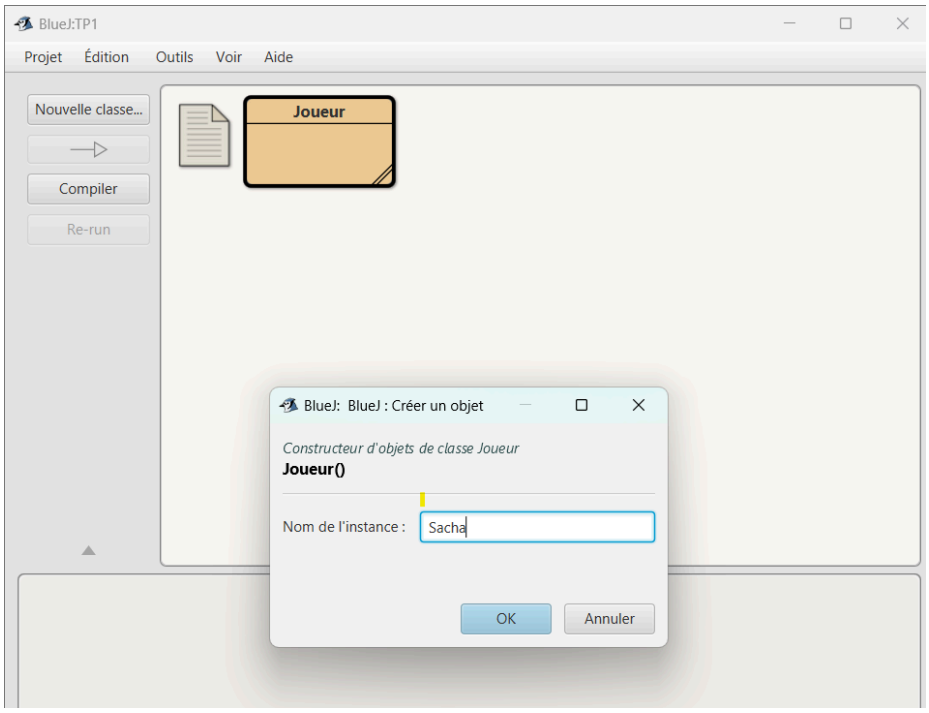
- **Des attributs** (variables d'instance) : un pseudo, de la mana et des points de vie
- **Des accesseurs** (getters et setters) : des méthodes pour accéder aux caractéristiques du joueur

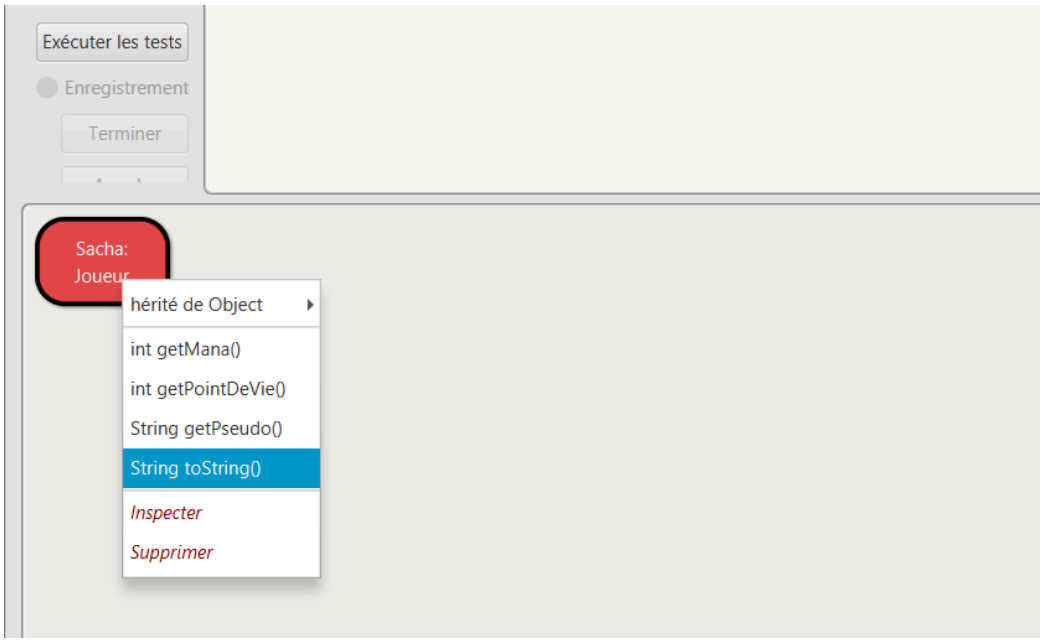
- **Des méthodes** (comportements de la classe) : par exemple (plus tard) attaquer un autre joueur

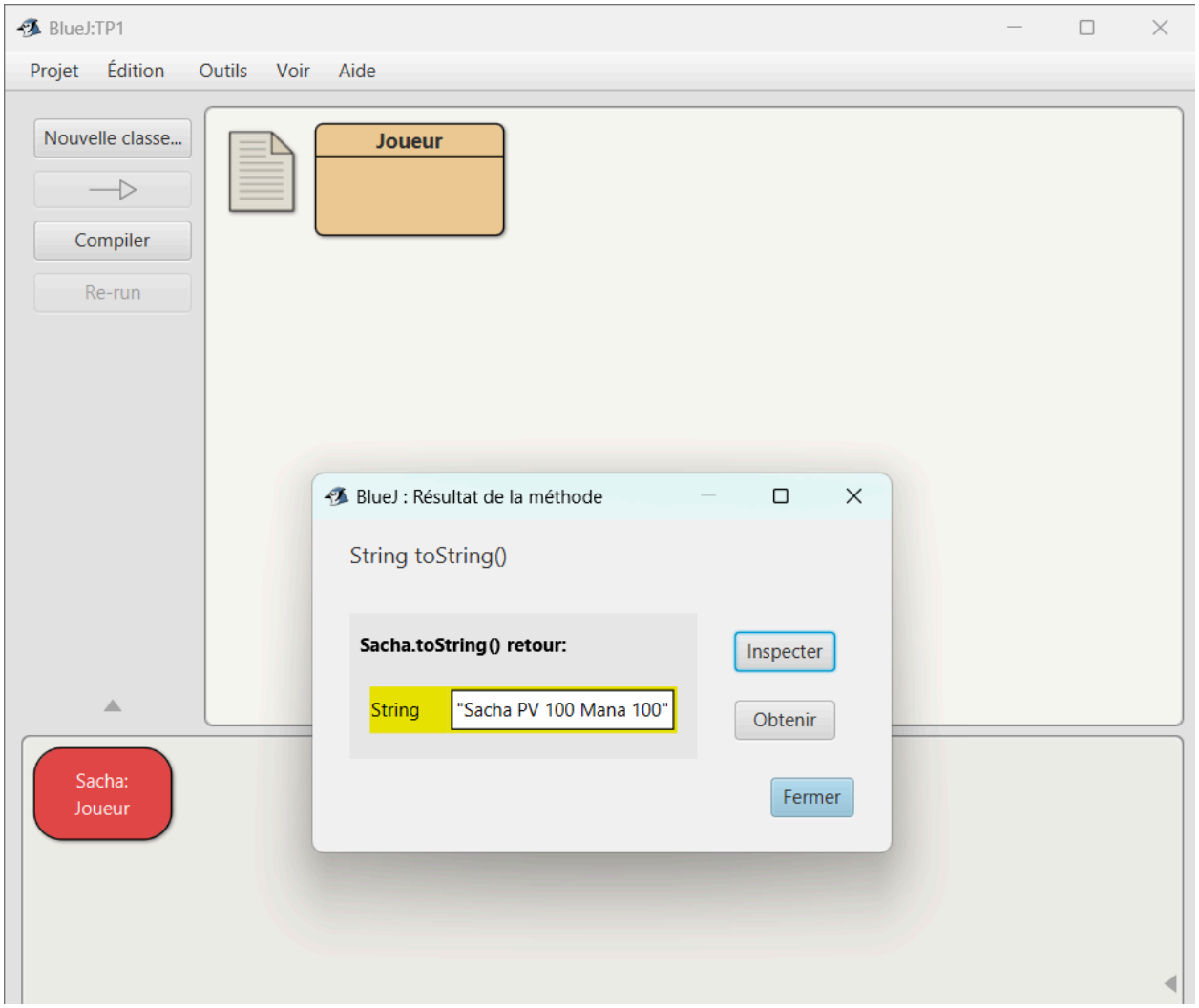
```
public class Joueur
{
    // variables d'instance - remplacez l'exemple qui suit par le vôtre
    private String pseudo;
    private int mana;
    private int pointDeVie;

    /**
     * Constructeur d'objets de classe Joueur
     */
    public Joueur()
    {
        this.pseudo = "Sacha";
        this.mana = 100;
        this.pointDeVie = 100;
    }

    /**
     * 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 getMana()
```

```
    public int getMana()
    {
        // Insérez votre code ici
        return this.mana;
    }

    public int getPointDeVie(){
        return this.pointDeVie;
    }

    public String getPseudo(){
        return this.pseudo;
    }

    public String toString(){
        return this.pseudo + " PV " + this.pointDeVie + " Mana " + this.mana;
    }
```

Étape 5 : Instancier un nouveau joueur

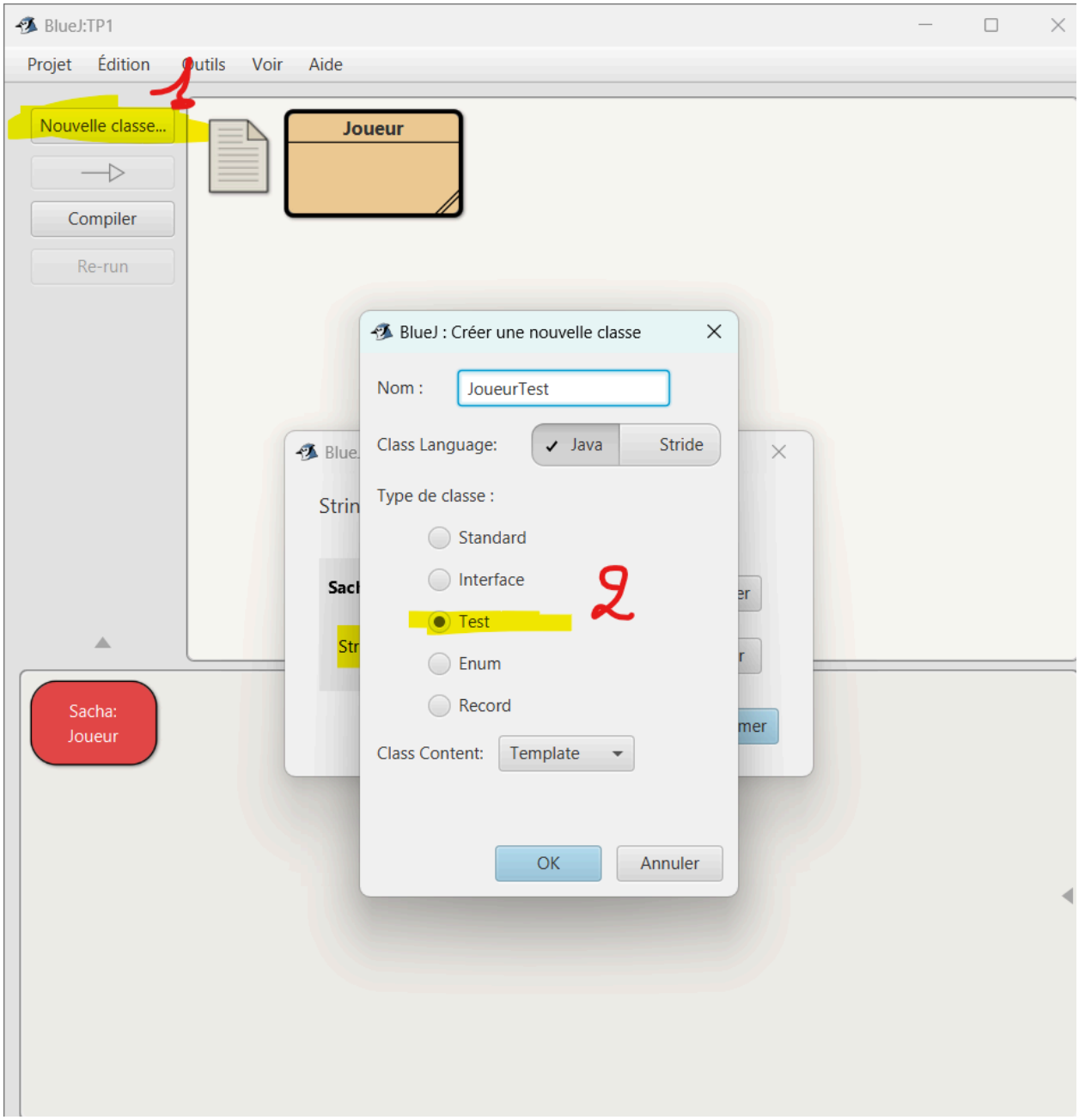




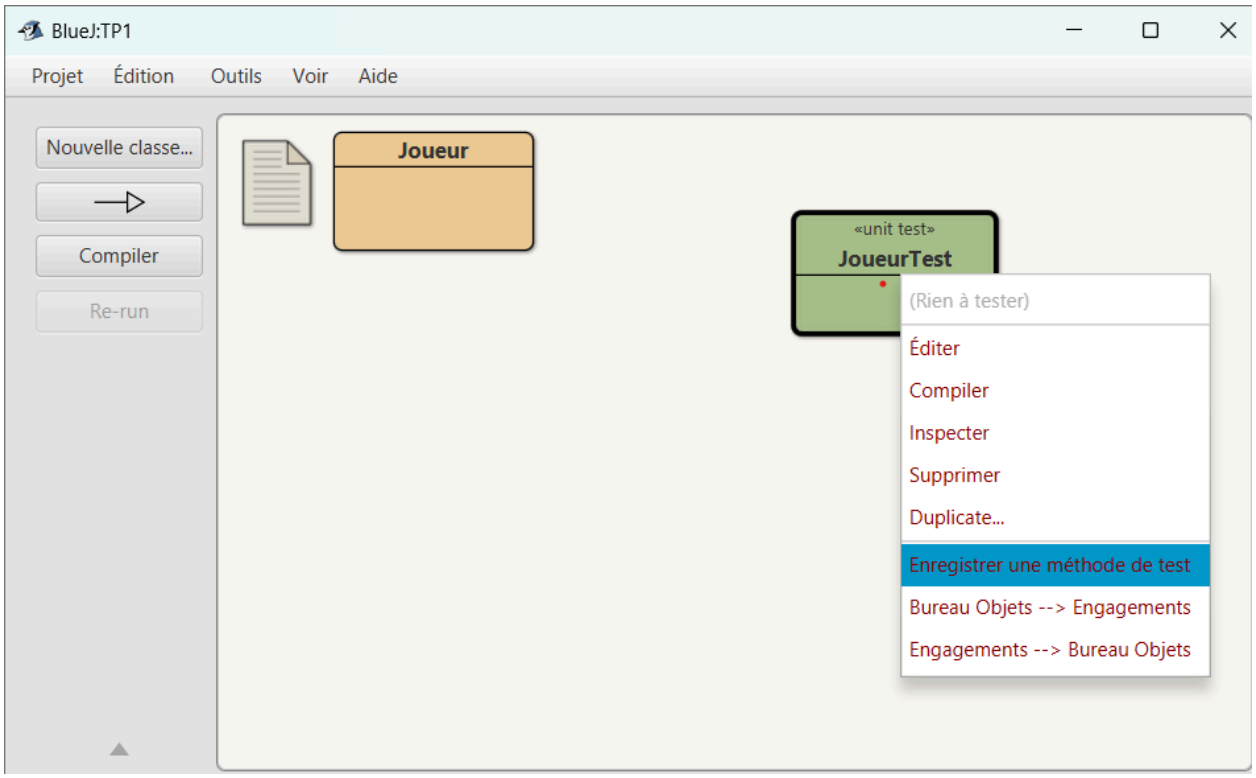


Ceci affiche les caractéristiques du Joueur.

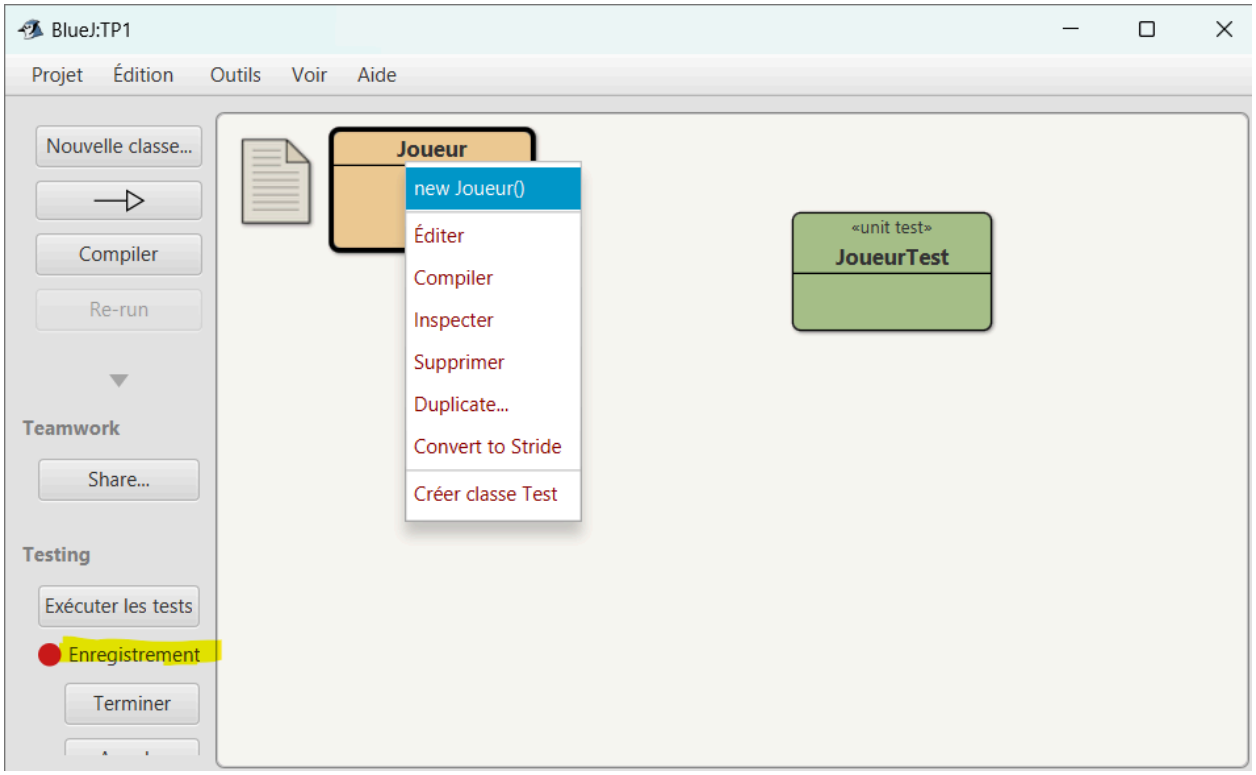
Étape 6 : Créer une classe de test



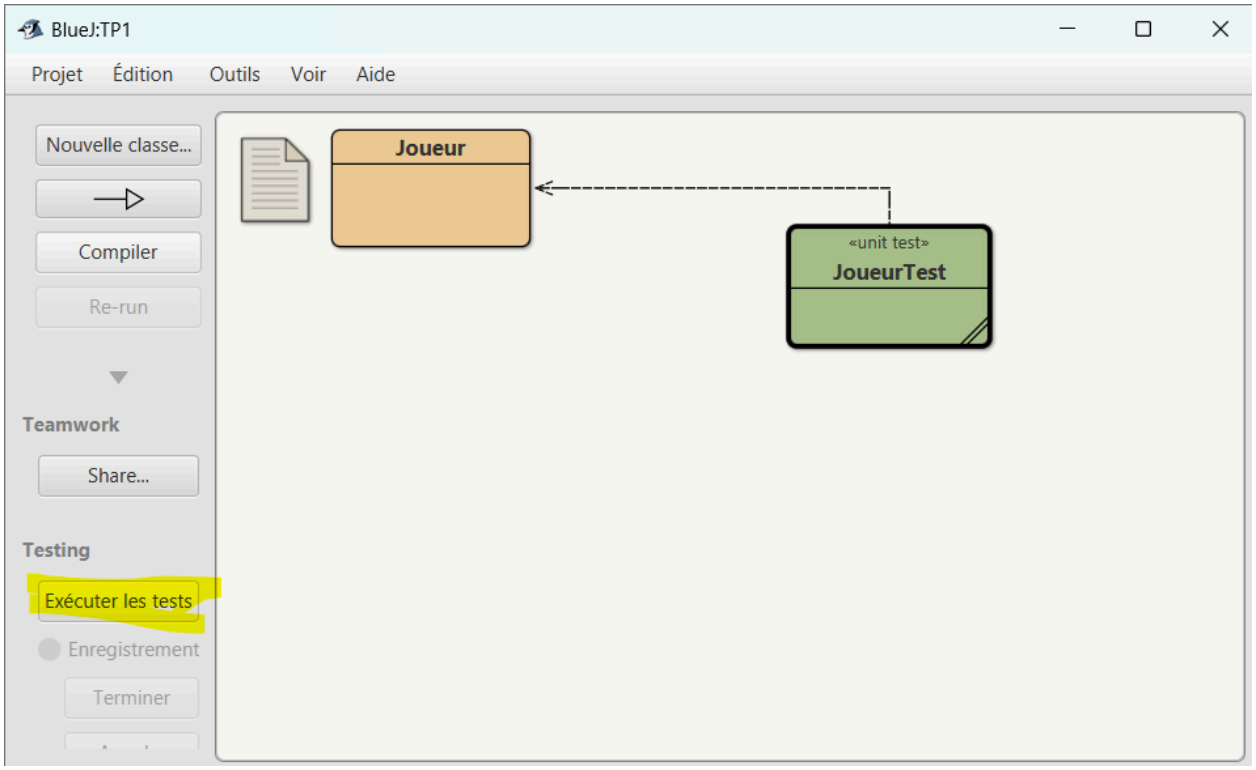
Créer une nouvelle classe de test



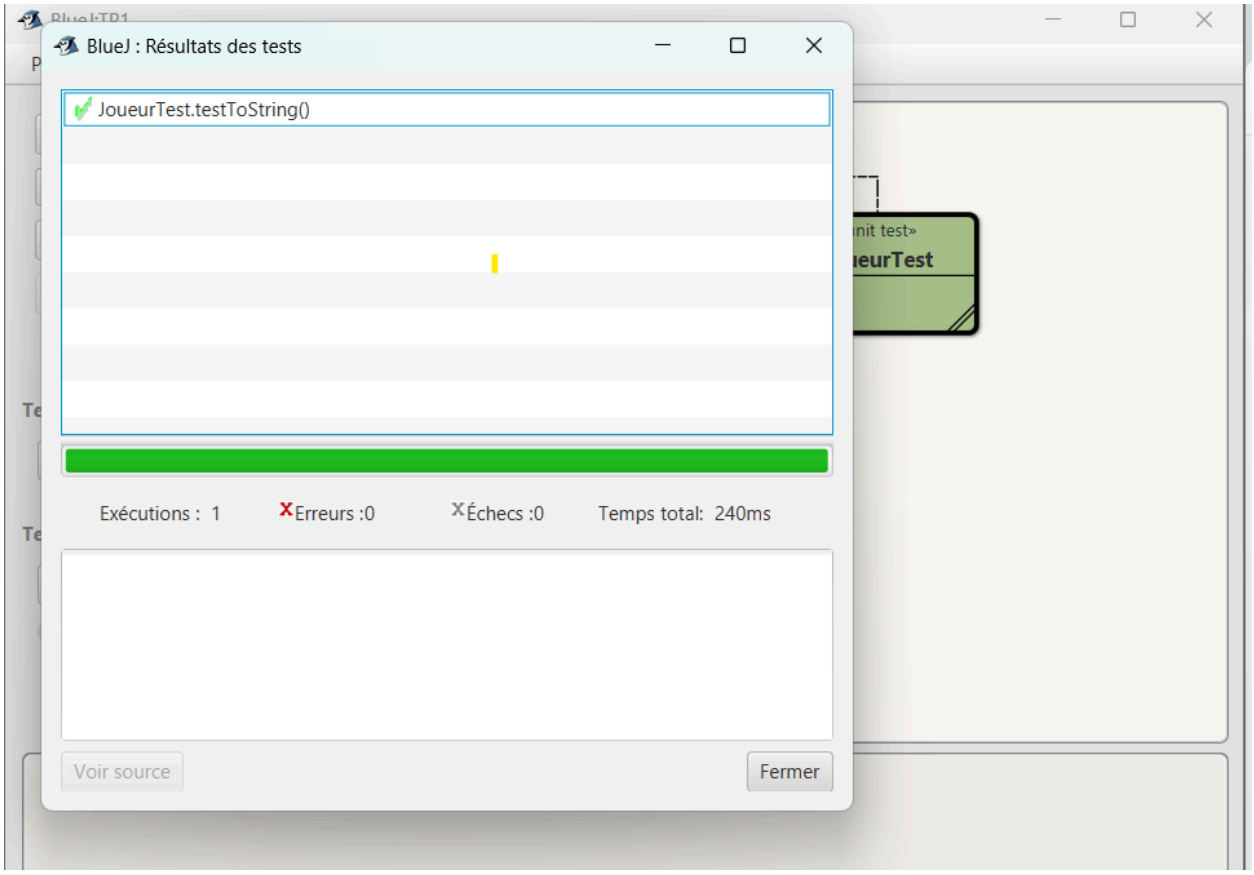
Enregistrer une méthode de test pour tester notre classe Joueur



Créer un nouveau joueur et utiliser la méthode toString > tout sera enregistré



Exécuter les tests

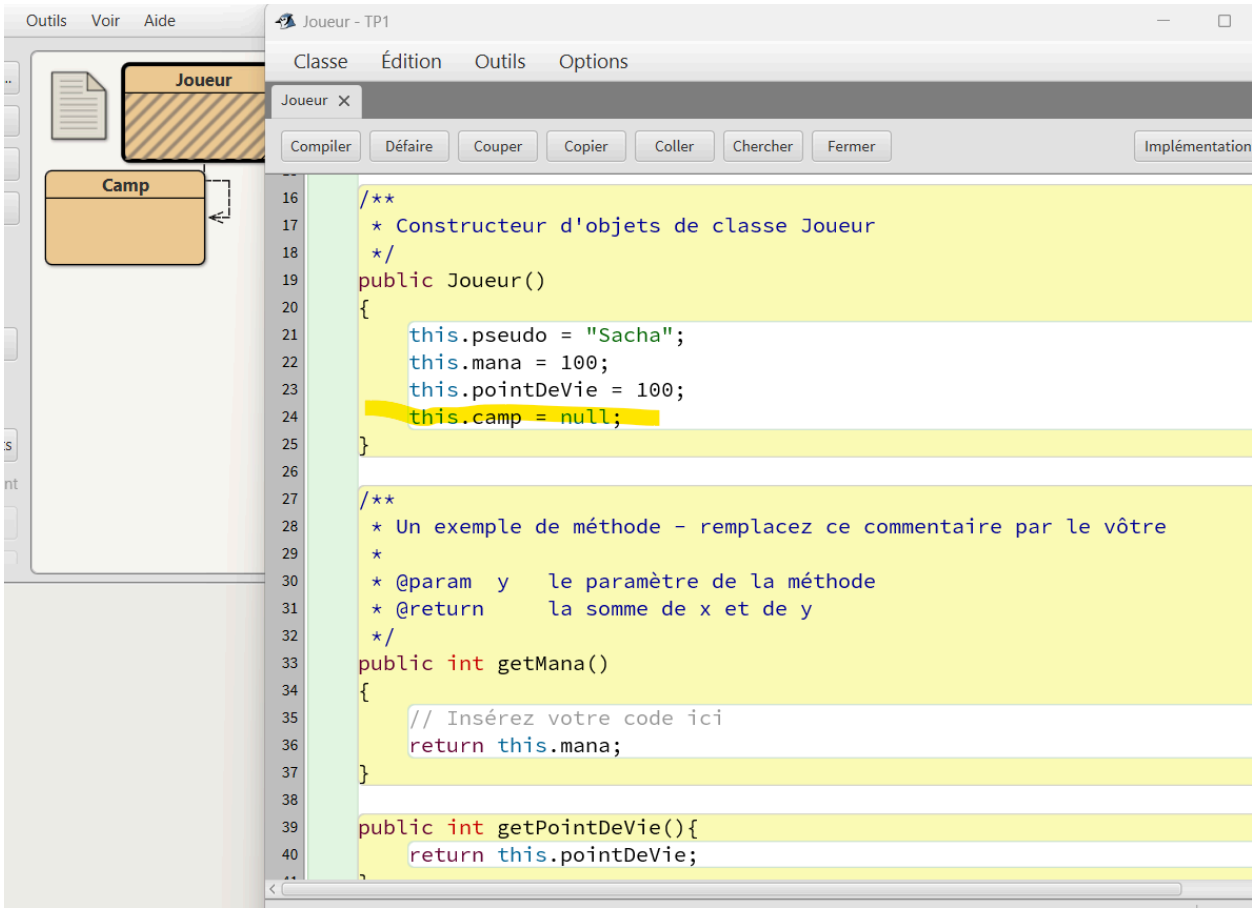


Tous les tests sont passés

À vous de jouer : Créer une classe Camp avec un attribut nom et une méthode setNom et une méthode getNom et une méthode toString

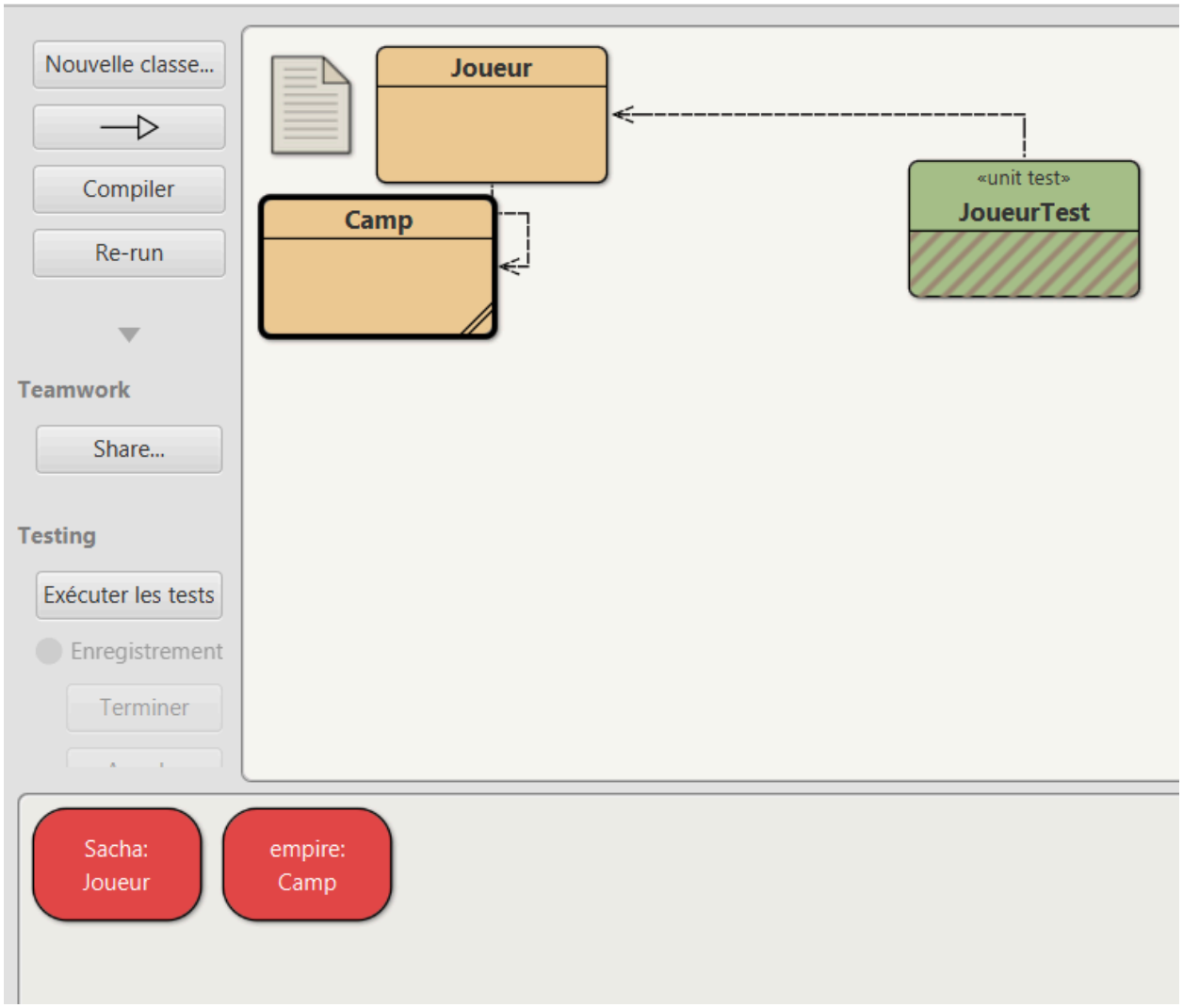
```
    private String nom = "Empire";

    /**
     * Constructeur d'objets de classe Camp
     */
    public Camp()
    {
        // initialisation des variables d'instance

    }
    public void setNom(String nom){
        this.nom = nom;
    }
    public String getNom(){
        return this.nom;
    }

    public int boostMana(){
        return 50;
    }

    public String toString(){
        return this.nom;
    }

    /**
     * Un exemple de méthode - remplacez ce commentaire par le vôtre
```

```
    public String getPseudo(){
        return this.pseudo;
    }

    public Camp getCamp(){
        return this.camp;
    }

    public void setCamp(Camp camp){
        this.camp = camp;
    }

    public int boostMana(){
        if(this.camp.getNom() == "Empire"){
            this.mana += this.camp.boostMana();
        }
        return this.mana;
    }
    public String toString(){
        StringBuilder s = new StringBuilder(this.pseudo + " PV " + this.pointDe
        " Mana " + this.mana);
        return s.toString();
    }
}
```

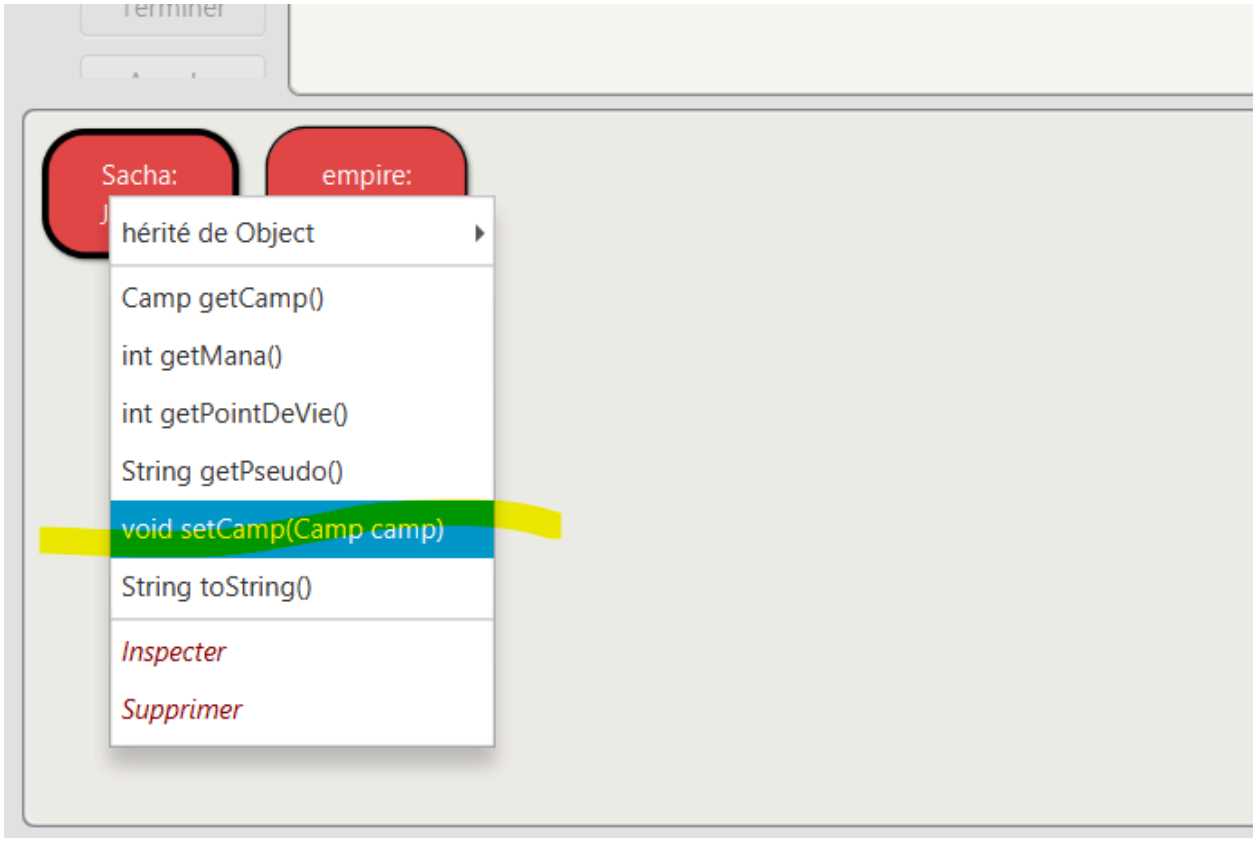

Outils
Voir
Aide
Joueur
Camp
Joueur - TP1
Classe
Édition
Outils
Options
Joueur
Compiler
Défaire
Couper
Copier
Coller
Chercher
Fermer
Implémentation
/**
* Constructeur d'objets de classe Joueur
*/
public Joueur()
{
this.pseudo = "Sacha";
this.mana = 100;
this.pointDeVie = 100;
this.camp = null;
}
/**
* 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 getMana()
{
// Insérez votre code ici
return this.mana;
}
public int getPointDeVie(){
return this.pointDeVie;

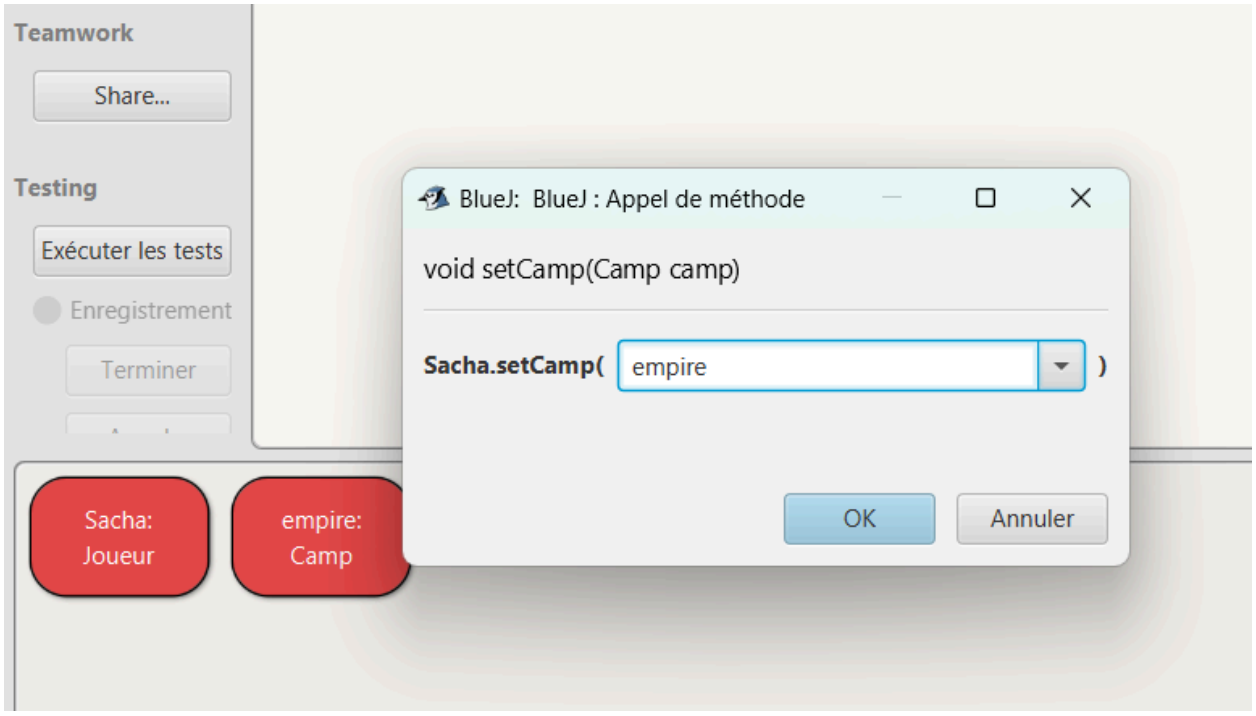
```
    public Camp getCamp(){
        return this.camp;
    }

    public void setCamp(Camp camp){
        this.camp = camp;
    }

    public int boostMana(){
        if(this.camp.getNom() == "Empire"){
            this.mana += this.camp.boostMana();
        }
        return this.mana;
    }
    public String toString(){
        StringBuilder s = new StringBuilder(this.pseudo + " PV " + this.point
        " Mana " + this.mana);
        if( this.camp != null){
            s.append("J'appartiens à l' " + this.camp);
        }
        return s.toString();
    }
}
```

on instancie les deux objets

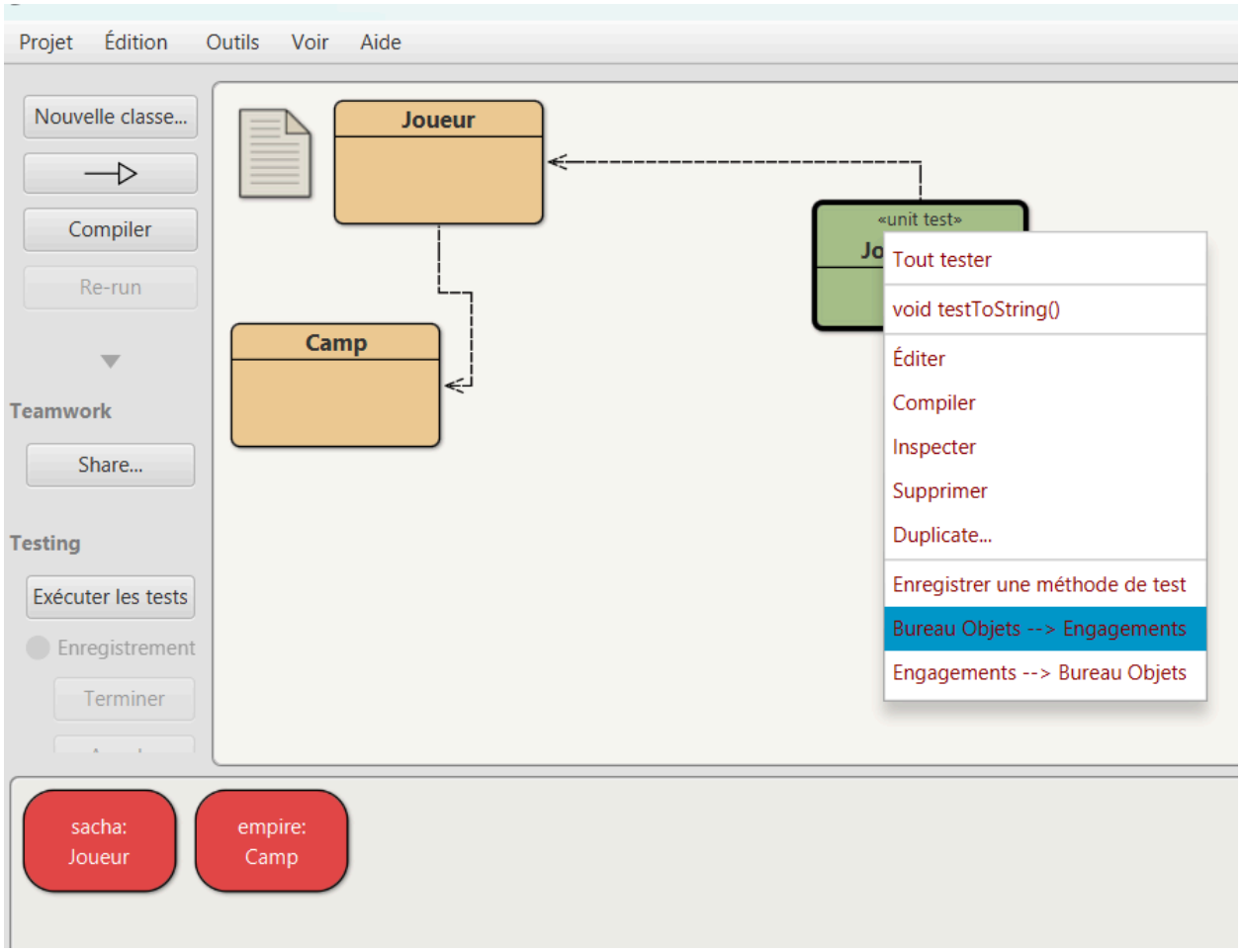

Nouvelle classe...
Compiler
Re-run
Teamwork
Share...
Testing
Exécuter les tests
Enregistrement
Terminer
Joueur
Camp
«unit test»
JoueurTest
Sacha:
Joueur
empire:
Camp

on les lie

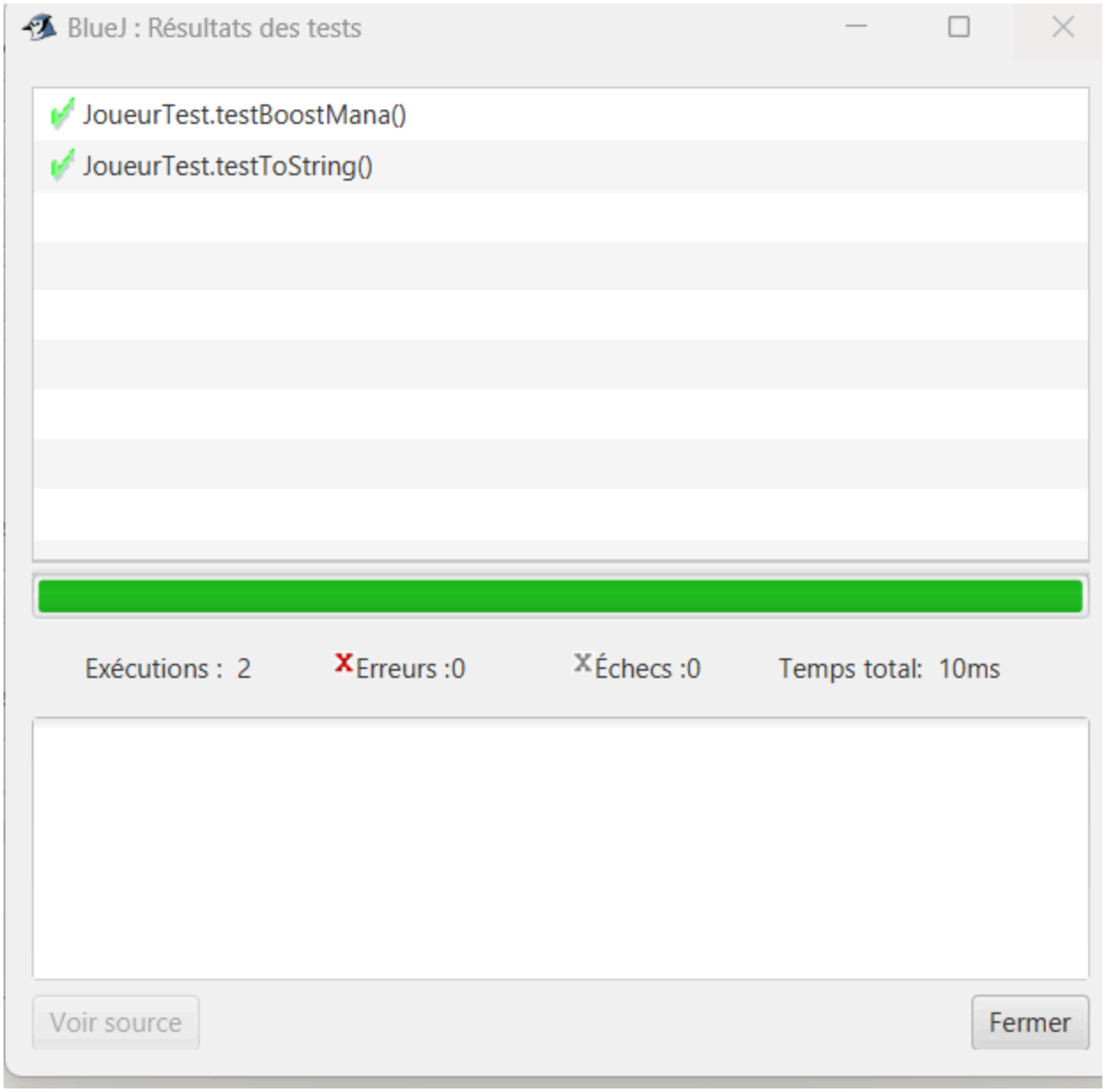




on affiche le resultat de .



À vous de jouer > Exécuter la méthode boostMana de la classe Joueur


BlueJ : Résultats des tests
JoueurTest.testBoostMana()
JoueurTest.testToString()
Exécutions : 2
Erreurs :0
Échecs :0
Temps total: 10ms
Voir source
Fermer