

BlueJ: BlueJ : Créer un objet

Constructeur d'objets de classe *Voyage*

**Voyage()**

Nom de l'instance :

BlueJ: BlueJ : Créer un objet

**Voyage(String nom, String ville)**

Nom de l'instance :

**new Voyage(**  **,**  
 **)**

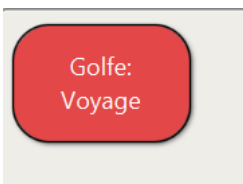
```
/**
 * La classe Voyage représente un voyage avec un nom (par exemple, le nom de l
 * et une ville de destination.
 *
 * Exemple : un voyage nommé "Vacances" à "Paris".
 *
 * Auteur : (votre nom)
 * Version : (votre version ou la date)
 */
public class Voyage {
    // Variables d'instance
    private String nom;
    private String villeDestination; |
```

```
// Getters
public String getNom() {
    return nom;
}

public String getVilleDestination() {
    return villeDestination;
}

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

public void setVilleDestination(String villeDestination) {
    this.villeDestination = villeDestination;
}
```

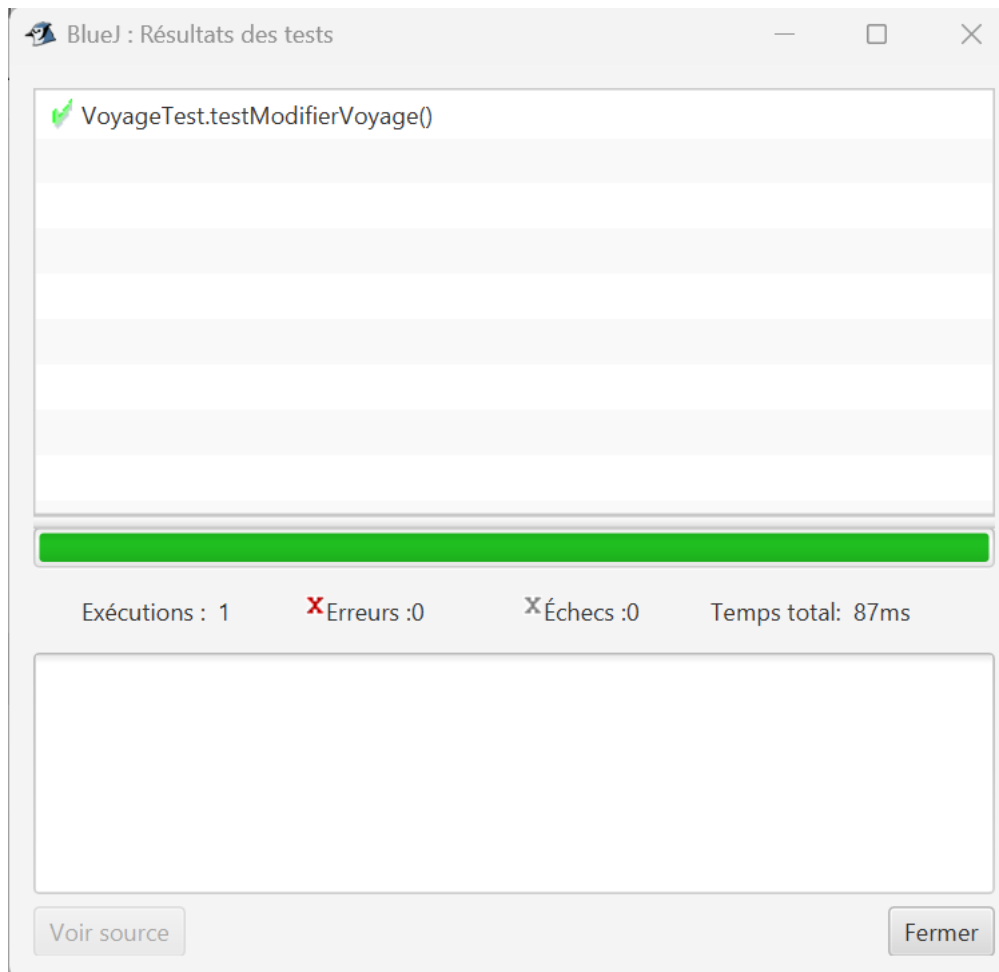


```
public void modifierVoyage(String nouveauNom, String nouvelleVille) {
    this.nom = nouveauNom;
    this.villeDestination = nouvelleVille;
}
```

```
/**
 * Teste la méthode modifierVoyage.
 */
@Test
public void testModifierVoyage() {
    assertEquals("Voyage initial", voyage.getNom());
    assertEquals("Londres", voyage.getVilleDestination());

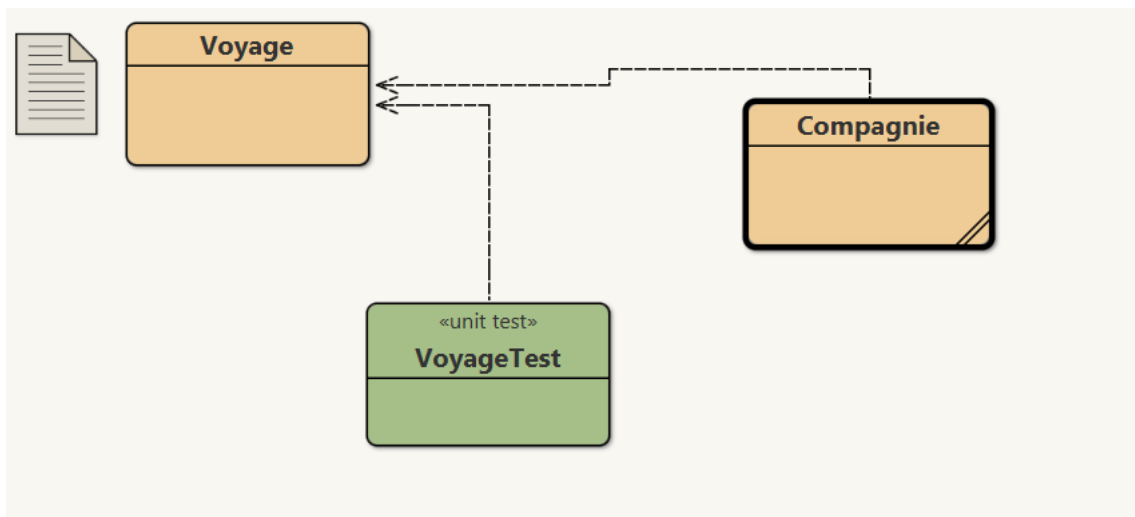
    voyage.modifierVoyage("Voyage modifié", "Rome");

    assertEquals("Voyage modifié", voyage.getNom());
    assertEquals("Rome", voyage.getVilleDestination());
}
```

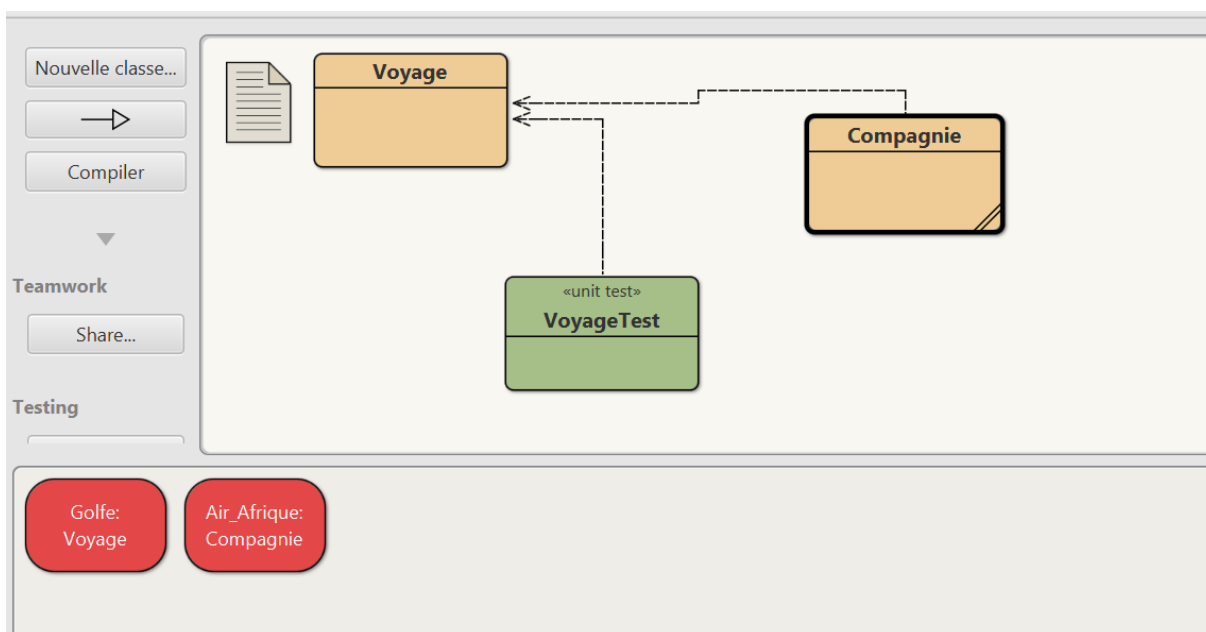


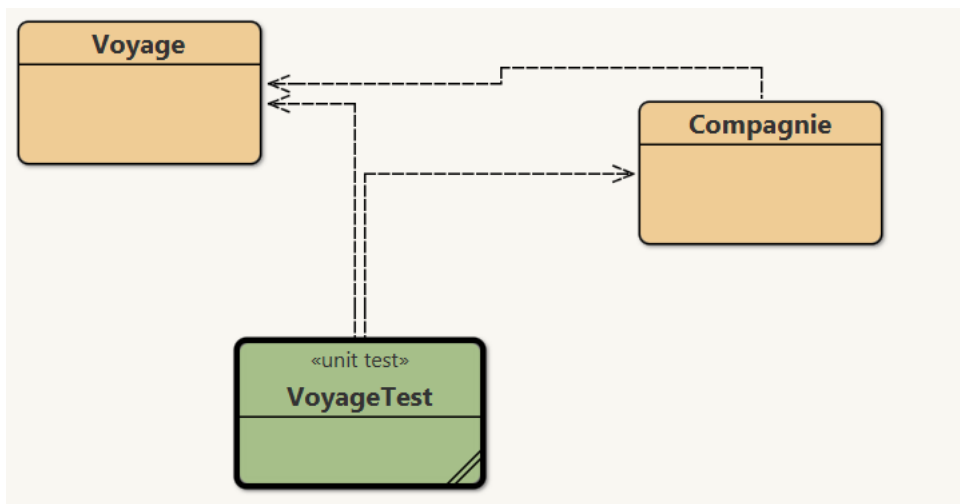
```
import java.util.List;  
import java.util.ArrayList;
```

```
/**  
 * Décrivez votre classe Compagnie ici.  
 *  
 * @author (votre nom)  
 * @version (un numéro de version ou une date)  
 */  
public class Compagnie  
{  
    // variables d'instance - remplacez l'exemple qui suit par le vôtre  
    private List<Voyage> voyage;  
    private String name;  
  
    /**  
     * Constructeur d'objets de classe Compagnie  
     */  
    public Compagnie()  
    {  
        // initialisation des variables d'instance  
        this.voyage=new ArrayList<>();  
        this.name="";  
    }  
}
```



```
public void compagnieVoyage(Voyage c)
{
    this.voyage.add(c);
}
```





```
@Test
public void testCompagnieVoyage() {
    assertEquals(0, Air_Afrique.getVoyages().size());

    Air_Afrique.compagnieVoyage(Golfe);

    assertEquals(1, Air_Afrique.getVoyages().size());

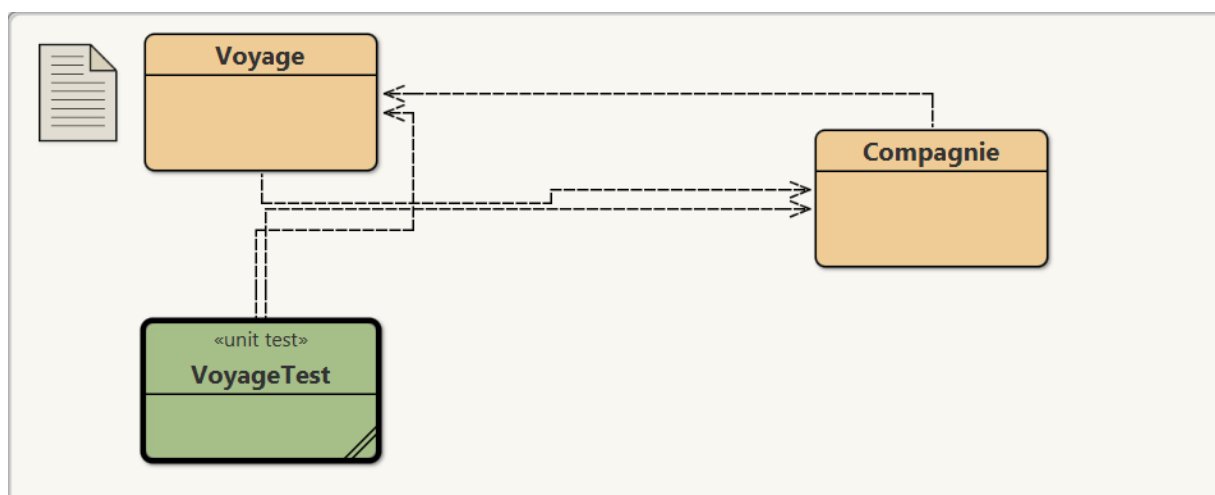
    assertEquals("Londres", Air_Afrique.getVoyages().get(0).getVilleDestination());
    assertEquals("Voyage initial", Air_Afrique.getVoyages().get(0).getNom());
}
```

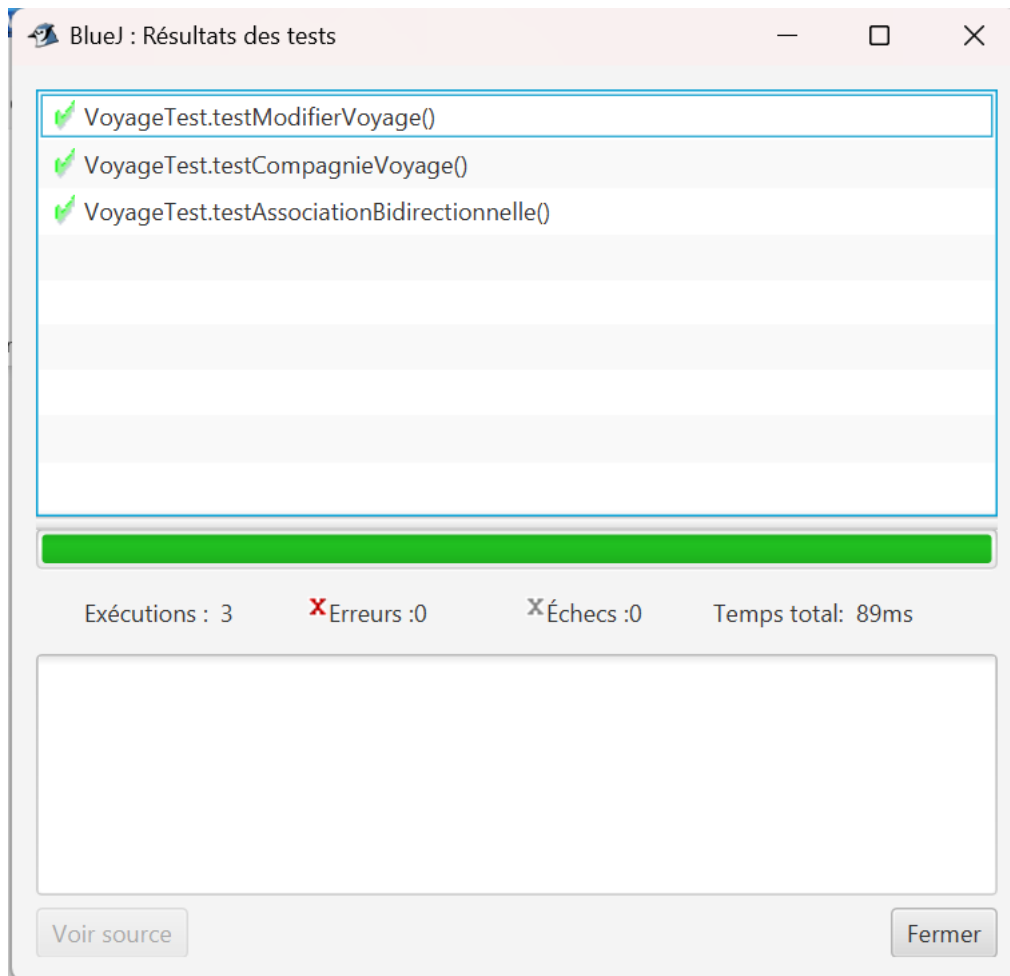
BlueJ : Résultats des tests

✓ VoyageTest.testModifierVoyage()  
✓ VoyageTest.testCompagnieVoyage()

Exécutions : 2    ✖ Erreurs :0    ✖ Échecs :0    Temps total: 12ms

Voir source    Fermer





Avant

```
/**
 * 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 compagnieVoyage(Voyage c)
{
    this.voyage.add(c);
}
```

Après

```
public void ajouterVoyageSimple(Voyage voyage) {
    this.voyage.add(voyage);
}
```

Avant



```

public String afficherVoyages() {
    if (voyage.isEmpty()) {
        return "Compagnie " + name + " : Aucun voyage disponible.";
    }

    StringBuilder sb = new StringBuilder();
    sb.append("Compagnie ").append(name).append(" :\n");
    for (Voyage v : voyage) {
        sb.append("- ").append(v.getNom())
          .append(" vers ").append(v.getVilleDestination())
          .append("\n");
    }
    return sb.toString();
}

```

Après

```

public String afficherVoyages() {
    if (voyage.isEmpty()) {
        return "Compagnie " + name + " : Aucun voyage disponible.";
    }

    return "Compagnie " + name + " :\n" + voyagesToString();
}

private String voyagesToString() {
    StringBuilder sb = new StringBuilder();
    for (Voyage v : voyage) {
        sb.append("- ").append(v.getNom())
          .append(" vers ").append(v.getVilleDestination())
          .append("\n");
    }
    return sb.toString();
}

```