

L'atelier de Verrocchio

—

Ou l'école de la lutte contre les forces du mal



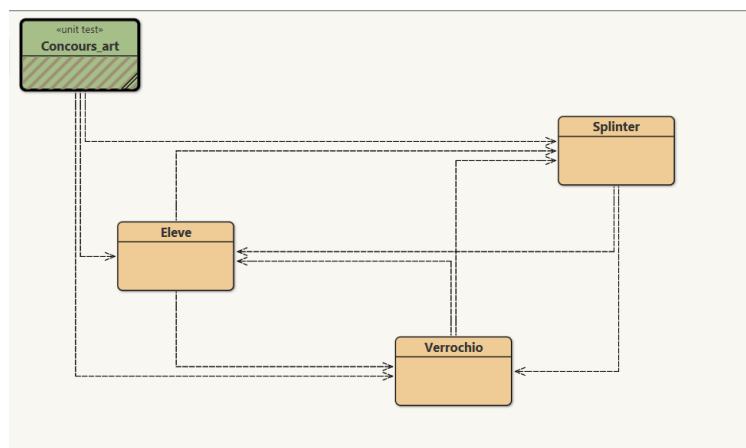
Est-ce bien sérieux de comparer le grand atelier de Verrocchio de la Renaissance qui a façonné de nombreux artistes plus talentueux les uns que les autres au point qu'ils ont façonné notre histoire, aux tortus ninjas qui vivent dans les égouts de Gotham City ?

Non évidemment.

Pourtant ces deux entités ont le point commun le plus noble de notre humanité : vaincre l'obscurantisme !

Nous allons présenter succinctement les différentes classes qui structurent notre projet d'atelier d'artistes.

Voici une vue d'ensemble préliminaire de notre organisation :



Nous avons d'abord la classe du maître Splinter qui a ouvert l'atelier de Verrocchio dans les souterrains de Florence :

```
public class Splinter
{
    // variables d'instance - remplacez l'exemple qui suit par le vôtre
    private String nom;
    private int anneeNaissance;
    private Eleve maitreDe;
    private Verrocchio atelier;
    /**
     * Constructeur d'objets de classe Splinter
     */
    public Splinter(String nom, int anneeNaissance) {
        this.nom = nom;
        this.anneeNaissance = anneeNaissance;
    }
    /**
     * getter and setter
     */
    public String getNom() {
        return this.nom;
    }

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

    /**
     * Fonction
     */
    public void prendreElev(Eleve artiste) {
        this.atelier.ajouterArtiste(artiste);
    }

    public void enCours() {
        this.maitreDe.apprentissage();
    }
}
```

Puis ensuite la classe de l'atelier Verrocchio que le maître Splinter a construit pour accueillir ses futurs apprentis :

```

import java.util.ArrayList;

public class Verrochio
{
    // variables d'instance
    private String nom;
    private int dateFondation;
    private ArrayList<Eleve> artistes;

    /**
     * Constructeur d'objets de classe Verrochio
     */
    public Verrochio(String nom, int dateFondation, ArrayList<Eleve> artistes) {
        this.nom = nom;
        this.dateFondation = dateFondation;
        this.artistes = new ArrayList<Eleve>();
    }

    /**
     * Getter et settrer
     */
    public String getNom() {
        return this.nom;
    }

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

    public int getDateFondation() {
        return this.dateFondation;
    }

    public void setDateFondation(int dateFondation) {
        this.dateFondation = dateFondation;
    }

    public ArrayList<Eleve> getArtistes() {
        return this.artistes;
    }

    public void setArtistes(ArrayList<Eleve> artistes) {
        this.artistes = artistes;
    }

    /**
     * Fonction de la classe
     */
    public void ajouterArtiste(Eleve nomArtiste) {
        this.artistes.add(nomArtiste);
    }

    public int nombreArtistes() {
        return this.artistes.size();
    }

    public void afficherArtistes() {
        System.out.println("Les artistes travaillant dans l'atelier sont : ");
        for (Eleve artiste : this.artistes) {
            System.out.println(artiste);
        }
    }
}

```

```

public static void main(String[] args) {
    ArrayList<Eleve> artistes = new ArrayList<Eleve>();
    Verrochio atelier = new Verrochio("Verrochio", 1438, artistes);
    Splinter master = new Splinter("Splinter", 1435);
    Eleve leonardo = new Eleve("Leonardo da Vinci", 1452, atelier, master);
    Eleve donatello = new Eleve("Donatello", 1452, atelier, master);
    Eleve raphaelo = new Eleve("Raphaelo", 1452, atelier, master);
    Eleve michelAngelo = new Eleve("Michel Angelo", 1452, atelier, master);
    atelier.ajouterArtiste(leonardo);
    atelier.ajouterArtiste(donatello);
    atelier.ajouterArtiste(raphaelo);
    atelier.ajouterArtiste(michelAngelo);
    System.out.println("L'atelier a " + atelier.nombreArtistes() + " artistes.");

    leonardo.dessiner();
    michelAngelo.apprentissage();
}
}

```

Et enfin la classe Eleve, qui permettra au maître Splinter de former ses élèves :

```

public class Eleve
{
    // variables d'instance - remplacez l'exemple qui suit par le vôtre
    private String nom;
    private int anneeNaissance;
    private Verrochio atelier;
    private Splinter master;
    /**
     * Constructeur d'objets de classe Leonardo
     */
    public Eleve(String nom, int anneeNaissance, Verrochio atelier, Splinter master) {
        this.nom = nom;
        this.anneeNaissance = anneeNaissance;
        this.atelier = atelier;
        this.master = master;
    }

    public void dessiner() {
        System.out.println(this.nom + " dessine dans l'atelier " + this.atelier.getNom());
    }

    public void apprentissage() {
        System.out.println(this.nom + " apprend à dessiner dans l'atelier " + this.atelier.getNom());
    }
}

```

Voilà les bases de construction de notre atelier de surhumain prête à se battre pour la survie de la dignité humaine !

Notre atelier est donc jeune et vient d'être créé. Il faut donc laisser le temps pour de nouvelle fonctionnalité.

Tout d'abord, testons la viabilité de notre atelier Verrocchio, pour savoir dans un premier temps si maître Splinter peut accueillir ces élèves correctement :

```

import static org.junit.jupiter.api.Assertions.assertEquals;

import java.util.ArrayList;

import org.junit.jupiter.api.AfterEach;
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Test;

public class Concours_art {

    protected double fValeur1;
    protected double fValeur2;

    public Concours_art() {

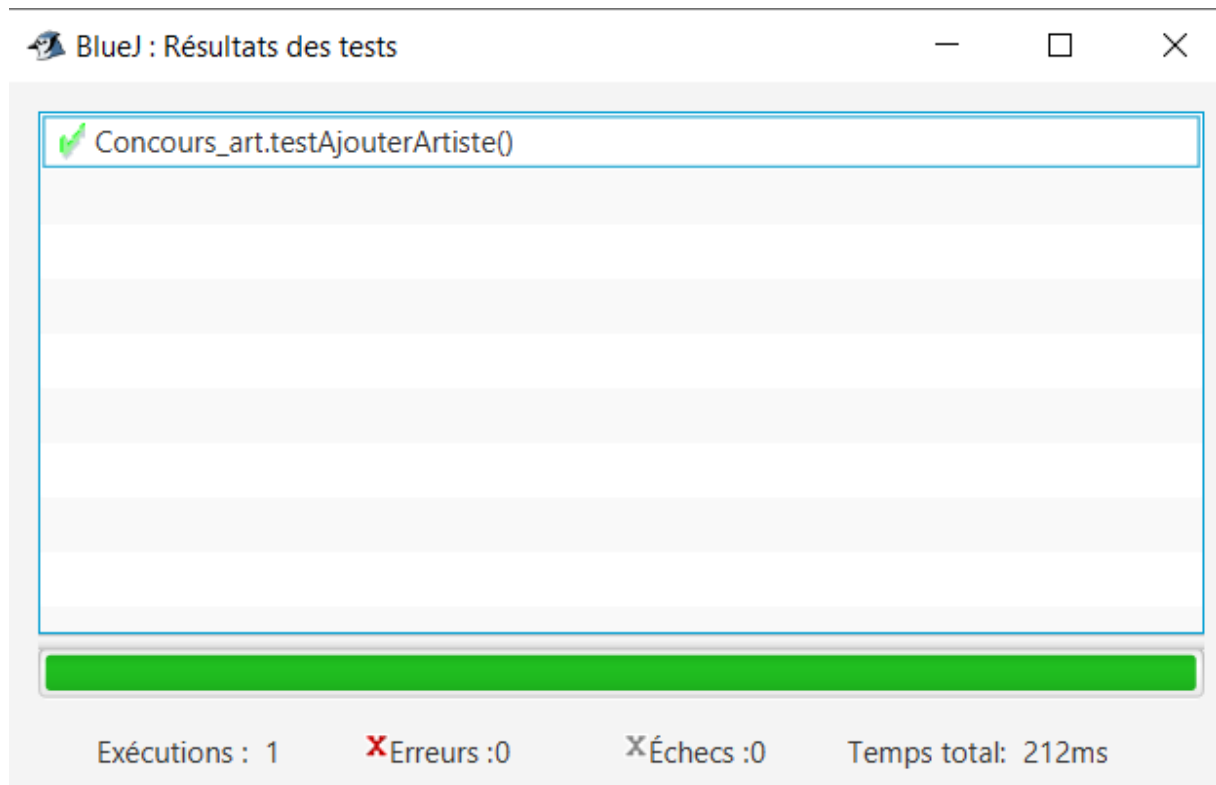
    }

    @BeforeEach
    public void setUp() // throws java.lang.Exception
    {
        // Initialisez ici vos engagements
        fValeur1 = 2.0;
        fValeur2 = 3.0;
    }

    @AfterEach
    public void tearDown() // throws java.lang.Exception
    {
        //Libérez ici les ressources engagées par setUp()
    }

    @Test
    public void testAjouterArtiste() {
        ArrayList<Eleve> artistes = new ArrayList<Eleve>();
        Splinter master = new Splinter("Splinter", 1435);
        Verrochio atelier = new Verrochio("Verrochio", 1438, artistes);
        Eleve artiste1 = new Eleve("Leonardo da Vinci", 1452, atelier, master);
        Eleve artiste2 = new Eleve("Donatello", 1386, atelier, master);
        atelier.ajouterArtiste(artiste1);
        atelier.ajouterArtiste(artiste2);
        assertEquals(2, atelier.nombreArtistes());
    }
}

```



Le test est passé avec succès : Splinter peut enfin ouvrir l'atelier Verrocchio !

Voyons cela dans les jours à venir comment l'atelier évoluera. Pour le moment nos artistes en herbe prêt à remodeler le monde pour le meilleurs, commence les rudiments du dessin.