

# PROJET HILISIT

Hybrldation en Llcence ScLenTifique

Financé  
par



**GOVERNEMENT**

*Liberté  
Égalité  
Fraternité*



Financé par  
l'Union européenne  
NextGenerationEU

**Modules Informatiques — Usages des modules POO Java  
et SDD C++ avec Caséine**

 **FST**  
Faculté des Sciences et Techniques  
UNIVERSITÉ HAUTE-ALSACE

- **Programmation Orientée Objet et Java (6 ECTS)**
- **Structures de données en C++ (3 ECTS)**

# Modules informatiques

- Bases de données (6 ECTS)
- Réseaux (6 ECTS)
- Math pour l'info (6 ECTS)
- Programmation Python (3 ECTS)
- Programmation C++ (6 ECTS)
- **Programmation Orientée Objet et Java (6 ECTS)**
- **Structures de données en C++ (3 ECTS)**
- NSI (12 ECTS)

# Modules informatiques

- Bases de données (6 ECTS)
- Réseaux (6 ECTS)
- Math pour l'info (6 ECTS)
- Programmation Python (3 ECTS)
- Programmation C++ (6 ECTS)
- **Programmation Orientée Objet et Java (6 ECTS)**
- **Structures de données en C++ (3 ECTS)**
- NSI (12 ECTS)

Sur la plateforme *CaseInE*

# Modules informatiques

- Bases de données (6 ECTS)
- Réseaux (6 ECTS)
- Math pour l'info (6 ECTS)
- Programmation Python (3 ECTS)
- Programmation C++ (6 ECTS)
- **Programmation Orientée Objet et Java (6 ECTS)**
- **Structures de données en C++ (3 ECTS)**
- NSI (12 ECTS)

Caseine x +

← → ↻ Q caseine.org →

Caseine Français (fr) ▾ Non connecté. (Connexion)

## Caseine

### Caseine



Caseine est une plateforme d'apprentissage en génie industriel, informatique et mathématiques. Son objectif est de favoriser l'apprentissage et l'autonomie des étudiants tout en améliorant la qualité du temps que l'enseignant leur consacre. Basée sur Moodle elle permet

1. d'évaluer automatiquement le code informatique et les modèles mathématiques des étudiants,
2. d'assurer un suivi pédagogique des étudiants,
3. de partager des contenus entre les enseignants à travers une communauté d'utilisateurs.

[Contacts](#) - [Infos](#) - [Les enseignants](#) - [Pour en savoir plus](#) - [Film de présentation](#)

**667** utilisateurs connectés au cours des dernières 24 heures (**8171** au cours du semestre écoulé)  
 **204** cours actifs ?  
 **3914** soumissions VPL quotidiennes en moyenne sur l'année écoulée



Caseine est une plateforme d'apprentissage en génie industriel, informatique et mathématiques. Son objectif est de favoriser l'apprentissage et l'autonomie des étudiants tout en améliorant la qualité du temps que l'enseignant leur consacre.

Basée sur Moodle elle permet

- 1 d'évaluer automatiquement le code informatique et les modèles mathématiques des étudiants,
- 2 d'assurer un suivi pédagogique des étudiants,
- 3 de partager des contenus entre les enseignants à travers une communauté d'utilisateurs.

# Visite anonyme



Fédération eduGain  $\Leftrightarrow$  Renater

## Connexion

- 1 Choisir « Autres comptes universitaires »
- 2 Sélectionner son établissement
- 3 Se connecter avec ses identifiants institutionnels

## Étudiantes et étudiants peuvent

- faire les exercices des cours ouverts,
- accéder à leur cours.

## Étudiantes et étudiants peuvent

- faire les exercices des cours ouverts,
- accéder à leur cours.

## Enseignantes et enseignants peuvent

- créer leurs cours,
- inviter leurs étudiants,
- échanger avec la communauté.

## Étudiantes et étudiants peuvent

- faire les exercices des cours ouverts,
- accéder à leur cours.

## Enseignantes et enseignants peuvent

- créer leurs cours,
- inviter leurs étudiants,
- échanger avec la communauté.

## La communauté :

- plus de 9 000 utilisateurs,
- plus de 150 enseignants,
- 14 universités,
- plus de 10 lycées.

# Trois promesses



Caseine est une plateforme d'apprentissage en génie industriel, informatique et mathématiques. Son objectif est de favoriser l'apprentissage et l'autonomie des étudiants tout en améliorant la qualité du temps que l'enseignant leur consacre.

Basée sur Moodle elle permet

- ① d'évaluer automatiquement le code informatique et les modèles mathématiques des étudiants,
- ② d'assurer un suivi pédagogique des étudiants,
- ③ de partager des contenus entre les enseignants à travers une communauté d'utilisateurs.

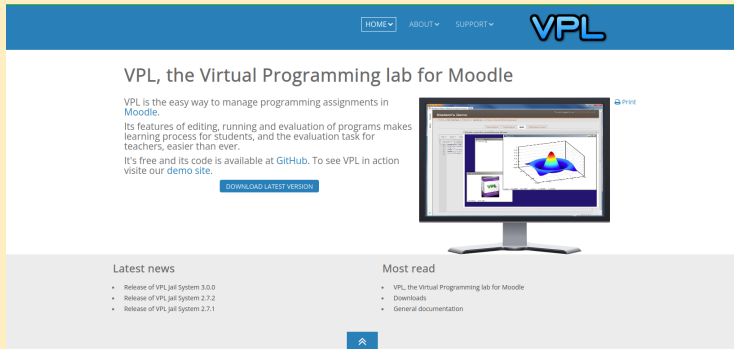
# 1. Évaluation automatique

Grâce aux activités Moodle.

# 1. Évaluation automatique

Grâce aux activités Moodle et en particulier le VPL.

<https://vpl.dis.ulpgc.es/>

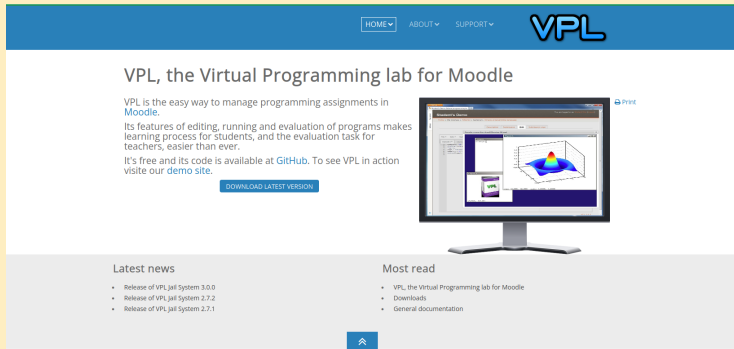


The screenshot shows the homepage of the VPL (Virtual Programming Lab) website. The header is blue with navigation links for HOME, ABOUT, and SUPPORT, and the VPL logo. The main content area has a white background with the title 'VPL, the Virtual Programming lab for Moodle'. Below the title, there is a paragraph describing VPL as an easy way to manage programming assignments in Moodle, followed by a list of features and a link to the demo site. A 'DOWNLOAD LATEST VERSION' button is present. To the right, there is a 'Print' icon and a monitor displaying a 3D surface plot. The footer is grey and contains two columns: 'Latest news' with three bullet points about releases (3.0.0, 2.7.2, 2.7.1) and 'Most read' with three bullet points (VPL description, Downloads, General documentation). A blue arrow icon is at the bottom center.

# 1. Évaluation automatique

Grâce aux activités Moodle et en particulier le VPL (**enrichi**).

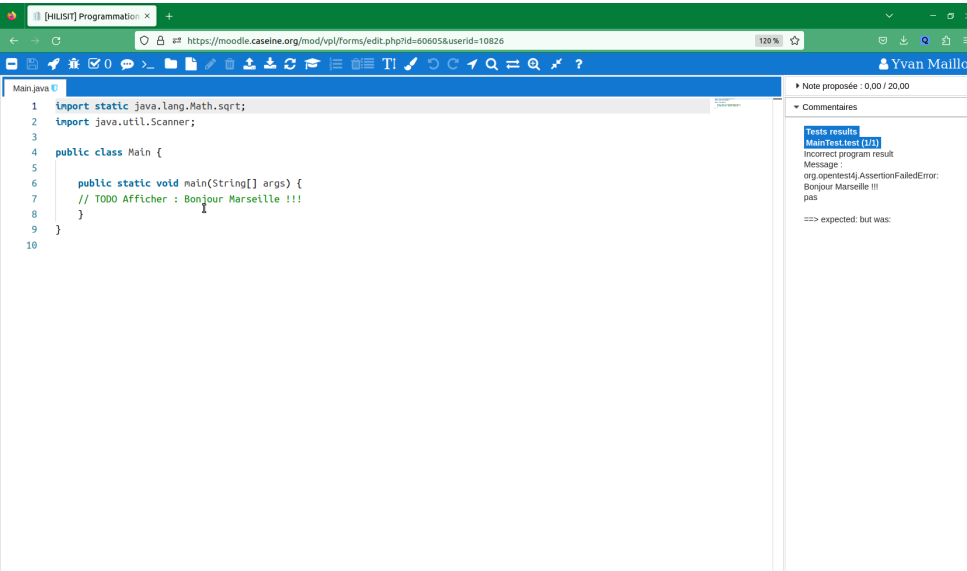
<https://vpl.dis.ulpgc.es/>



The screenshot shows the homepage of the VPL (Virtual Programming Lab) website. The header is blue with navigation links: HOME, ABOUT, and SUPPORT. The main heading is "VPL, the Virtual Programming lab for Moodle". Below this, there is a paragraph describing VPL as an easy way to manage programming assignments in Moodle, highlighting its features for editing, running, and evaluation. A "DOWNLOAD LATEST VERSION" button is present. To the right, there is a monitor displaying a 3D surface plot. Below the main content, there are two columns: "Latest news" with three bullet points about VPL jai! System releases (3.0.0, 2.7.2, 2.7.1) and "Most read" with three bullet points: "VPL, the Virtual Programming lab for Moodle", "Downloads", and "General documentation". A blue arrow button is at the bottom center.



# Accueil/Cours/Formations/HILISIT/POO- Java-1/Présentation/Bonjour Marseille



The screenshot shows a web browser window with a Moodle LMS interface. The browser address bar shows the URL: `https://moodle.caseine.org/mod/vpl/forms/edit.php?id=60605&userid=10826`. The user is identified as Yvan Maillot. The main content area is a code editor for a file named `Main.java`. The code is as follows:

```
1 import static java.lang.Math.sqrt;
2 import java.util.Scanner;
3
4 public class Main {
5
6     public static void main(String[] args) {
7         // TODO Afficher : Bonjour Marseille !!!
8     }
9 }
10
```

On the right side, there is a 'Note proposée' section showing a score of 0,00 / 20,00. Below it, the 'Commentaires' section displays the test results:

**Tests results**  
`MainTest.test [1/1]`  
Incorrect program result  
Message :  
org.opentest4j.AssertionFailedError:  
Bonjour Marseille !!!  
pas  
==> expected: but was:

# Accueil/Cours/Formations/HILISIT/POO- Java-1/Présentation/Bonjour Marseille

# Quelques expressions booléennes

Cours : [HILISIT] Program X +

https://moodle.caseine.org/course/view.php?id=899&section=2 120%

Caseine Français (fr) Yvan Mailliot Étudiant

[← Introduction](#)

## Présentation du langage

Le but de cette section est de faire

- un petit historique du langage,
- présenter succinctement ses caractéristiques
- et surtout d'apprendre à écrire son premier programme pour
  - afficher du texte à l'écran
  - et lire des données au clavier.

### Apprendre

 [Cours de présentation du langage \(version web\)](#)

### S'exercer

 [Bonjour Marseille](#)

 [Bonjour Qui ?](#)

 [Programme Minimum](#)

 [Quelques expressions booléennes](#)

 [Régions du plan](#)

# Quelques expressions booléennes

[HILISIT] Programmation X +

https://moodle.caseine.org/mod/vpl/forms/edit.php?id=60509&userid=10826 120%

Liste des fichiers

- edu
- euha
- miage
- QuelquesExpress
- Main.java

```
31 /**
32  * Écrivez le contenu des méthodes de cette classe pour qu'elles remplissent
33  * les contraintes demandées dans leur JavaDoc, en UNE SEULE INSTRUCTION.
34  */
35 public class QuelquesExpressionsBooleennes {
36
37     /**
38      * Méthode qui retourne vrai si son paramètre c est une lettre de l'alphabet
39      * latin sans accent et minuscule. Dans le cas contraire elle retourne faux.
40      *
41      * Le "défi" est de le faire en une seule instruction : le "return"
42      *
43      * @param c un caractère quelconque
44      * @return vrai si c est une lettre sans accent minuscule, faux sinon.
45      */
46     public static boolean estMinuscule(char c) {
47         return false;
48     }
49     /* TODO 1
50      * Écrire estMinuscule qui retourne vrai
51      * si c est une lettre minuscule
52      * et faux sinon.
53      * Pour simplifier l'exercice, les caractères accentués
54      * ne sont pas considérés comme minuscule
55     */
56 }
57
58 /**
59  * Méthode qui retourne vrai si son paramètre c n'est pas un chiffre
60  * décimal. faux sinon.
```

Note proposée : 0,00 / 20,00

Commentaires

Tests results

edu.uha.miage.QuelquesExpressionsBooleennes (1/12)

Incorrect program result

Message : org.opentest4j.AssertionFailedError: devrait être considéré comme un chiffre décimal ==> expected: but was: false

edu.uha.miage.QuelquesExpressionsBooleennes (2/12)

Incorrect program result

Message : org.opentest4j.AssertionFailedError: devrait être considéré comme un chiffre décimal ==> expected: but was: false

edu.uha.miage.QuelquesExpressionsBooleennes (3/12)

Incorrect program result

Message : org.opentest4j.AssertionFailedError: devrait pas être considéré comme un chiffre décimal ==> expected: but was: true

edu.uha.miage.QuelquesExpressionsBooleennes (4/12)

Incorrect program result

Message : org.opentest4j.AssertionFailedError: devrait pas être considéré comme un chiffre décimal ==> expected: but was: true

edu.uha.miage.QuelquesExpressionsBooleennes (5/12)

Incorrect program result

Message : org.opentest4j.AssertionFailedError: devrait être considéré comme un chiffre décimal ==> expected: but was: false

edu.uha.miage.QuelquesExpressionsBooleennes (6/12)

Incorrect program result

Message : org.opentest4j.AssertionFailedError: devrait être considéré comme un chiffre décimal ==> expected: but was: false

# Quelques expressions booléennes

# Plusieurs Classes

## Nombreux points d'évaluation

Java L3 (UHA) Des triangles plutôt classes Éditer - Google Chrome

Java L3 (UHA) Des triangles.p... x +

moodle.caseine.org/mod/vpl/forms/edit.php?id=16986&userid=10826

```
1 - *
2 * @author : Y
3 *
4 package edu.uh
5
6 public class P
7 // T000 1.
8 //
9 // T000 1.
10 //
11 // T000 1.
12 //
13 // T000 1.
14 //
15 // T000 1.
16 //
17 // T000 1.
18 //
19 // T000 1.
20 //
21 }
22
```

Note proposée: 0 / 20

Commentaires

Exécution

Description

On pourrait traduire cette définition par plusieurs modélisations orientées objets en Java, par exemple en mettant en attribut de la classe `Triangle` les 3 points. D'après la figure ci-dessus, on aurait un triangle défini par `p1`, `p2` et `p3` en vert.

On décide cependant ici de modéliser un triangle non pas par ses sommets, mais par ses côtés. Dans la figure ci-dessus, il s'agit de définir un triangle par les segments `s1`, `s2` et `s3` en bleu tels que

- `s1 = [p1, p2]`
- `s2 = [p2, p3]`
- `s3 = [p3, p1]`

Ainsi, les 3 attributs de la classe `Triangle` sont les 3 côtés du triangle.

En modélisant le problème à l'aide d'un diagramme de classes UML, on obtient le résultat ci-dessous que vous devez implémenter en totalité.

# Plusieurs Classes

## Nombreux points d'évaluation

Java L3 (UHA) Des triangles plutôt classes Éditer - Google Chrome

Java L3 (UHA) Des triangles p: x +

moodle.caseine.org/mod/vpl/forms/edit.php?id=16986&userid=10826


```
1- /*
2  * @author : Yvan Maillot (yvan.maillot@uha.fr)
3  */
4 package edu.uha.miage;
5
6 public class Point {
7     // TODO 1.01. Déclarer x et y, les coordonnées cartésiennes de ce point
8     private double x, y;
9     // TODO 1.02. Écrire le constructeur aux coordonnées cartésiennes
10
11     // TODO 1.03. Écrire un getter pour x
12
13     // TODO 1.04. Écrire un getter pour y
14
15     // TODO 1.05. Écrire un setter pour x
16
17     // TODO 1.06. Écrire un setter pour y
18
19     // TODO 1.07. public toString() qui retourne par exemple (2.5, 3.2)
20
21 }
22
```

Note proposée: 0 / 20

Commentaires

Exécution

Description



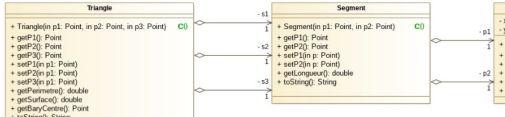
On pourrait traduire cette définition par plusieurs modélisations orientées objets en Java, par exemple en mettant en attribut de la classe Triangle le 3 points aurait un triangle défini par p1, p2 et p3 en vert.

On décide cependant ici de modéliser un triangle non pas par ses sommets, mais par ses côtés. Dans la figure ci-dessus, il s'agit de définir un triangle par bleu tels que

- s1 = [p1, p2]
- s2 = [p2, p3]
- s3 = [p3, p1]

Ainsi, les 3 attributs de la classe Triangle sont les 3 côtés du triangle.

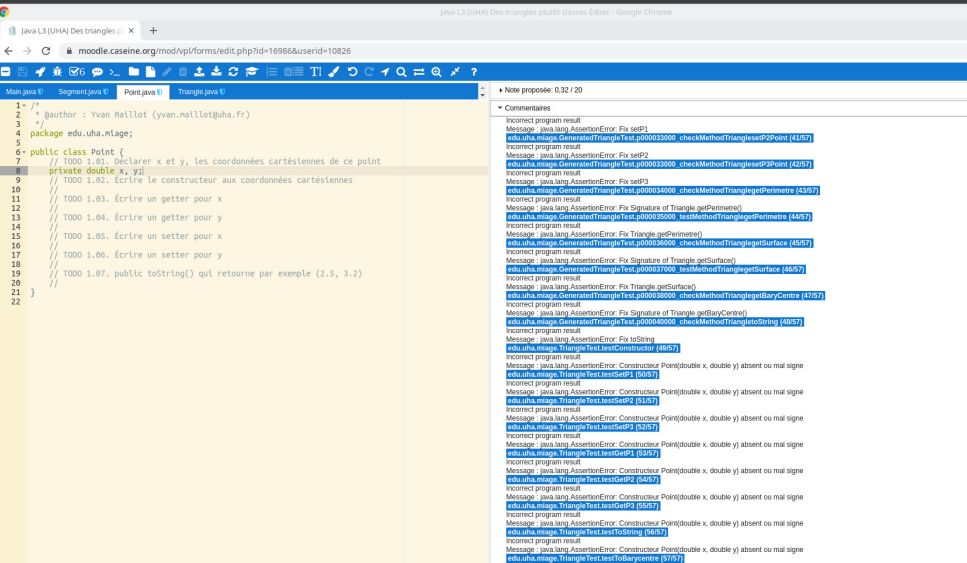
En modélisant le problème à l'aide d'un diagramme de classes UML, on obtient le résultat ci-dessous que vous devrez implémenter en totalité.



```
classDiagram
    class Triangle {
        + Point p1
        + Point p2
        + Point p3
        + getP1(): Point
        + getP2(): Point
        + getP3(): Point
        + setP1(p1: Point)
        + setP2(p2: Point)
        + setP3(p3: Point)
        + getPerimetre(): double
        + getSurface(): double
        + getBaryCentre(): Point
        + toString(): String
    }
    class Segment {
        + Point p1
        + Point p2
        + getP1(): Point
        + getP2(): Point
        + setP1(p1: Point)
        + setP2(p2: Point)
        + getLongueur(): double
        + toString(): String
    }
    Triangle "1" *-- "1" Segment : s1
    Triangle "1" *-- "1" Segment : s2
    Triangle "1" *-- "1" Segment : s3
```

# Plusieurs Classes

## Nombreux points d'évaluation



Java L3 (UHA) Des triangles plutôt classes Éditer - Google Chrome

java L3 (UHA) Des triangles p: x +

moodle.caseine.org/mod/vpl/forms/edit.php?id=16986&userid=10826

```
1 /*  
2  * @author : Yvan Maillot (yvan.maillot@uha.fr)  
3  */  
4 package edu.uha.mtage;  
5  
6 public class Point {  
7     // TODO 1.01. Déclarer x et y, les coordonnées cartésiennes de ce point  
8     private double x, y;  
9     // TODO 1.02. Écrire le constructeur aux coordonnées cartésiennes  
10    //  
11    // TODO 1.03. Écrire un getter pour x  
12    //  
13    // TODO 1.04. Écrire un getter pour y  
14    //  
15    // TODO 1.05. Écrire un setter pour x  
16    //  
17    // TODO 1.06. Écrire un setter pour y  
18    //  
19    // TODO 1.07. public toString() qui retourne par exemple (2.5, 3.2)  
20 }  
21  
22
```

Note proposée: 0,32 / 20

Commentaires

- Incorrect program result  
Message : java.lang.AssertionError: Fix setP1  
[edu.uha.mtage.GeneratedTriangleTest.p000033000\\_checkMethodTrianglesetP2Point \(41/57\)](#)  
Incorrect program result
- Message : java.lang.AssertionError: Fix setP2  
[edu.uha.mtage.GeneratedTriangleTest.p000033000\\_checkMethodTrianglesetP3Point \(42/57\)](#)  
Incorrect program result
- Message : java.lang.AssertionError: Fix setP3  
[edu.uha.mtage.GeneratedTriangleTest.p000034000\\_checkMethodTrianglegetPerimetre \(43/57\)](#)  
Incorrect program result
- Message : java.lang.AssertionError: Fix Signature of Triangle.getPerimetre()  
[edu.uha.mtage.GeneratedTriangleTest.p000035000\\_testMethodTrianglegetPerimetre \(44/57\)](#)  
Incorrect program result
- Message : java.lang.AssertionError: Fix Triangle.getPerimetre()  
[edu.uha.mtage.GeneratedTriangleTest.p000036000\\_checkMethodTrianglegetSurface \(45/57\)](#)  
Incorrect program result
- Message : java.lang.AssertionError: Fix Signature of Triangle.getSurface()  
[edu.uha.mtage.GeneratedTriangleTest.p000037000\\_testMethodTrianglegetSurface \(46/57\)](#)  
Incorrect program result
- Message : java.lang.AssertionError: Fix Triangle.getSurface()  
[edu.uha.mtage.GeneratedTriangleTest.p000038000\\_checkMethodTrianglegetBaryCentre \(47/57\)](#)  
Incorrect program result
- Message : java.lang.AssertionError: Fix Signature of Triangle.getBaryCentre()  
[edu.uha.mtage.GeneratedTriangleTest.p000040000\\_checkMethodTriangletoString \(48/57\)](#)  
Incorrect program result
- Message : java.lang.AssertionError: Fix toString  
[edu.uha.mtage.TriangleTest.testConstructor \(49/57\)](#)  
Incorrect program result
- Message : java.lang.AssertionError: Constructeur Point(double x, double y) absent ou mal signe  
[edu.uha.mtage.TriangleTest.testSetP1 \(50/57\)](#)  
Incorrect program result
- Message : java.lang.AssertionError: Constructeur Point(double x, double y) absent ou mal signe  
[edu.uha.mtage.TriangleTest.testSetP2 \(51/57\)](#)  
Incorrect program result
- Message : java.lang.AssertionError: Constructeur Point(double x, double y) absent ou mal signe  
[edu.uha.mtage.TriangleTest.testSetP3 \(52/57\)](#)  
Incorrect program result
- Message : java.lang.AssertionError: Constructeur Point(double x, double y) absent ou mal signe  
[edu.uha.mtage.TriangleTest.testGetP1 \(53/57\)](#)  
Incorrect program result
- Message : java.lang.AssertionError: Constructeur Point(double x, double y) absent ou mal signe  
[edu.uha.mtage.TriangleTest.testGetP2 \(54/57\)](#)  
Incorrect program result
- Message : java.lang.AssertionError: Constructeur Point(double x, double y) absent ou mal signe  
[edu.uha.mtage.TriangleTest.testGetP3 \(55/57\)](#)  
Incorrect program result
- Message : java.lang.AssertionError: Constructeur Point(double x, double y) absent ou mal signe  
[edu.uha.mtage.TriangleTest.testToString \(56/57\)](#)  
Incorrect program result
- Message : java.lang.AssertionError: Constructeur Point(double x, double y) absent ou mal signe  
[edu.uha.mtage.TriangleTest.testGetBarycentre \(57/57\)](#)



# Plusieurs Classes

## Nombreux points d'évaluation

Java L3 (UHA) Des triangles plutôt classes Éditer - Google Chrome

java L3 (UHA) Des triangles p... x +

moodle.caseine.org/mod/vpl/forms/edit.php?id=16986&userid=10826

```
main.java v Segmant.java v Point.java v Triangle.java v
1 /*
2  * @author : Yvan Maillot (yvan.maillot@uha.fr)
3  */
4 package edu.uha.miage;
5
6 public class Point {
7
8     private double x, y;
9
10    public Point(double x, double y) {
11        this.x = x;
12        this.y = y;
13    }
14
15    public double getX() {
16        return x;
17    }
18
19    public double getY() {
20        return y;
21    }
22
23    public void setX(double x) {
24        this.x = x;
25    }
26
27    public void setY(double y) {
28        this.y = y;
29    }
30
31    @Override
32    public String toString() {
33        // return "(" + x + " " + y + ")";
34        return String.format("(%.1f, %.1f)", x, y);
35    }
36 }
37 }
```

Note proposée: 2.88 / 20

Commentaires

Tests results

- edu.uha.miage.GeneratedPointTest.p00001000\_checkPointFieldX (1/57)
- edu.uha.miage.GeneratedPointTest.p00001000\_checkPointFieldY (2/57)
- edu.uha.miage.GeneratedPointTest.p00002000\_checkConstructorPointtoDoubledouble (3/57)
- edu.uha.miage.GeneratedPointTest.p00003000\_checkMethodPointGetX (4/57)
- edu.uha.miage.GeneratedPointTest.p00003000\_checkMethodPointGetY (5/57)
- edu.uha.miage.GeneratedPointTest.p00004000\_checkMethodPointSetXdouble (6/57)
- edu.uha.miage.GeneratedPointTest.p00004000\_checkMethodPointSetYdouble (7/57)
- edu.uha.miage.GeneratedPointTest.p00005000\_checkMethodPointToString (8/57)
- edu.uha.miage.GeneratedPointTest.p00006000\_checkGetterPointX (9/57)
- edu.uha.miage.GeneratedPointTest.p00006000\_checkGetterPointY (10/57)
- edu.uha.miage.GeneratedPointTest.p00007000\_checkSetterPointX (11/57)
- edu.uha.miage.GeneratedPointTest.p00007000\_checkSetterPointY (12/57)
- edu.uha.miage.PointTest.testConstructor (13/57)
- edu.uha.miage.GeneratedSegmentTest.p000020000\_checkSegmentFieldP1 (15/57)
- Incorrect program result
- Message : java.lang.AssertionError:
- edu.uha.miage.GeneratedSegmentTest.p000020000\_checkSegmentFieldP2 (16/57)
- Incorrect program result
- Message : java.lang.AssertionError:
- edu.uha.miage.GeneratedSegmentTest.p000021000\_checkConstructorSegmentPointPoint (17/57)
- Incorrect program result
- Message : java.lang.AssertionError: Constructor of edu.uha.miage.Segment
- edu.uha.miage.GeneratedSegmentTest.p000022000\_checkMethodSegmentGetP1 (18/57)
- Incorrect program result
- Message : java.lang.AssertionError: Fix getP1
- edu.uha.miage.GeneratedSegmentTest.p000022000\_checkMethodSegmentGetP2 (19/57)
- Incorrect program result
- Message : java.lang.AssertionError: Fix getP2
- edu.uha.miage.GeneratedSegmentTest.p000023000\_checkMethodSegmentSetP1Point (20/57)
- Incorrect program result
- Message : java.lang.AssertionError: Fix setP1
- edu.uha.miage.GeneratedSegmentTest.p000023000\_checkMethodSegmentSetP2Point (21/57)
- Incorrect program result
- Message : java.lang.AssertionError: Fix setP2
- edu.uha.miage.GeneratedSegmentTest.p000024000\_checkMethodSegmentLength (22/57)
- Incorrect program result
- Message : (java.lang.AssertionError: Fix Signature length) of Segment
- edu.uha.miage.GeneratedSegmentTest.p000025000\_checkMethodSegmentProjectionLengthOnX (23/57)
- Incorrect program result
- Message : java.lang.AssertionError: Fix Signature projectionLengthOnX() of Segment
- edu.uha.miage.GeneratedSegmentTest.p000025000\_checkMethodSegmentToString (24/57)
- Incorrect program result
- Message : java.lang.AssertionError: Fix toString
- edu.uha.miage.GeneratedSegmentTest.p000026000\_checkGetterSegmentP1 (25/57)
- Incorrect program result
- Message : java.lang.AssertionError: Fix getP1
- edu.uha.miage.GeneratedSegmentTest.p000026000\_checkGetterSegmentP2 (26/57)
- Incorrect program result
- Message : java.lang.AssertionError: Fix getP2

# Plusieurs Classes

## Nombreux points d'évaluation

Java L3 (UHA) Des triangles plutôt classes Éditer - Google Chrome

Java L3 (UHA) Des triangles plutôt classes

moodle.caseine.org/mod/vpl/forms/edit.php?id=16986&userid=10826

Segment.java Point.java Triangle.java

Note proposée: 10,08 / 20

Commentaires

Tests results

```
edu.uha.miage.GeneratedPointTest.p00001000 checkPointFieldX (1/57)
edu.uha.miage.GeneratedPointTest.p00001009 checkPointFieldY (2/57)
edu.uha.miage.GeneratedPointTest.p00002000 checkConstructorPointDoubleDouble (3/57)
edu.uha.miage.GeneratedPointTest.p00003000 checkMethodPointGet (4/57)
edu.uha.miage.GeneratedPointTest.p00003000 checkMethodPointGet (5/57)
edu.uha.miage.GeneratedPointTest.p00004000 checkMethodPointSetXDouble (6/57)
edu.uha.miage.GeneratedPointTest.p00004000 checkMethodPointSetYDouble (7/57)
edu.uha.miage.GeneratedPointTest.p00005000 checkMethodPointToString (8/57)
edu.uha.miage.GeneratedPointTest.p00006000 checkGetterPointX (9/57)
edu.uha.miage.GeneratedPointTest.p00006000 checkGetterPointY (10/57)
edu.uha.miage.GeneratedPointTest.p00007000 checkSetterPointX (11/57)
edu.uha.miage.GeneratedPointTest.p00007000 checkSetterPointY (12/57)
edu.uha.miage.PointTest.testConstructor (13/57)
edu.uha.miage.GeneratedSegmentTest.p00002000 checkSegmentFieldP1 (15/57)
edu.uha.miage.GeneratedSegmentTest.p00002000 checkSegmentFieldP2 (16/57)
edu.uha.miage.GeneratedSegmentTest.p000021000 checkConstructorSegmentPointPoint (17/57)
edu.uha.miage.GeneratedSegmentTest.p000022000 checkMethodSegmentGetP1 (18/57)
edu.uha.miage.GeneratedSegmentTest.p000022000 checkMethodSegmentGetP2 (19/57)
edu.uha.miage.GeneratedSegmentTest.p000023000 checkMethodSegmentGetP1Point (20/57)
edu.uha.miage.GeneratedSegmentTest.p000023000 checkMethodSegmentGetP2Point (21/57)
edu.uha.miage.GeneratedSegmentTest.p000024000 checkMethodSegmentLength (22/57)
edu.uha.miage.GeneratedSegmentTest.p000025000 checkMethodSegmentProjectionLengthOnX (23/57)
edu.uha.miage.GeneratedSegmentTest.p000025000 checkMethodSegmentToString (24/57)
edu.uha.miage.GeneratedSegmentTest.p000026000 checkGetterSegmentP1 (25/57)
edu.uha.miage.GeneratedSegmentTest.p000026000 checkGetterSegmentP2 (26/57)
edu.uha.miage.GeneratedSegmentTest.p000027000 checkSetterSegmentP1 (27/57)
edu.uha.miage.GeneratedSegmentTest.p000027000 checkSetterSegmentP2 (28/57)
edu.uha.miage.GeneratedSegmentTest.p000028000 testMethodSegmentLength (29/57)
edu.uha.miage.GeneratedSegmentTest.p000029000 testMethodSegmentProjectionLengthOnX (30/57)
edu.uha.miage.SegmentTest.testConstructor (31/57)
edu.uha.miage.SegmentTest.testToString (32/57)
edu.uha.miage.GeneratedTriangleTest.p000030000 checkTriangleFieldS1 (33/57)
Incorrect program result
Message : java.lang.AssertionError:
edu.uha.miage.GeneratedTriangleTest.p000030000 checkTriangleFieldS2 (34/57)
Incorrect program result
Message : java.lang.AssertionError:
edu.uha.miage.GeneratedTriangleTest.p000030000 checkTriangleFieldS3 (35/57)
Incorrect program result
Message : java.lang.AssertionError:
edu.uha.miage.GeneratedTriangleTest.p000031000 checkConstructorTrianglePointPoint (36/57)
Incorrect program result
Message : java.lang.AssertionError: Constructor of edu.uha.miage.Triangle
edu.uha.miage.GeneratedTriangleTest.p000032000 checkMethodTriangleGetP1 (37/57)
Incorrect program result
Message : java.lang.AssertionError: Fix getP1
edu.uha.miage.GeneratedTriangleTest.p000032000 checkMethodTriangleGetP2 (38/57)
Incorrect program result
Message : java.lang.AssertionError: Fix getP2
```

```
1 /*
2  * @author : Yvan Maillot (yvan.maillot@uha.fr)
3  */
4 package edu.uha.miage;
5
6 /**
7  * @author yvan
8  */
9 public class Segment {
10
11     private Point p1, p2;
12
13     public Segment(Point p1, Point p2) {
14         this.p1 = p1;
15         this.p2 = p2;
16     }
17
18     public Point getP1() {
19         return p1;
20     }
21
22     public void setP1(Point p1) {
23         this.p1 = p1;
24     }
25
26     public Point getP2() {
27         return p2;
28     }
29
30     public void setP2(Point p2) {
31         this.p2 = p2;
32     }
33
34     public double length() {
35         return Math.sqrt((p2.getX() - p1.getX()) * (p2.getX() - p1.getX()) + (p2.getY() - p1.getY())
36     }
37
38     public double projectionLengthOnX() {
39         return Math.abs(p1.getX() - p2.getX());
40     }
41
42     @Override
43     public String toString() {
44         return "[" + p1 + " ; " + p2 + " ]";
45     }
46 }
47
```

# Plusieurs Classes

## Nombreux points d'évaluation

Java L3 (UHA) Des triangles p... x +

Java L3 (UHA) Des triangles plutôt classes Éditer - Google Chrome

moodle.caseine.org/mod/vpl/forms/edit.php?id=16986&userid=10826

```
12 this.s2 = new Segment(p2, p3);
13 this.s3 = new Segment(p3, p1);
14 }
15
16 public Point getP1() {
17     return s1.getP1();
18 }
19
20 public Point getP2() {
21     return s2.getP1();
22 }
23
24 public Point getP3() {
25     return s3.getP1();
26 }
27
28 public void setP1(Point p1) {
29     this.s1 = new Segment(s1.getP1(), s2.getP1());
30     this.s3 = new Segment(s3.getP1(), p1);
31 }
32
33 public void setP2(Point p2) {
34     this.s1 = new Segment(s1.getP1(), p2);
35     this.s2 = new Segment(p2, s3.getP1());
36 }
37
38 public void setP3(Point p3) {
39     this.s2 = new Segment(s2.getP1(), p3);
40     this.s3 = new Segment(p3, s1.getP1());
41 }
42
43 public double getPerimetre() {
44     return s1.length() + s2.length() + s3.length();
45 }
46
47 public double getSurface() {
48     double a = s1.length();
49     double b = s2.length();
50     double c = s3.length();
51     double p = (a + b + c) / 2;
52     return Math.sqrt(p * (p - a) * (p - b) * (p - c));
53 }
54
55 point // @ToCompare(value="Behavior of Triangle.getBaryCentre()", priority = 39, grade = 5)
56 Point getBaryCentre() {
57     Point a = s1.getP1();
58     Point b = s2.getP1();
59     Point c = s3.getP1();
60     return new Point((a.getX() + b.getX() + c.getX()) / 3, (a.getY() + b.getY() + c.getY())
```

• Note proposée: 20 / 20

▼ Commentaires

Tests results

- edu.uha.miage.GeneratedPointTest.p00001000 checkPointFielDX (15/7)
- edu.uha.miage.GeneratedPointTest.p00001009 checkPointFielY (2/57)
- edu.uha.miage.GeneratedPointTest.p00002000 checkConstructorPointdoubledouble (3/57)
- edu.uha.miage.GeneratedPointTest.p00003000 checkMethodPointGetX (4/57)
- edu.uha.miage.GeneratedPointTest.p00003000 checkMethodPointGetY (6/57)
- edu.uha.miage.GeneratedPointTest.p00004000 checkMethodPointSetXdouble (6/57)
- edu.uha.miage.GeneratedPointTest.p00004000 checkMethodPointSetYdouble (7/57)
- edu.uha.miage.GeneratedPointTest.p00005000 checkMethodPointToString (8/57)
- edu.uha.miage.GeneratedPointTest.p00006000 checkGetterPointX (9/57)
- edu.uha.miage.GeneratedPointTest.p00006000 checkGetterPointY (10/57)
- edu.uha.miage.GeneratedPointTest.p00007000 checkSetterPointX (11/57)
- edu.uha.miage.GeneratedPointTest.p00007000 checkSetterPointY (12/57)
- edu.uha.miage.PointTest.testToString (14/57)
- edu.uha.miage.GeneratedSegmentTest.p00002000 checkSegmentFieldP1 (15/57)
- edu.uha.miage.GeneratedSegmentTest.p00002000 checkSegmentFieldP2 (16/57)
- edu.uha.miage.GeneratedSegmentTest.p000021000 checkConstructorSegmentPointPoint (17/57)
- edu.uha.miage.GeneratedSegmentTest.p000022000 checkMethodSegmentGetP1 (18/57)
- edu.uha.miage.GeneratedSegmentTest.p000022000 checkMethodSegmentGetP2 (19/57)
- edu.uha.miage.GeneratedSegmentTest.p000023000 checkMethodSegmentSetP1Point (20/57)
- edu.uha.miage.GeneratedSegmentTest.p000023000 checkMethodSegmentSetP2Point (21/57)
- edu.uha.miage.GeneratedSegmentTest.p000024000 checkMethodSegmentLength (22/57)
- edu.uha.miage.GeneratedSegmentTest.p000025000 checkMethodSegmentProjectionLengthOnX (23/57)
- edu.uha.miage.GeneratedSegmentTest.p000025000 checkMethodSegmentToString (24/57)
- edu.uha.miage.GeneratedSegmentTest.p000026000 checkGetterSegmentP1 (25/57)
- edu.uha.miage.GeneratedSegmentTest.p000026000 checkGetterSegmentP2 (26/57)
- edu.uha.miage.GeneratedSegmentTest.p000027000 checkSetterSegmentP1 (27/57)
- edu.uha.miage.GeneratedSegmentTest.p000027000 checkSetterSegmentP2 (28/57)
- edu.uha.miage.GeneratedSegmentTest.p000028000 testMethodSegmentLength (29/57)
- edu.uha.miage.GeneratedSegmentTest.p00003000 testMethodSegmentProjectionLengthOnX (30/57)
- edu.uha.miage.SegmentTest.testToString (32/57)
- edu.uha.miage.GeneratedTriangleTest.p00003000 checkTriangleFieldS1 (33/57)
- edu.uha.miage.GeneratedTriangleTest.p00003000 checkTriangleFieldS2 (34/57)
- edu.uha.miage.GeneratedTriangleTest.p00003000 checkTriangleFieldS3 (35/57)
- edu.uha.miage.GeneratedTriangleTest.p000031000 checkConstructorTrianglePointPoint (36/57)
- edu.uha.miage.GeneratedTriangleTest.p000032000 checkMethodTriangleGetP1 (37/57)
- edu.uha.miage.GeneratedTriangleTest.p000032000 checkMethodTriangleGetP2 (38/57)
- edu.uha.miage.GeneratedTriangleTest.p000032000 checkMethodTriangleGetP3 (39/57)
- edu.uha.miage.GeneratedTriangleTest.p000033000 checkMethodTriangleGetPoint (40/57)
- edu.uha.miage.GeneratedTriangleTest.p000033000 checkMethodTriangleSetP2Point (41/57)
- edu.uha.miage.GeneratedTriangleTest.p000033000 checkMethodTriangleSetP3Point (42/57)
- edu.uha.miage.GeneratedTriangleTest.p000034000 checkMethodTriangleGetPerimetre (43/57)
- edu.uha.miage.GeneratedTriangleTest.p000035000 testMethodTrianglePerimetre (44/57)
- edu.uha.miage.GeneratedTriangleTest.p000036000 checkMethodTriangleGetSurface (45/57)
- edu.uha.miage.GeneratedTriangleTest.p000037000 testMethodTriangleGetSurface (46/57)
- edu.uha.miage.GeneratedTriangleTest.p000038000 checkMethodTriangleGetBaryCentre (47/57)
- edu.uha.miage.GeneratedTriangleTest.p00004000 checkMethodTriangleToString (48/57)
- edu.uha.miage.TriangleTest.testConstructor (49/57)

# Évaluation de performance

The screenshot shows a web browser window with a Java code editor. The code is for a class named `Polynome.java` and includes methods for generating polynomials, evaluating them, and converting them to strings. The browser's address bar shows the URL `moodle.caseine.org/mod/vpl/forms/edit.php?id=17515&userid=10826`. On the right side, there is a panel for test results, showing a score of 14.74 / 20 and a list of test cases with their respective scores. The test results panel also displays error messages for some tests, such as `Message : java.lang.AssertionError: Methode evaluer() incorrecte`.

```
38     this.a = new Fraction[0 + 1];
39     for (int i = 0; i < this.a.length; ++i) {
40         this.a[i] = a[i];
41     }
42     degree = as.length - 1;
43     while (degree >= 0 && as[degree].equals(new Fraction())) c
44         a = Arrays.copyOf(as, degree + 1);
45     }
46
47     public Fraction get(int i) {
48         return a[i];
49     }
50
51     public int getDegre() {
52         return degree;
53     }
54
55     public Fraction evaluer(Fraction x) {
56         return x;
57     }
58
59     @Override
60     public String toString() {
61         if (a.length == 0) {
62             return "Polynome nul";
63         } else {
64             StringBuilder sb = new StringBuilder();
65             int n = a.length - 1;
66             sb.append(a[n]).append("x^").append(n);
67             for (int i = n - 1; i >= 0; --i) {
68                 sb.append("+").append(a[i]).append("x^").append(
69
70
```

Note proposée: 14,74 / 20

Commentaires

edu.uha.miage.fractions.FractionTest.p00000200 - Test (1/12)

edu.uha.miage.fractions.GeneratedPolynomeTest.p000 (2/12)

edu.uha.miage.fractions.GeneratedPolynomeTest.p000 (3/12)

edu.uha.miage.fractions.GeneratedPolynomeTest.p000 (4/12)

edu.uha.miage.fractions.GeneratedPolynomeTest.p000 (5/12)

edu.uha.miage.fractions.GeneratedPolynomeTest.p000 (6/12)

edu.uha.miage.fractions.PolynomeTest.testConstructe (8/12)

edu.uha.miage.fractions.PolynomeTest.testGetDegre (5 (edu.uha.miage.fractions.PolynomeTest.testEvaluer (10

Incorrect program result

Message : java.lang.AssertionError: Methode evaluer() incor

edu.uha.miage.fractions.PolynomeTest.performanceEv

Incorrect program result

Message : java.lang.AssertionError: Methode evaluer() incor

edu.uha.miage.fractions.PolynomeTest.p000000100 re (12/12)

Tests results

evaluationTime (1/1)

Message : The runtime complexity of your algorithm seems (Expected by teacher : O(n)) (Attempt of detection : O(1)) Method : evaluationTime

Exécution

# Évaluation de performance

The screenshot shows a Moodle LMS interface with a Java code editor and test results. The code editor displays the following Java code for a polynomial class:

```
38     this.a = new Fraction[0 + 1];
39     for (int i = 0; i < this.a.length; ++i) {
40         this.a[i] = a[i];
41     }
42     degree = as.length - 1;
43     while (degree >= 0 && as[degree].equals(new Fraction())) c
44         a = Arrays.copyOf(as, degree + 1);
45     }
46
47     public Fraction get(int i) {
48         return a[i];
49     }
50
51     public int getDegre() {
52         return degree;
53     }
54
55     public Fraction evaluer(Fraction x) {
56         try {
57             Fraction r = a[0];
58             Fraction xpi = new Fraction(1);
59             for (int i = 1; i < a.length; ++i) {
60                 for(int j = 0; j < i; ++j) {
61                     new Fraction(1).fois(x);
62                 }
63                 xpi = xpi.fois(x);
64                 r = r.plus(a[i].fois(xpi));
65             }
66             return r;
67         } catch (ArrayIndexOutOfBoundsException ai) {
68             throw new PolynomeNulException();
69         }
70     }
```

The right sidebar shows the test results for the code:

Note proposée: 16,84 / 20

Commentaires

- edu.uha.miage.fractions.GeneratedPolynomeTest.p000000100 (2/12)
- edu.uha.miage.fractions.GeneratedPolynomeTest.p000000101 (3/12)
- edu.uha.miage.fractions.GeneratedPolynomeTest.p000000102 (4/12)
- edu.uha.miage.fractions.GeneratedPolynomeTest.p000000103 (5/12)
- edu.uha.miage.fractions.GeneratedPolynomeTest.p000000104 (6/12)
- edu.uha.miage.fractions.PolynomeTest.testConstructeur (8/12)
- edu.uha.miage.fractions.PolynomeTest.testGetDegre (9/12)
- edu.uha.miage.fractions.PolynomeTest.testEvaluer (10/12)

Incorrect program result  
Message : The time limit set to 5000 milliseconds has been exceeded while executing this test. You might have an infinite loop or a very long code.

edu.uha.miage.fractions.PolynomeTest.p000000100\_re (12/12)

Tests results

evaluationTime (1/1)

Message : We are unable to guess the runtime complexity of this code accurately. It seems worse than the expected complexity but we will check with a teacher.  
(Attempt of detection : n^(2.32293))  
Method : evaluationTime

Exécution

# Évaluation de performance

The screenshot shows a web-based IDE interface. The browser address bar indicates the URL: `moodle.caseine.org/mod/vpl/forms/edit.php?id=17515&userid=10826`. The IDE has a file explorer on the left showing a project structure with folders like 'uhua', 'miage', 'fractions', and 'exceptions', and files like 'Polynome.java', 'Fraction.java', 'PasUneEquationDu...', 'InversionFrac...', 'DenominateurN...', 'PolynomeNulEx...', 'SituationImposs...', 'DivisionParUneF...', 'main', and 'Main.java'. The main editor displays the code for 'Polynome.java' with line numbers 39 to 70. The code includes a constructor for 'a', methods for 'get(int i)', 'getDegre()', and 'evaluer(Fraction x)', and an overridden 'toString()' method. The 'evaluer' method uses a recursive approach to evaluate a polynomial. On the right side, there is a 'Note proposée: 20 / 20' and a 'Commentaires' section. The comments section shows 'Tests results' for various test cases, all with a score of 1/1. A message states: 'Message : The runtime complexity of your algorithm seems (Expected by teacher : O(n)) (Attempt of detection : O(n) or O(nlog(n))) Method : evaluationTime'.

```
39 -     for (int i = 0; i < this.a.length; ++i) {
40 -         this.a[i] = a[i];
41 -     }
42 -     /*
43 -     degre = as.length - 1;
44 -     while (degre >= 0 && as[degre].equals(new Fraction())) {
45 -         a = Arrays.copyOf(as, degre + 1);
46 -     }
47 -     public Fraction get(int i) {
48 -         return a[i];
49 -     }
50 -     public int getDegre() {
51 -         return degre;
52 -     }
53 -     public Fraction evaluer(Fraction x) {
54 -         try {
55 -             Fraction r = a[0];
56 -             Fraction xpi = new Fraction(1);
57 -             for (int i = 1; i < a.length; ++i) {
58 -                 xpi = xpi.fois(x);
59 -                 r = r.plus(a[i].fois(xpi));
60 -             }
61 -             return r;
62 -         } catch (ArrayIndexOutOfBoundsException ai) {
63 -             throw new PolynomeNulException();
64 -         }
65 -     }
66 -     @Override
67 -     public String toString() {
```

# Du graphisme

```
1 package edu.uha.miage.frames;  
2  
3 import java.awt.BorderLayout;  
4  
5  
6  
7  
8  
9  
10  
11 /* TODO 1.01 La classe PointsCardinaux est une JFrame qui affiche -  
12    au nord, au sud, à l'est à l'ouest et un label (JLabel) au  
13  
14    L'appui sur le bouton au nord fait s'afficher "Nord" au ce  
15    L'appui sur le bouton au sud fait s'afficher "Sud" au cent  
16    L'appui sur le bouton à l'est fait s'afficher "Est" au cen  
17    L'appui sur le bouton à l'ouest fait s'afficher "Ouest" au  
18  
19    Cette classe do  
20    */  
21 //  
22
```

Démonstration de l'exo "Les points cardinaux"

Nord

Ouest Centre Est

Items	Description
2 items	
!	TODO 1.01 Ajouter au bouton lance une action
	TODO 1.01 La classe PointsCardinaux est une

Path
/Bandit manchot/src/edu/uha
/Points Cardinaux/src/edu/uha

# Du graphisme

The screenshot shows the Eclipse IDE interface. The top menu bar includes File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, and Help. The toolbar contains various icons for file operations and development tools. The Package Explorer on the left shows the project structure for 'Correction Taquin', including 'src' and 'Taquin.java'. The main editor window displays the code for 'Taquin.java', which starts with the package declaration: `package edu.uha.miage.taquin;`. A window titled 'Jeu du taquin' is open, showing a colorful abstract painting with geometric shapes and lines. The painting features a grid, various colored circles, triangles, and lines, creating a complex and dynamic composition. The right-hand side of the IDE shows the 'Correction Taquin' window with a 'Descriptor' field and a 'Reload description' button. The bottom status bar indicates the current path: 'Bandit manchot/src/edu/u'.



# Du graphisme

test-caseine - Correction Taquin/src/edu/uha/miage/taquin/Taquin.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Ex

Correction Taquin

- JRE System Library [J]
- src
  - edu.uha.miage.di
  - edu.uha.miage.e
  - edu.uha.miage.m
  - Main.java**
  - ManipulateurT
  - Kandinski.jpg
  - kandinski2.jpg
  - edu.uha.miage.pi
  - edu.uha.miage.ta
    - Taquin.java
      - Taquin
        - MAXIML
        - MAXIMU
        - MINIMU
        - MINIMU
        - image
        - melange
        - piece
        - trou
        - Taquin(ir
        - cliq(Piece
        - deplacer

```

1 package edu.uha.miage.taquin;
2
3
14
15 Action
16 Melanger
17 Dimensions
18 Choix de l'image
19 Enlever l'image
20
21
22
23
24
25
26
27
28

```

Jeu du taquin

Correction Taquin

Descriptor

Reload description

ath

bandit manchot/src/edu/u

# Du graphisme

The screenshot shows the Eclipse IDE interface. The main editor displays the following code in `Taquin.java`:

```
1 package edu.uha.miage.taquin;  
2  
3  
14  
15 Action  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28
```

A window titled "Jeu du taquin" is overlaid on the code, showing a 4x4 grid. The grid contains various geometric shapes and colors, including circles, squares, and lines, representing a game board. The shapes are arranged in a way that suggests a puzzle or game state.

The left sidebar shows the project structure for "Correction Taquin", including the "src" directory and the "Main.java" file. The right sidebar shows the "Correction Taquin" descriptor.

# Du graphisme

test-caseine - Correction Taquin/src/edu/uha/miage/taquin/Taquin.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Quick Access

Package Ex

Bandit.java Taquin.java

```

1 package edu.uha.miage.taquin;
2
3
14
15 Action
16
17
18
19
20
21
22
23
24
25
26
27
28

```

Correction Taquin

Descriptor

Reload description

ath

Bandit manchot/src/edu/u

The screenshot shows the Eclipse IDE interface. The main editor window displays the source code for 'Taquin.java', which starts with a package declaration: 'package edu.uha.miage.taquin;'. A window titled 'Jeu du taquin' is overlaid on the code, showing a 4x4 grid of abstract shapes and colors, characteristic of Piet Mondrian's 'De Witte Doek' painting. The IDE's left sidebar shows a project structure with 'Main.java' selected. The right sidebar shows a 'Correction Taquin' panel with a 'Descriptor' field and a 'Reload description' button. The bottom status bar shows the file path 'Bandit manchot/src/edu/u'.

# Du graphisme



```

156 }
157
158 pub
159
160
161
162
163 }
164 }
165


```

```

// possible de le relancer.
}
};

private void addActionListenerToLance() {
    TODO 1.01 Ajouter au bouton lance une actionListene
    trois roues se mettent à tourner indépendemment
}

```



```

(À FAIRE)");
    demo = new

```

# Circuits logiques (avec Logisim, à l'UGA) – Denis Bouhineau

Cours : Circuits logiques | +

https://moodle.caseine.org/course/view.php?id=642

120%

Caseine Français (fr)

Yvan Maillot

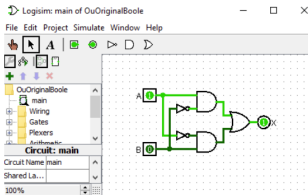
## Circuits logiques (avec Logisim, à l'UGA)

Accueil / Mes cours / Circuits

### Circuits logiques

Dessins, exécutions et validations de circuits logiques avec Logisim et Vpl.

Avec l'éditeur et simulateur de circuit "Logisim" (<http://www.cburch.com/logisim/>)



Contact : Denis Bouhineau ([denis.bouhineau@imag.fr](mailto:denis.bouhineau@imag.fr))

Cours associé : <https://im2ag-moodle.univ-grenoble-alpes.fr/course/view.php?id=14>

Votre progression ?

Circuits essentiels



0%

Voir mon progrès

Pour être à l'aise, une bouteille bien remplie ?



# Circuits logiques (avec Logisim, à l'UGA) – Denis Bouhineau

Cours : Circuits logiques | +

https://moodle.caseine.org/course/view.php?id=642

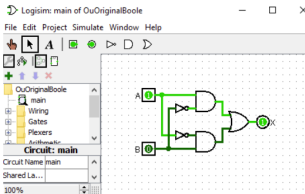
120%

Caseine Français (fr) | Yvan Maillot

## Circuits logiques

Dessins, exécutions et validations de circuits logiques avec Logisim et Vpl.

Avec l'éditeur et simulateur de circuit "Logisim" (<http://www.cburch.com/logisim/>)



Votre progression ?

### Circuits essentiels



0%

Voir mon progrès

Pour être à l'aise, une bouteille bien remplie ?



2%

Voir mon progrès

Contact : Denis Bouhineau ([denis.bouhineau@imag.fr](mailto:denis.bouhineau@imag.fr))

Cours associé : <https://im2ag-moodle.univ-grenoble-alpes.fr/course/view.php?id=14>

Important (Mode d'emploi)

# Circuits logiques (avec Logisim, à l'UGA) – Denis Bouhineau

Cours : Circuits logiques | +

https://moodle.caseine.org/course/view.php?id=642

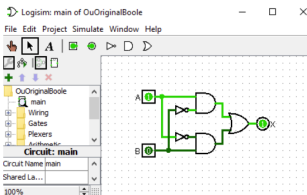
120%

Caseine Français (fr)

Yvan Maillot

Dessins, exécutions et validations de circuits logiques avec Logisim et Vpl.

Avec l'éditeur et simulateur de circuit "Logisim" (<http://www.cburch.com/logisim/>)



Contact : Denis Bouhineau ([denis.bouhineau@imag.fr](mailto:denis.bouhineau@imag.fr))

Cours associé : <https://im2ag-moodle.univ-grenoble-alpes.fr/course/view.php?id=14>

## Important (Mode d'emploi)

### Mode d'emploi\_LogisimVplVersion 1.6.5

Dessins, exécutions et validations de circuits logiques avec Logisim et Vpl (intégrés).

- choisir une exercice dans le cours, prendre connaissance de l'énoncé de l'exercice
- lancer l'édition de l'exercice

0%

Voir mon progrès

Pour être à l'aise, une bouteille bien remplie ?

3%

Voir mon progrès

La totale

# Circuits logiques (avec Logisim, à l'UGA) – Denis Bouhineau

Cours : Circuits logiques | x

https://moodle.caseine.org/course/view.php?id=642

120%

Caseine Français (fr)

Logisim: main of OuOriginalBoole

File Edit Project Simulate Window Help

OuOriginalBoole

- main
- Wiring
- Gates
- Plexers

Circuit: main

Circuit Name main

Shared La...

100%

0%

Voir mon progrès

Contact : Denis Bouhineau (denis.bouhineau@imag.fr)

Cours associé : <https://im2ag-moodle.univ-grenoble-alpes.fr/course/view.php?id=14>

## Important (Mode d'emploi)

### Mode d'emploi, LogisimVpl Version 1.6.5

Dessins, exécutions et validations de circuits logiques avec Logisim et Vpl (intégrés).

- choisir une exercice dans le cours, prendre connaissance de l'énoncé de l'exercice
- lancer l'édition de l'exercice
- [la première fois], sauvegarder les fichiers (icone "disquette")
- visualisation et édition du circuit : lancer le "debug" (icone "coccinelle")
  - sous Logisim, "éditer" le circuit (icone "flèche"), ne pas modifier les éléments fournis dans le circuit initial
  - [facultatif] sous Logisim, "simuler" le circuit (icone "Index"), en cliquant sur les plots d'entrée
  - à la fin, important, sous Logisim, "sauvegarder" son circuit (**menu File/Save**)
  - retour à l'exercice (fermeture de la fenêtre modale) et acceptation des nouveaux fichiers

Pour être à l'aise, une bouteille bien remplie ?



2%

Voir mon progrès

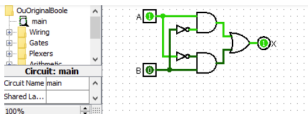
La totale



2%



# Circuits logiques (avec Logisim, à l'UGA) – Denis Bouhineau



Contact : Denis Bouhineau (denis.bouhineau@imag.fr)

Cours associé : <https://im2ag-moodle.univ-grenoble-alpes.fr/course/view.php?id=14>

## Important (Mode d'emploi)

### Mode d'emploi LogisimVpl Version 1.6.5

Dessins, exécutions et validations de circuits logiques avec Logisim et Vpl (intégrés).

- choisir une exercice dans le cours, prendre connaissance de l'énoncé de l'exercice
- lancer l'édition de l'exercice
- [la première fois], sauvegarder les fichiers (icone "disquette")
- visualisation et édition du circuit : lancer le "debug" (icone "coccinelle")
  - sous Logisim, "éditer" le circuit (icone "flèche"), ne pas modifier les éléments fournis dans le circuit initial
  - [facultatif] sous Logisim, "simuler" le circuit (icone "index"), en cliquant sur les plots d'entrée
  - à la fin, important, sous Logisim, "sauvegarder" son circuit (**menu File/Save**)
  - retour à l'exercice (fermeture de la fenêtre modale), et acceptation des nouveaux fichiers
- [facultatif] exécution textuelle du circuit : lancer "run" (icone avec "fusée")
- test : lancer la "validation" du circuit avec les tests automatiques (icone "boîte à cocher")

## Forum de discussion

# Circuits logiques (avec Logisim, à l'UGA) – Denis Bouhineau

The screenshot shows a Moodle course page for 'Circuits logiques' on the Caseine platform. The browser address bar shows the URL: <https://moodle.caseine.org/course/view.php?id=642>. The page header includes the course name 'Caseine' and the language 'Français (fr)'. The contact information for Denis Bouhineau is provided, along with the course URL: <https://im2ag-moodle.univ-grenoble-alpes.fr/course/view.php?id=14>.

The main content area features a section titled 'Important (Mode d'emploi)' with a document icon. Below this, the text reads 'Mode d'emploi LogisimVpl Version 1.6.5' and describes the use of Logisim and Vpl for logic circuit design. A list of instructions follows:

- choisir une exercice dans le cours, prendre connaissance de l'énoncé de l'exercice
- lancer l'édition de l'exercice
- [la première fois], sauvegarder les fichiers (icone "disquette")
- visualisation et édition du circuit : lancer le "debug" (**icone "coccinelle"**)
  - sous Logisim, "éditer" le circuit (icone "flèche"), ne pas modifier les éléments fournis dans le circuit initial
  - [facultatif] sous Logisim, "simuler" le circuit (icone "index"), en cliquant sur les plots d'entrée
  - à la fin, important, sous Logisim, "sauvegarder" son circuit (**menu File/Save**)
  - retour à l'exercice (fermeture de la fenêtre modale), et acceptation des nouveaux fichiers
- [facultatif] exécution textuelle du circuit : lancer "run" (icone avec "fusée")
- test : lancer la "validation" du circuit avec les tests automatiques (**icone "boîte à cocher"**)

Below the instructions, there are links for 'Forum de discussion', 'Exercice de prise en main (logisimVpl V1.6.5)', and 'Vidéo de prise en main'. The 'Exercice de prise en main' link has a notification icon with the number '5'. The 'Vidéo de prise en main' link is accompanied by a video player thumbnail showing a circuit simulation interface.

On the right side of the page, there are two progress bars, each showing 2% completion and a 'Voir mon progrès' button. The top progress bar is associated with 'Yvan Maillot' and the bottom one with 'La totale', which has a milk carton icon.

# Circuits logiques (avec Logisim, à l'UGA) – Denis Bouhineau

Cours : Circuits logiques | +

https://moodle.caseine.org/course/view.php?id=642

120%

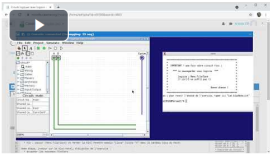
Caseine Français (fr) | Yvan Maillot

Forum de discussion

Exercice de prise en main (logisimVpl V1.6.5) \*\*

Vidéo de prise en main

Vidéo de prise en main :



Announcements

Prérequis

Un peu de binaire et de logique : <https://moodle.caseine.org/course/view.php?id=96>

Un peu de langage machine : <https://moodle.caseine.org/course/view.php?id=716>

Test de positionnement en Circuits Logiques

Petits circuits

Petits circuits à réaliser avec des portes "et", "ou", "non" seulement.

# Circuits logiques (avec Logisim, à l'UGA) – Denis Bouhineau

Cours : Circuits logiques | +

https://moodle.caseine.org/course/view.php?id=642

120%

Caseine Français (fr) | Yvan Maillot

## Petits circuits

Petits circuits à réaliser avec des portes "et", "ou", "non" seulement.

Exemples (exercices corrigés)

- Vidéo sur un compteur de Gray
- Vidéo sur la multiplication

Exercices

- Circuit à partir d'une formule logique sans forme particulière (logisimVpl V1.6.5) \*\*
- Circuit à partir d'une formule disjonctive (logisimVpl V1.6.5)
- Circuit à partir d'une formule conjonctive (logisimVpl V1.6.5)
- Circuit à partir d'une formule algébrique (logisimVpl V1.6.5)
- Circuit à partir d'une table de vérité (logisimVpl V1.6.5) \*\*
- Circuit à partir d'un tableau de Karnaugh (logisimVpl V1.6.5)
- Circuit à partir d'un arbre de Shannon (logisimVpl V1.6.5)
- Un et un seul vrai (logisimVpl V1.6.5)
- Majorité (logisimVpl V1.6.5) \*
- Dé 6 faces (logisimVpl V1.6.5) \*

Versez de plus gros circuits

# Circuits logiques (avec Logisim, à l'UGA) – Denis Bouhineau

Circuits logiques (avec Logisim, à l'UGA) | Caseine | Français (fr) | Yvan Maillot

## Circuits logiques (avec Logisim, à l'UGA)

[Accueil](#) / [Mes cours](#) / [Circuits](#) / [Petits circuits](#) / [Majorité \(logisimVpl V1.6.5\)](#) / [Description](#)

[Description](#) [Remise](#) [Modifier](#) [Vue de dépôt](#) [Liste des soumissions précédentes](#)

### Majorité (logisimVpl V1.6.5)

**Required Files:** blocNote.txt, circuit.circ ([Télécharger](#))

**Nombre maximal de fichiers:** 10

**Type de travail:** Travail individuel

**Exercice :** E0, E1, E2 => S

La sortie (S) est donnée par la valeur la plus présente parmi E0, E1, E2, la valeur majoritaire.

Exemple, pour E0=0, E1=1 et E2=0, on veut S=0.

Mode d'emploi succinct :

[la première fois, sauvegarder (icone "disquette") pour commencer]  
en général :

- visualisation et édition du circuit : "debug" (icone "coccinelle")
- [facultatif] exécution du circuit : "run" (icone avec "fusée")
- test : validation du circuit (icone "boite à cocher")

### Résumé de l'état du travail

**Remises:** 7

**Dernière soumission:**

Déposé le jeudi 11 mai 2023, 18:24 ([Télécharger](#))

Révisé le jeudi 11 mai 2023, 18:24 par Note automatique

**Note:** 100,00 / 100,00

[Détails](#)

# Circuits logiques (avec Logisim, à l'UGA) – Denis Bouhineau

## Un VPL peut proposer des activités

- d'initiation ou avancées
- graphiques
- de tests unitaires
- ...
- dans de nombreux langages (C, C++, Python, Java, OCaml, Prolog, Lisp, Bash, R, SQL...)

## Un VPL peut mesurer

- la fonctionnalité du code (avec +/- de souplesse)
- sa qualité (linter, parser...)
- son efficacité (estimation de complexité, mesure du temps)

# Travailler dans un IDE

Existence de plugins Caseine pour

- Eclipse



# Travailler dans un IDE

## Existence de plugins Caseine pour

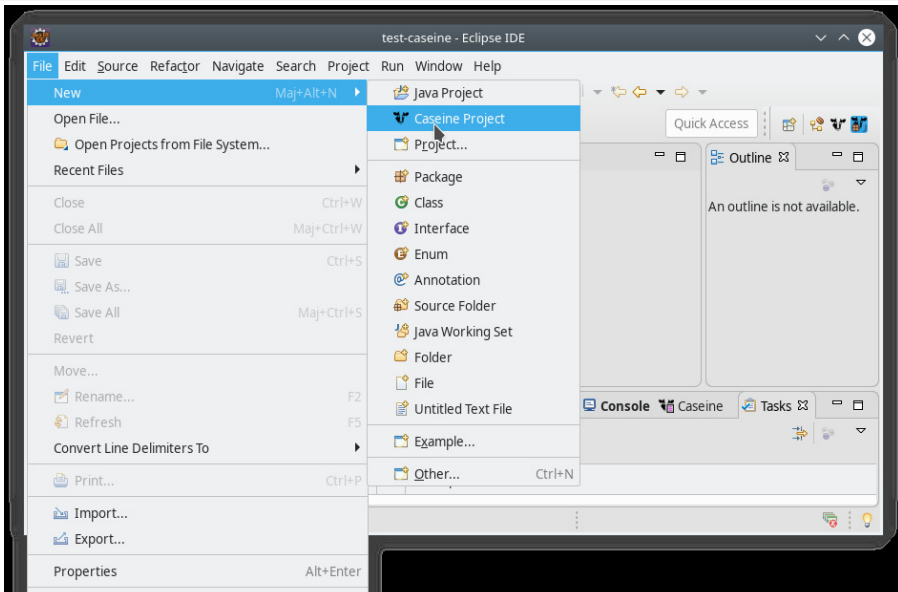
- Eclipse
- JetBrains

# Travailler dans un IDE

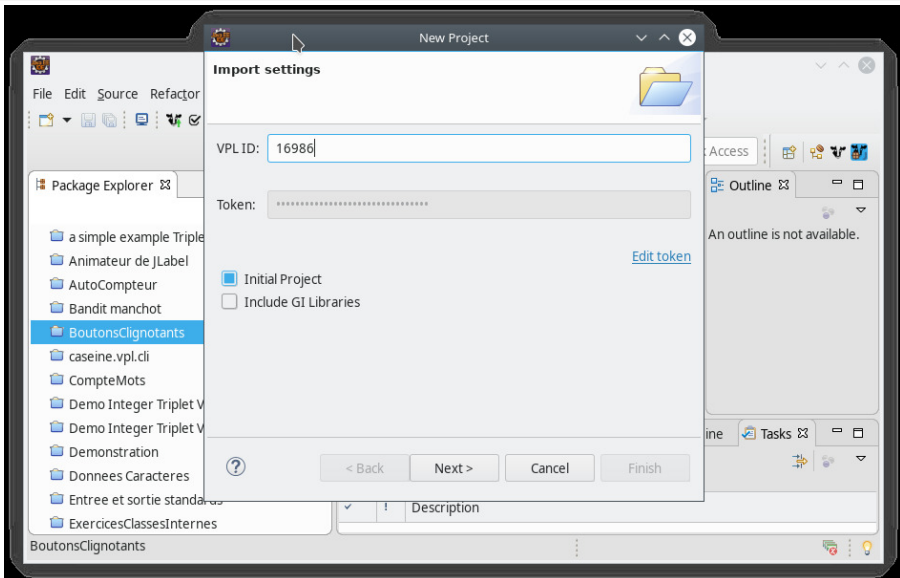
## Existence de plugins Caseine pour

- Eclipse
- JetBrains
- VS Code

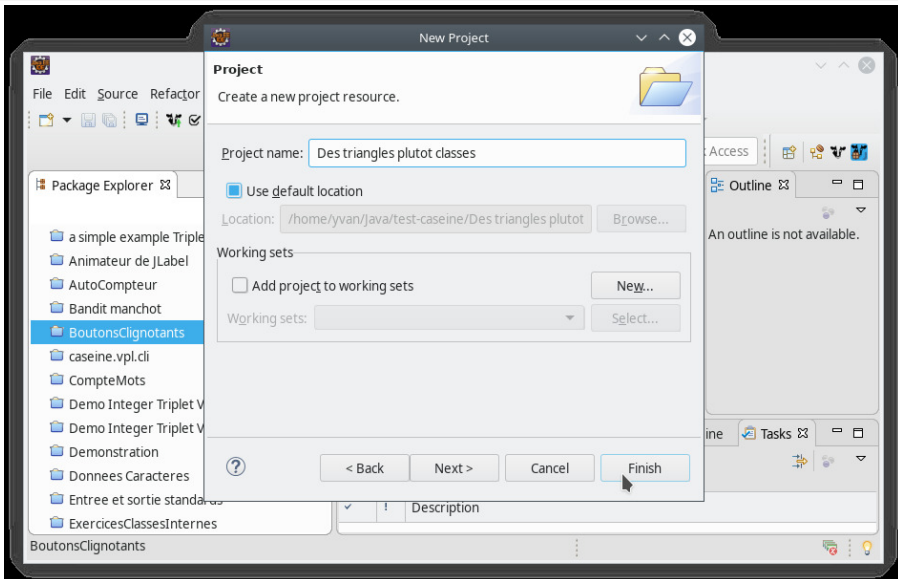
# Le plugin Eclipse



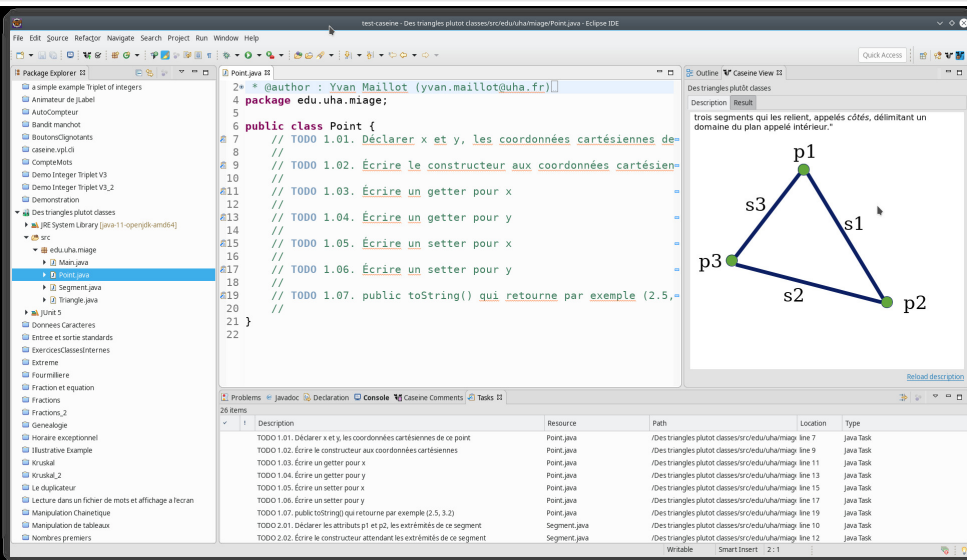
# Le plugin Eclipse



# Le plugin Eclipse



# Le plugin Eclipse



The screenshot shows the Eclipse IDE interface. The main editor displays the code for the `Point` class in `Point.java`. The code includes package declarations, class declarations, and several TODO comments for implementing methods like `declarer`, `ecrire`, `getter`, `setter`, and `toString`.

On the right side, the Outline view shows the project structure, and the Case View shows a diagram of a triangle with vertices labeled `p1`, `p2`, and `p3`, and sides labeled `s1`, `s2`, and `s3`. The diagram is a blue triangle with green dots at the vertices.

At the bottom, the Problems view shows a list of 26 items, including tasks for implementing the methods defined in the code.

Description	Resource	Path	Location	Type
TODO 1.01. Déclarer x et y, les coordonnées cartésiennes de ce point	Point.java	/Des triangles plutot classes/src/edu/uha/miage	line 7	Java Task
TODO 1.02. Écrire le constructeur aux coordonnées cartésiennes	Point.java	/Des triangles plutot classes/src/edu/uha/miage	line 9	Java Task
TODO 1.03. Écrire un getter pour x	Point.java	/Des triangles plutot classes/src/edu/uha/miage	line 11	Java Task
TODO 1.04. Écrire un getter pour y	Point.java	/Des triangles plutot classes/src/edu/uha/miage	line 13	Java Task
TODO 1.05. Écrire un setter pour x	Point.java	/Des triangles plutot classes/src/edu/uha/miage	line 15	Java Task
TODO 1.06. Écrire un setter pour y	Point.java	/Des triangles plutot classes/src/edu/uha/miage	line 17	Java Task
TODO 1.07. public toString() qui retourne par exemple (2.5, 3.2)	Point.java	/Des triangles plutot classes/src/edu/uha/miage	line 19	Java Task
TODO 2.01. Déclarer les attributs p1 et p2, les extrémités de ce segment	Segment.java	/Des triangles plutot classes/src/edu/uha/miage	line 10	Java Task
TODO 2.02. Écrire le constructeur attendant les extrémités de ce segment	Segment.java	/Des triangles plutot classes/src/edu/uha/miage	line 12	Java Task

# Le plugin Eclipse

The screenshot shows the Eclipse IDE interface. The main editor displays the code for `Point.java` with several TODO comments. The right-hand side shows the 'Caseine View' with a table of test results. The bottom panel shows a list of tasks derived from the TODO comments.

```
2* * @author : Yvan Maillot (yvan.maillot@uha.fr)
4 package edu.uha.miage;
5
6 public class Point {
7     // TODO 1.01. Déclarer x et y, les coord=
8     //
9     // TODO 1.02. Écrire le constructeur aux=
10    //
11    // TODO 1.03. Écrire un getter pour x
12    //
13    // TODO 1.04. Écrire un getter pour y
14    //
15    // TODO 1.05. Écrire un setter pour x
16    //
17    // TODO 1.06. Écrire un setter pour y
18    //
19    // TODO 1.07. public toString() qui reto=
20    //
21 }
22
```

**Caseine View**

Description	Result
57 Tests	Errors : 0
	Failures : 57
<b>Proposed grade: 0 / 20</b>	
All tests [18/11/2020 18:05]	
edu.uha.miage.GeneratedPointTest	
p000001000_checkPointFieldX	
p000001000_checkPointFieldY	
p000002000_checkConstructorPointdoubledouble	
p000003000_checkMethodPointGetX	
p000003000_checkMethodPointGetY	
p000004000_checkMethodPointSetXdouble	
p000004000_checkMethodPointSetYdouble	
p000005000_checkMethodPointToString	
p000006000_checkGetterPointX	
p000006000_checkGetterPointY	

**Tasks**

Description	Resource	Path	Location	Type
TODO 1.01. Déclarer x et y, les coordonnées cartésiennes de ce point	Point.java	/Des triangles plutot classes/src/edu/uha/miage	line 7	Java Task
TODO 1.02. Écrire le constructeur aux coordonnées cartésiennes	Point.java	/Des triangles plutot classes/src/edu/uha/miage	line 9	Java Task
TODO 1.03. Écrire un getter pour x	Point.java	/Des triangles plutot classes/src/edu/uha/miage	line 11	Java Task
TODO 1.04. Écrire un getter pour y	Point.java	/Des triangles plutot classes/src/edu/uha/miage	line 13	Java Task
TODO 1.05. Écrire un setter pour x	Point.java	/Des triangles plutot classes/src/edu/uha/miage	line 15	Java Task
TODO 1.06. Écrire un setter pour y	Point.java	/Des triangles plutot classes/src/edu/uha/miage	line 17	Java Task
TODO 1.07. public toString() qui retourne par exemple (2.5, 3.2)	Point.java	/Des triangles plutot classes/src/edu/uha/miage	line 19	Java Task
TODO 2.01. Déclarer les attributs p1 et p2, les extrémités de ce segment	Segment.java	/Des triangles plutot classes/src/edu/uha/miage	line 10	Java Task

# Le plugin Eclipse

The screenshot shows the Eclipse IDE interface. The main editor displays a Java file named `*Point.java` with the following code:

```
2 * @author : Yvan Maillot (yvan.maillot@uha.fr)
4 package edu.uha.miage;
5
6 public class Point {
7     // TODO 1.01. Déclarer x et y, les coord
8     private double x, y;
9     // TODO 1.02. Écrire le constructeur aux
10    //
11    // TODO 1.03. Écrire un getter pour x
12    //
13    // TODO 1.04. Écrire un getter pour y
14    //
15    // TODO 1.05. Écrire un setter pour x
16    //
17    // TODO 1.06. Écrire un setter pour y
18    //
19    // TODO 1.07. public toString() qui reto
20    //
21 }
22
```

The Caseine view on the right shows the following summary:

- 57 Tests
- Errors: 0
- Failures: 57
- Proposed grade: 0 / 20

The Caseine view also lists all tests, including:

- edu.uha.miage.GeneratedPointTest
  - p000001000\_checkPointFieldX
  - p000001000\_checkPointFieldY
  - p000002000\_checkConstructorPointdoubledouble
  - p000003000\_checkMethodPointGetX
  - p000003000\_checkMethodPointGetY
  - p000004000\_checkMethodPointsetXdouble
  - p000004000\_checkMethodPointsetYdouble
  - p000005000\_checkMethodPointtoString
  - p000006000\_checkGetterPointX
  - p000006000\_checkGetterPointY

The Problems view at the bottom shows 26 items:

Description	Resource	Path	Location	Type
TODO 1.01. Déclarer x et y, les coordonnées cartésiennes de ce point	Point.java	/Des triangles plutot classes/src/edu/uha/miage	line 7	Java Task
TODO 1.02. Écrire le constructeur aux coordonnées cartésiennes	Point.java	/Des triangles plutot classes/src/edu/uha/miage	line 9	Java Task
TODO 1.03. Écrire un getter pour x	Point.java	/Des triangles plutot classes/src/edu/uha/miage	line 11	Java Task
TODO 1.04. Écrire un getter pour y	Point.java	/Des triangles plutot classes/src/edu/uha/miage	line 13	Java Task
TODO 1.05. Écrire un setter pour x	Point.java	/Des triangles plutot classes/src/edu/uha/miage	line 15	Java Task
TODO 1.06. Écrire un setter pour y	Point.java	/Des triangles plutot classes/src/edu/uha/miage	line 17	Java Task
TODO 1.07. public toString() qui retourne par exemple (2.5, 3.2)	Point.java	/Des triangles plutot classes/src/edu/uha/miage	line 19	Java Task
TODO 2.01. Déclarer les attributs p1 et p2, les extrémités de ce segment	Segment.java	/Des triangles plutot classes/src/edu/uha/miage	line 10	Java Task



# Le plugin Eclipse

The screenshot shows the Eclipse IDE interface. The main editor displays the code for `Point.java` with annotations for a test suite. The left sidebar shows the project structure, and the right sidebar shows the Outline and Case View tabs. The bottom panel displays a table of 26 items, which are tasks related to the code.

```
2* * @author : Yvan Maillot (yvan.maillot@uha.fr)
4 package edu.uha.miage;
5
6 public class Point {
7     // TODO 1.01. Déclarer x et y, les coord=
8     private double x, y;
9     // TODO 1.02. Écrire le constructeur aux=
10    //
11    // TODO 1.03. Écrire un getter pour x
12    //
13    // TODO 1.04. Écrire un getter pour y
14    //
15    // TODO 1.05. Écrire un setter pour x
16    //
17    // TODO 1.06. Écrire un setter pour y
18    //
19    // TODO 1.07. public toString() qui reto=
20    //
21 }
22
```

Outline Case View

Des triangles plutôt classes

Description	Result
57 Tests	Errors : 0
	Failures : 55

Proposed grade: 0.32 / 20

All tests [18/11/2020 18:08]

- edu.uha.miage.GeneratedPointTest
  - p000001000\_checkPointFieldX
  - p000001000\_checkPointFieldY
  - p000002000\_checkConstructorPointdouble
  - p000003000\_checkMethodPointGetX
  - p000003000\_checkMethodPointGetY
  - p000004000\_checkMethodPointSetXdouble
  - p000004000\_checkMethodPointSetYdouble
  - p000005000\_checkMethodPointToString
  - p000006000\_checkGetterPointX
  - p000006000\_checkGetterPointY

26 items

Description	Resource	Path	Location	Type
TODO 1.01. Déclarer x et y, les coordonnées cartésiennes de ce point	Point.java	/Des triangles plutot classes/src/edu/uha/miage/	line 7	Java Task
TODO 1.02. écrire le constructeur aux coordonnées cartésiennes	Point.java	/Des triangles plutot classes/src/edu/uha/miage/	line 9	Java Task
TODO 1.03. Écrire un getter pour x	Point.java	/Des triangles plutot classes/src/edu/uha/miage/	line 11	Java Task
TODO 1.04. Écrire un getter pour y	Point.java	/Des triangles plutot classes/src/edu/uha/miage/	line 13	Java Task
TODO 1.05. Écrire un setter pour x	Point.java	/Des triangles plutot classes/src/edu/uha/miage/	line 15	Java Task
TODO 1.06. Écrire un setter pour y	Point.java	/Des triangles plutot classes/src/edu/uha/miage/	line 17	Java Task
TODO 1.07. public toString() qui retourne par exemple (2.5, 3.2)	Point.java	/Des triangles plutot classes/src/edu/uha/miage/	line 19	Java Task
TODO 2.01. Déclarer les attributs p1 et p2, les extrémités de ce segment	Segment.java	/Des triangles plutot classes/src/edu/uha/miage/	line 10	Java Task

## 2. Suivi pédagogique des étudiants

## 2. Suivi pédagogique des étudiants

- Moodle => Carnet de notes et autres outils de suivi

## 2. Suivi pédagogique des étudiants

- Moodle => Carnet de notes et autres outils de suivi
- Suivi grâce à l'activité VPL

## 2. Suivi pédagogique des étudiants

- Moodle => Carnet de notes et autres outils de suivi
- Suivi grâce à l'activité VPL

Description   Liste des devoirs rendus   Similarité   Activité test

Groupes séparés Gr1

Sélection de la soumission Tous les envois

Évaluer Choisir...

	Prénom ↑ / Nom ↓	Soumis le ↓	Devoirs rendus ↓	Note ↓	Évaluateur ↓	Noté le ↓
7	[redacted]	dimanche 8 mars 2020, 14:45	2	2,21 / 20,00 <b>Caché</b>	Note automatique	dimanche 8 mars 2020, 14:45
8	[redacted]	mercredi 19 février 2020, 16:33	8	7,05 / 20,00 <b>Caché</b>	Note automatique	mercredi 19 février 2020, 16:33
14	[redacted]	lundi 9 mars 2020, 00:12	14	8,63 / 20,00 <b>Caché</b>	Note automatique	lundi 9 mars 2020, 00:12
27	[redacted]	mercredi 19 février 2020, 12:40	27	11,53 / 20,00 <b>Caché</b>	Note automatique	mercredi 19 février 2020, 12:40
27	[redacted]	mercredi 4 mars 2020, 18:44	27	14,00 / 20,00 <b>Caché</b>	Note automatique	mercredi 4 mars 2020, 18:44
36	[redacted]	dimanche 8 mars 2020, 13:09	36	15,00 / 20,00 <b>Caché</b>	Note automatique	dimanche 8 mars 2020, 13:09
49	[redacted]	samedi 7 mars 2020, 13:33	49	16,11 / 20,00 <b>Caché</b>	Note automatique	samedi 7 mars 2020, 13:33

### 3. Partage

- Mattermost (un espace de discussion)
- Espace de partage (panier)
- Méta-données (système de mesure)
- Cours partagés
- Événements fréquents :
  - demi-heures Caseine
  - demi-journées Caseine
- Communauté d'enseignants très active, plateforme ouverte ⇒ évolution au profit de tous
  - développement d'activités Moodle
  - de plugins,
  - de scripts
  - ...




### 3. Partage : mattermost



# 3. Partage : profiter des ressources partagées par les collègues

- UHA
- Java-M2-UHA
- Intro-réseaux-L2-Info-UHA
- InformatiqueLycee
- UML open
- Java-M1-UHA
- Plus...
- Espace de partage
- Ajouter un bloc

## + Classes Mutables vs Classes Immuables ✎

- +  Cours à lire avant de cocher (Classes Mutables vs Classes Immuables)
  -  [03-classes-immuables-fascicule-paysage.pdf](#)
  -  [03-classes-immuables-fascicule-portrait.pdf](#)

Télécharger le dossier

- +  [Caractères UNICODE](#) ✎
- +  [Equation](#) ✎
- +  [Points MUTABLES et polygones MUTABLES](#) ✎
- +  Observations à propos de l'exercice Points MUTABLES et Polygones MUTABLES  
**Accès restreint** Non disponible à moins que : L'activité [Points MUTABLES et polygones MUTABLES](#) soit terminée
- +  [Points IMMUABLES et polygones MUTABLES](#) ✎  
**Accès restreint** Non disponible à moins que l'une des conditions suivantes soit remplie



# 3. Partage : profiter des ressources partagées par les collègues

The screenshot shows a web browser window with the URL `moodle.caseine.org/local/sharedspace/h/index.php?advsearch=1`. The page title is "Shared Caseine Space - Google Chrome". The user is logged in as "Yvan Maillot". The interface displays several search filters:

- Author(s):** A search box with the text "Rechercher" and a dropdown arrow. Below it, the text "Aucune sélection" is displayed.
- Programming Language:** A search box with the text "Rechercher" and a dropdown arrow. Below it, a blue tag with the text "x Java" is visible.
- Pedagogical objectives:** A search box with the text "Rechercher" and a dropdown arrow. Below it, a list of options is shown: Python, OPL, ProLog, R, gnu-r, Excel LP solver, C, Java8, -, and Erlang.
- Academic Level:** A search box with the text "Rechercher" and a dropdown arrow.

# 3. Partage : profiter des ressources partagées par les collègues

The screenshot shows a web browser window displaying a Moodle Caseine interface. The page title is "Shared Caseine Space - Google Chrome". The URL is "moodle.caseine.org/local/sharedspace/hv/index.php". The user is logged in as "Yvan Maillot".

The main content area shows a search result for "Resultat: 235". Below this, there is a download button for "Microsoft Excel (.xlsx)" and a "Télécharger" button. A pagination bar shows page 1 selected.

A table lists activities with the following columns: Line, Id, Module, Add Cart, Course Name, Activity shared to teaching community, Activity type, Main Dewey classification, and Second Dewey classification.

Line	Id	Module	Add Cart	Course Name	Activity shared to teaching community	Activity type	Main Dewey classification	Second Dewey classification
1	28	Palindrome		Informatique - python (GI, 1A-IPID)	1	vpl	Computer programming, programs & data	Data proces: compl scienc
2	57	Couleur		Informatique - python (GI, 1A-IPID)	1	vpl	Computer programming, programs & data	Data proces: compl scienc

The right sidebar contains several filter sections, each with a "Rechercher" button:

- Activity type: Aucune sélection
- Main Dewey classification: Aucune sélection
- Disciplinary key concepts: Aucune sélection
- Author(s): Aucune sélection
- Programming Language: Java (selected)

At the bottom right, there is a "Panier d'activités" (Activity Basket) section with a plus icon and a settings gear icon.

# 3. Partage : profiter des ressources partagées par les collègues

The screenshot shows a web browser window displaying a Moodle Caseine Space. The page title is "Caseine" and the language is set to "Français (fr)". The main content area displays a list of shared resources. The first resource is titled "Patient" and is associated with the course "Informatique - python (GI, 1A-IPID)". The second resource is titled "MultiplesDeTroisEtCinq\*" and is also associated with the same course. The sidebar on the right contains a "Panier d'activités" (Activity Basket) with four items: "Coquille vide", "Coquille vide avec Tests Graphiques", "Coquille vide pour C++", and "Palindrome".

ID	Number of Resources	Resource Name	Course	Category	Subject
3	58	Patient	Informatique - python (GI, 1A-IPID)	vpl	Data processing & computer science
4	67	MultiplesDeTroisEtCinq*	Informatique - python (GI, 1A-IPID)	vpl	Computer programming, programs & data

**Panier d'activités**

- Coquille vide
- Coquille vide avec Tests Graphiques
- Coquille vide pour C++
- Palindrome

# 3. Partage : profiter des ressources partagées par les collègues

The screenshot shows a Moodle course interface. The browser address bar indicates the URL: moodle.caseine.org/course/view.php?id=446#section-0. The course name is 'Caseine' and the specific course is 'Java-M2-UHA'. The left sidebar contains a navigation menu with items: Participants, Compétences, Notes, Introduction, Introspection, Énumération, Annotations, Lambda-Expressions, Internationalisation, Client-Serveur, and Synchronisation avancée. The main content area is organized into sections:

- Annonces**: Contains 'Palindrome' (marked 'Caché pour les étudiants') and a '+ Ajouter une activité ou ressource' button.
- Introduction**: Contains 'IntroCMJAVA-article' and a '+ Ajouter une activité ou ressource' button.
- Introspection**: Contains 'introspection-fascicule-paysage', 'introspection-fascicule-portrait', 'Dumper', and 'AnalyseurDeClassesARemplir', each with a 'Modifier' button and a checkbox.

The right-hand panel, titled 'Partager d'activités', shows a list of activities to share: 'Coquille vide', 'Coquille vide avec Tests', 'Graphiques', 'Coquille vide pour C++', and 'Palindrome', each with a 'Partager' icon and a 'Supprimer' icon. The user 'Yvan Maillot' is logged in, as indicated by the profile picture and name in the top right corner.

# 3. Partage : faire profiter les collègues de ses ressources

The screenshot shows a Moodle course interface. The browser address bar indicates the URL is `moodle.caseine.org/course/view.php?id=446&section=0`. The course name is "Compléments Java M2 (UHA)".

**Sidebar (Left):**

- Java-M2-UHA (selected)
- Participants
- Compétences
- Notes
- Introduction
- Introspection
- Énumération
- Annotations
- Lambda-Expressions
- Internationalisation
- Client-Serveur
- Synchronisation avancée

**Main Content Area:**

- Introspection** (Section Header)
  - IntroCMJAVA-article (Resource)
  - + Ajouter une activité ou ressource
  - Introspection** (Section Header)
    - introspection-fascicule-paysage (Resource)
    - introspection-fascicule-portrait (Resource)
    - Dumper (Resource)
    - AnalyseurDeClassesARemplir (Resource)
    - Vidéo Cours "Introspection" (Resource)
  - Le mot de passe est pRSzIFS3
  - + Ajouter une activité ou ressource
- Énumération** (Section Header)
  - Enum-fascicule-portrait (Resource)
  - + Ajouter une activité ou ressource

# 3. Partage : faire profiter les collègues de ses ressources

Compléments Java M2 (UHA) Dumper Description - Google Chrome

moodle.caseine.org/mod/vpl/view.php?id=29351

Caseine

Yvan Maillot

Description Liste des devoirs rendus Similarité Activité test

## Dumper

**Required Files:** edu/uha/miage/ReflectUtil.java (Télécharger)

**Nombre maximal de fichiers:** 20

**Type de travail:** Travail individuel

**Réglages des notes:** Note maximale: 20 Caché

**Basé sur:** DefaultActivityJavaPackage

**Exécuter:** Oui. **Dégoguer:** Oui. **Évaluer:** Oui

**Note automatique:** Oui. **Temps d'exécution maximal:** 32 s. **Mémoire maximale utilisée:** 384 MiB. **Taille maximale d'un fichier:** 100 Mo

### Fichiers requis

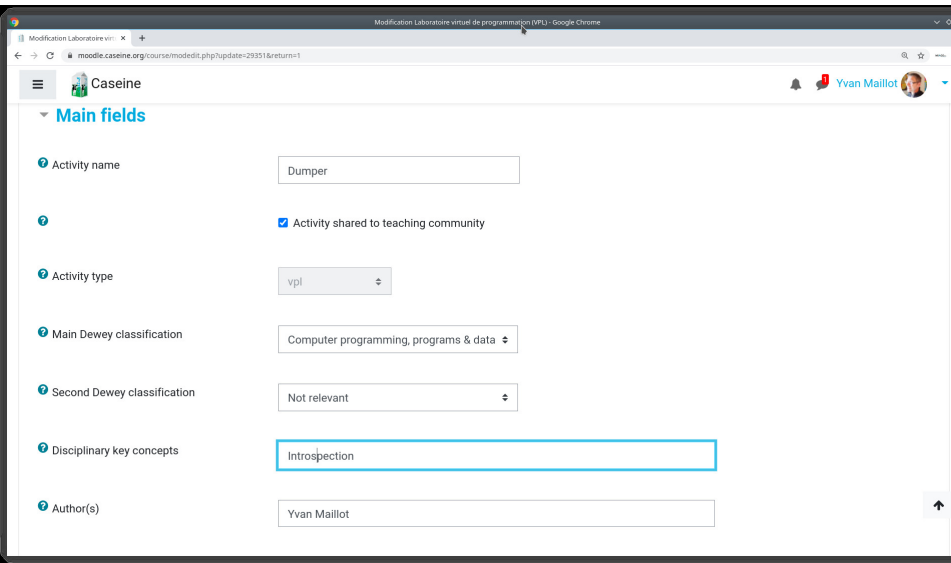
edu/uha/miage/ReflectUtil.java

```
1 package edu.uha.miage;
2
3 import java.lang.reflect.InvocationTargetException;
4 import java.lang.reflect.Method;
5 import java.lang.reflect.Modifier;
6 import java.util.*;
7 import java.util.logging.Level;
8 import java.util.logging.Logger;
9
10 /**
11  * @author Yvan Maillot (yvan.maillot@uha.fr)
12  */
13 public class ReflectUtil {
14
15     /**
16      * 1.01 Méthode qui retourne vrai si l'objet de classe passé en paramètre
17      * est de printif ou de type String, faux sinon.
18      */
19 }
```

Paramètres

- Cas de tests
- Options d'exécution
- Fichiers requis
- Corrected files
- Réglages avancés
  - Fichiers d'exécution
  - Limites maximales des ressources d'exécution
  - Fichiers à conserver à l'exécution
  - Variantes
  - Tester les serveurs d'exécution
  - Serveurs d'exécution locaux
- Activité test
  - Devoir rendu
  - Éditer
  - Voir la soumission
  - Note
  - Soumissions précédentes

# 3. Partage : faire profiter les collègues de ses ressources



The screenshot shows a web browser window with the URL `moodle.caseine.org/course/modedit.php?update=29331&return=1`. The page title is "Modification Laboratoire virtuel de programmation (VPL) - Google Chrome". The user is logged in as "Yvan Maillot". The page displays the "Main fields" section for editing an activity named "Dumper".

**Main fields**

- Activity name: Dumper
- Activity shared to teaching community
- Activity type: vpl
- Main Dewey classification: Computer programming, programs & data
- Second Dewey classification: Not relevant
- Disciplinary key concepts: Introspection
- Author(s): Yvan Maillot

# 3. Partage : faire profiter les collègues de ses ressources

Modification Laboratoire virtuel de programmation (VPL) - Google Chrome

moodle.caseine.org/course/modedit.php?update=29331&return=1

Caseine

Yvan Maillot

Difficulty level: Medium

Contact(s):

Required concepts:

Pedagogical objectives: Discovery

License: CC-BY-SA v4.0

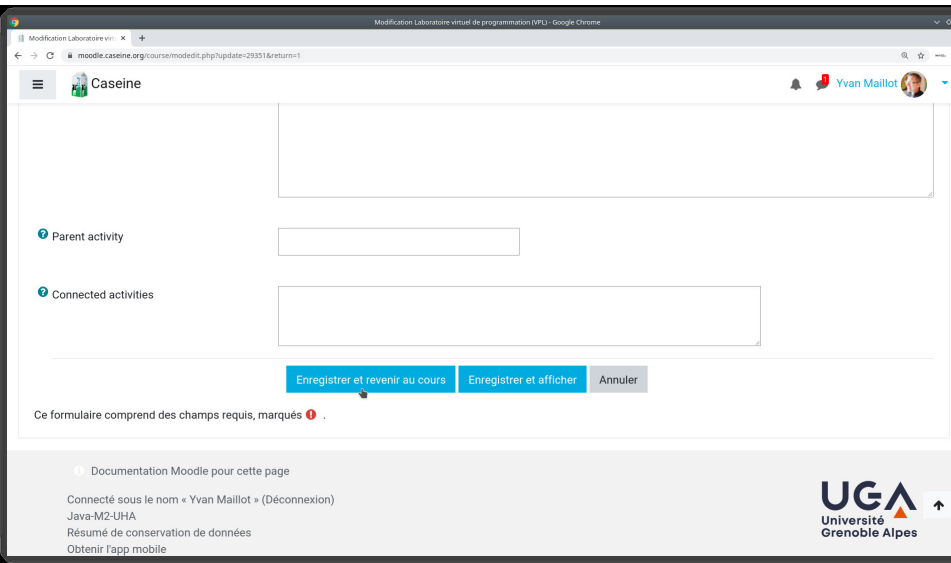
Version:

Reviewers:

Correction available to students



# 3. Partage : faire profiter les collègues de ses ressources



The screenshot shows a Moodle course page titled "Modification Laboratoire virtuel de programmation (VPL) - Google Chrome". The URL is "moodle.caseine.org/course/modedit.php?update=29351&return=1". The page header includes the Moodle logo, the course name "Caseine", and the user profile "Yvan Maillot".

The main content area contains a form with the following sections:

- A large empty text area at the top.
- A section labeled "Parent activity" with a small input field.
- A section labeled "Connected activities" with a larger empty text area.

At the bottom of the form, there are three buttons: "Enregistrer et revenir au cours", "Enregistrer et afficher", and "Annuler".

Below the form, a message states: "Ce formulaire comprend des champs requis, marqués ⓘ".

The footer contains the following information:

- Documentation Moodle pour cette page
- Connecté sous le nom « Yvan Maillot » (Déconnexion)
- Java-M2-UHA
- Résumé de conservation de données
- Obtenir l'app mobile

The UGA logo (Université Grenoble Alpes) is visible in the bottom right corner.

# 3. Partage : système de mesure des méta-données

## + Exercices à faire en autonomie (avant le 23 janvier)

Modifier ▾



**Accès restreint** Disponible à partir du **16 janvier 2023, 16:00**

### + Programme Minimum

100%

Modifier ▾

**Accès restreint** Disponible à partir du **16 janvier 2023, 16:05** (caché sinon)

### + Programme Minimum (CORRECTION)

69%

Modifier ▾

# 3. Partage : cours partagés



## Caseine



Caseine est une plateforme d'apprentissage en génie industriel, informatique et mathématiques. Son objectif est de favoriser l'apprentissage et l'autonomie des étudiants tout en améliorant la qualité du temps que l'enseignant leur consacre. Basée sur Moodle elle permet

1. d'évaluer automatiquement le code informatique et les modèles mathématiques des étudiants,
2. d'assurer un suivi pédagogique des étudiants,
3. de partager des contenus entre les enseignants à travers une communauté d'utilisateurs.

[Contacts](#) - [Infos](#) - [Les enseignants](#) - [Pour en savoir plus](#) - [Film de présentation](#)

**402** utilisateurs connectés au cours des dernières 24 heures (**8060** au cours du semestre écoulé)  
 **204** cours actifs ?  
 **3927** soumissions VPL quotidiennes en moyenne sur l'année écoulée

L'exécution et l'évaluation des activités de programmation (VPL) sont à nouveau disponibles. En cas de difficulté résiduelle, contactez vos responsables de cours.

## Mes cours

[Découvrir caseine](#)

[Operations Research \(UGA\)](#)

[Théorie des Graphes](#)

[Jeu de piste en Graphistan](#)

# 3. Partage : cours partagés



## Caseine



Caseine est une plateforme d'apprentissage en génie industriel, informatique et mathématiques. Son objectif est de favoriser l'apprentissage et l'autonomie des étudiants tout en améliorant la qualité du temps que l'enseignant leur consacre. Basée sur Moodle elle permet

1. d'évaluer automatiquement le code informatique et les modèles mathématiques des étudiants,
2. d'assurer un suivi pédagogique des étudiants,
3. de partager des contenus entre les enseignants à travers une communauté d'utilisateurs.

[Contacts](#) - [Infos](#) - [Les enseignants](#) - [Pour en savoir plus](#) - [Film de présentation](#)

**402** utilisateurs connectés au cours des dernières 24 heures (**8060** au cours du semestre écoulé)  
 **204** cours actifs ?  
 **3927** soumissions VPL quotidiennes en moyenne sur l'année écoulée

L'exécution et l'évaluation des activités de programmation (VPL) sont à nouveau disponibles. En cas de difficulté résiduelle, contactez vos responsables de cours.

## Mes cours

[Découvrir caseine](#)



[Operations Research \(UGA\)](#)



[Théorie des Graphes](#)



[Jeu de piste en Graphistan](#)

# 3. Partage : cours partagés

Cours : Théorie des Graphes | Français (fr) | Yvan Maillot

**Contributors :** Nadia Brauner (Université Grenoble Alpes), Nicolas Catusse (Grenoble INP), Hadrien Cambazard (Grenoble INP), Bernard Penz (Grenoble INP)

**Contact :** [Nadia Brauner](#)

- Polycopié sur les graphes (G-INP) 1
- Exercices (UGA)
- Annonces
- An article on this course
- Some english activities in this course

---

### Un peu de combinatoire pour commencer

**En Java**

- Somme d'entiers Java
- Fibonacci Java
- Factorielle Java
- \* Sous-ensembles à k éléments Java
- \* MotsLongueurN Java  1

**En Python**

ACTIVITÉ DEPUIS LE DÉBUT DU COURS

2023, 11:56

[Rapport complet des activités récentes...](#)

Aucune activité récente

---

Ma bouteille de graphes

0%

[Voir mon progrès](#)

**Classement**

- 24%
- 15%
- 14%
- 13%
- 12%
- ⋮
- 57) 0% Yvan Maillot

# 3. Partage : cours partagés

Cours : Théorie des Grap... +

https://moodle.caseine.org/course/view.php?id=147

133%

Caseine Français (fr) Yvan Maillot

## Théorie des Graphes


Accueil / Mes cours / GraphsOpen

- Me désinscrire de GraphsOpen
- Prendre temporairement le rôle Visiteur de cours

### Graphes

Contributors : Nadia Brauner (Université Grenoble Alpes), Nicolas Catusse (Grenoble INP), Hadrien Cambazard (Grenoble INP), Bernard Penz (Grenoble INP)

Contact : [Nadia Brauner](#)

- Polycopié sur les graphes (GI-INP)  1
- Exercices (UGA)
- Annonces
- An article on this course
- Some english activities in this course


Activité récente

Activité depuis le samedi 13 mai 2023, 11:56

[Rapport complet des activités récentes...](#)

Aucune activité récente

Ma bouteille de graphes



0%

[Voir mon progrès](#)

Un peu de combinatoire pour commencer

En Java

Votre progression ?

# 3. Partage : cours partagés

Cours : Théorie des Graphes | Français (fr) | Yvan Maillot

Some english activities in this course

## Un peu de combinatoire pour commencer 🛒

### En Java

- Somme d'entiers Java
- Fibonacci Java
- Factorielle Java
- \* Sous-ensembles à k éléments Java
- \* MotsLongueurN Java

### En Python

- Somme d'entiers Python
- Fibonacci Python
- \* Sous-ensembles à k éléments Python
- \* MotsLongueurN Python
- Réversivité, énumération Python

HOW MANY DIGITS OF PI DO YOU HAVE

Activité récente

Activité depuis le samedi 13 mai 2023, 11:56

[Rapport complet des activités récentes...](#)

Aucune activité récente

Ma bouteille de graphes

0%

[Voir mon progrès](#)

# 3. Partage : cours partagés

Cours : Théorie des Grap... | Français (fr) | Yvan Maillot











## Graphes


Contributors : Nadia Brauner (Université Grenoble Alpes), Nicolas Catusse (Grenoble INP), Hadrien Cambazard (Grenoble INP), Bernard Penz (Grenoble INP)

Contact : [Nadia Brauner](#)


**Vous devez être inscrit dans le cours pour avoir accès à l'éditeur des exercices de programmation (roue dentée en haut à droite).**

**Accès restreint** Non disponible à moins que : Vous n'êtes pas *Étudiant* (caché sinon)


-  Polycopié sur les graphes (GHNP)  1 
-  Exercices (UGA) 
-  Annonces
-  An article on this course 
-  Some english activities in this course 

Un peu de combinatoire pour commencer 

**En Java**

Graphes  [Copier la section](#)

This block is currently on restricted visitor mode. You can only backup activities with a Shared metadata that is set.

 Leave a message to Nadia Brauner



# 3. Partage : cours partagés

Cours : Théorie des Grap... | Caseine | Français (fr) | Yvan Maillot

## Graphes

Votre progression ?

**Contributors :** Nadia Brauner (Université Grenoble Alpes), Nicolas Catusse (Grenoble INP), Hadrien Cambazard (Grenoble INP), Bernard Penz (Grenoble INP)

**Contact :** [Nadia Brauner](#)

**Vous devez être inscrit dans le cours pour avoir accès à l'éditeur des exercices de programmation (roue dentée en haut à droite).**

**Accès restreint** Non disponible à moins que : Vous n'êtes pas *Étudiant* (caché sinon)

Polycopié sur les graphes (G-INP) 1

Exercices (UGA)

Annonces

An article on this course

Some english activities in this course

Un peu de combinatoire pour commencer

En Java

## Panier d'activités

Capacités requises manquantes :

Coquille vide

Coquille vide avec Tests G...

Coquille vide python - unitt...

Coquille vide pour C++

\* Sous-ensembles à k élé...

Graphes

Copier la section

This block is currently on restricted visitor mode. You can only backup activities with a Shared metadata that is set.

Leave a message to Nadia Brauner

## 3. Partage

- Mattermost (un espace de discussion)
- Espace de partage (panier)
- Méta-données (système de mesure)
- Cours partagés
- **Événements fréquents :**
  - **demi-heures Caseine**
  - **demi-journées Caseine**
- Communauté d'enseignants très active, plateforme ouverte ⇒ évolution au profit de tous
  - développement d'activités Moodle
  - de plugins,
  - de scripts
  - ...

## 3. Partage

- Mattermost (un espace de discussion)
- Espace de partage (panier)
- Méta-données (système de mesure)
- Cours partagés
- Événements fréquents :
  - demi-heures Caseine
  - demi-journées Caseine
- **Communauté d'enseignants très active, plateforme ouverte** ⇒ évolution au profit de tous
  - développement d'activités Moodle
  - de plugins,
  - de scripts
  - ...

Une plateforme d'apprentissage en génie industriel, informatique et mathématiques.

Son objectif est de favoriser l'apprentissage et l'autonomie des étudiants tout en améliorant la qualité du temps que l'enseignant leur consacre.

Basée sur Moodle elle permet :

- d'évaluer automatiquement le code informatique et les modèles mathématiques des étudiants,
- d'assurer un suivi pédagogique des étudiants,
- de partager des contenus entre les enseignants à travers une communauté d'utilisateurs.

# Modules informatiques

- **Programmation Orientée Objet et Java (6 ECTS)**
- **Structures de données en C++ (3 ECTS)**

## Contributeurs

- Emmanuel Coquery
- Frédéric Cordier
- Yvan Maillot

# Modules informatiques

- **Programmation Orientée Objet et Java (6 ECTS)**
- **Structures de données en C++ (3 ECTS)**

## Contributeurs

- Emmanuel Coquery
- Frédéric Cordier
- Yvan Maillot

Université Claude Bernard  Lyon 1

# Modules informatiques

- **Programmation Orientée Objet et Java (6 ECTS)**
- **Structures de données en C++ (3 ECTS)**

## Contributeurs

- Emmanuel Coquery
- Frédéric Cordier
- Yvan Maillot

Université Claude Bernard  Lyon 1



# Modules informatiques

- **Programmation Orientée Objet et Java (6 ECTS)**
- **Structures de données en C++ (3 ECTS)**

## Leurs usages possibles (grâce à l'auto-évaluation)

- en autonomie
- par la classe inversée (hybride 50/50)
- de façon traditionnelle
- ...



# Modules informatiques

- **POO Java**
- **Structures de données en C++ (3 ECTS)**

## Leurs usages possibles (grâce à l'auto-évaluation)

- en autonomie
- par la classe inversée (hybride 50/50)
- de façon traditionnelle
- ...

# Modules informatiques

- **POO Java**
- **SDD C++**

## Leurs usages possibles (grâce à l'auto-évaluation)

- en autonomie
- par la classe inversée (hybride 50/50)
- de façon traditionnelle
- ...

# En autonomie

# Hybridation de POO Java

## Traditionnel

### Chaque semaine (5x)

- 2h CM (tous)
- 2h TD (groupe)



- Examen final

## Classe inversée

- 2h CM (tous)

### Chaque semaine (5x)

- Travail en autonomie
- 2h TD (groupe)

- Examen final

# POO Java en Classe inversée

## Classe inversée – 12h CM – 10h TD – 3ECTS

- Explication des modalités en amphi (tous) : **2h**

### Chaque semaine (5x)

- Travail en autonomie, cours + exos notés (coef  $\frac{1}{4}$ )
- TD en salle info en groupe (présence notée  $\frac{1}{20}$ ) : **2h**
- Examen final (coef  $\frac{7}{10}$ )

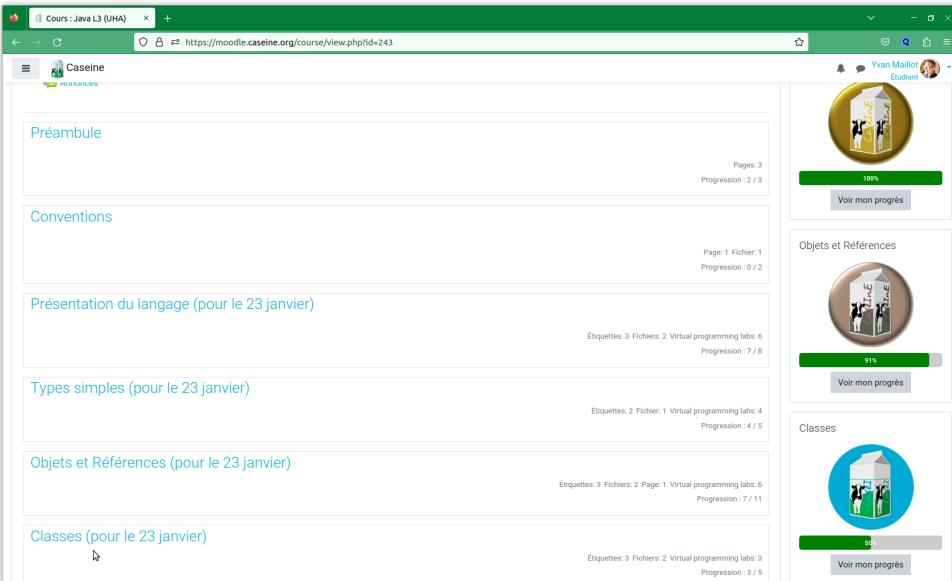
# POO Java en Classe inversée

The screenshot shows a web browser window with the URL `https://moodle.caseine.org/course/view.php?id=243`. The page title is "Java L3 (UHA)". The breadcrumb trail is "Accueil / Cours / Courses: Computer Science / JavaL3UHA".

On the left side, there is a section titled "Annonces" (Announcements) with a sub-section "Préambule" (Preamble) showing "Pages: 3" and "Progression: 2 / 3". Below it is "Conventions" showing "Page: 1 Fichier: 1" and "Progression: 0 / 2". Further down is "Présentation du langage (pour le 23 janvier)" (Language presentation (for January 23)) showing "Étiquettes: 3 Fichiers: 2 Virtual programming labs: 6" and "Progression: 7 / 8". At the bottom is "Types simples (pour le 23 janvier)" (Simple types (for January 23)) showing "Étiquettes: 2 Fichier: 1 Virtual programming labs: 4" and "Progression: 4 / 5".

On the right side, there are two progress indicators. The first is for "Présentation du langage" (Language presentation) with a progress bar at 100% and a "Voir mon progrès" (View my progress) button. The second is for "Objets et Références" (Objects and References) with a progress bar at 91% and a "Voir mon progrès" button. Below these is a section for "Classes".

# POO Java en Classe inversée



The screenshot shows a Moodle course page for 'Cours : Java L3 (UHA)'. The browser address bar shows the URL: <https://moodle.caseine.org/course/view.php?id=243>. The course name is 'Caseine'. The page displays a list of topics with their respective progress bars and completion status.

Topic	Progress	Completion Status
Préambule	Pages: 3 Progression : 2 / 3	100%
Conventions	Page: 1 Fichier: 1 Progression : 0 / 2	
Présentation du langage (pour le 23 janvier)	Étiquettes: 3 Fichiers: 2 Virtual programming labs: 6 Progression : 7 / 8	91%
Types simples (pour le 23 janvier)	Étiquettes: 2 Fichier: 1 Virtual programming labs: 4 Progression : 4 / 5	
Objets et Références (pour le 23 janvier)	Étiquettes: 3 Fichiers: 2 Page: 1 Virtual programming labs: 6 Progression : 7 / 11	
Classes (pour le 23 janvier)	Étiquettes: 3 Fichiers: 2 Virtual programming labs: 3 Progression : 3 / 5	50%

On the right side of the page, there is a user profile for 'Yvan Maillot' (Étudiant) and a navigation menu with icons for 'Objets et Références' and 'Classes'.





# POO Java en Classe inversée

Description

Liste des devoirs rendus

Similarité

Activité test

Groupes séparés

Gr1

Sélection de la soumission

Tous les envois

Évaluer

Choisir...

		Prénom / Nom	Soumis le	Devoirs rendus	Note	Évaluateur	Noté le
7		[Redacted]	dimanche 8 mars 2020, 14:45	2	2,21 / 20,00 Caché	Note automatique	dimanch
8		[Redacted]	mercredi 19 février 2020, 16:33	8	7,05 / 20,00 Caché	Note automatique	mercredi
9		[Redacted]	lundi 9 mars 2020, 00:12	14	8,63 / 20,00 Caché	Note automatique	lundi 9 m
10		[Redacted]	mercredi 19 février 2020, 12:40	27	11,53 / 20,00 Caché	Note automatique	mercredi
11		[Redacted]	mercredi 4 mars 2020, 18:44	27	14,00 / 20,00 Caché	Note automatique	mercredi
12		[Redacted]	dimanche 8 mars 2020, 13:09	36	15,00 / 20,00 Caché	Note automatique	dimanch
13		[Redacted]	samedi 7 mars 2020, 13:33	49	16,11 / 20,00 Caché	Note automatique	samedi 7

## Classe traditionnelle – 14h CM – 18h TD – 4 ECTS

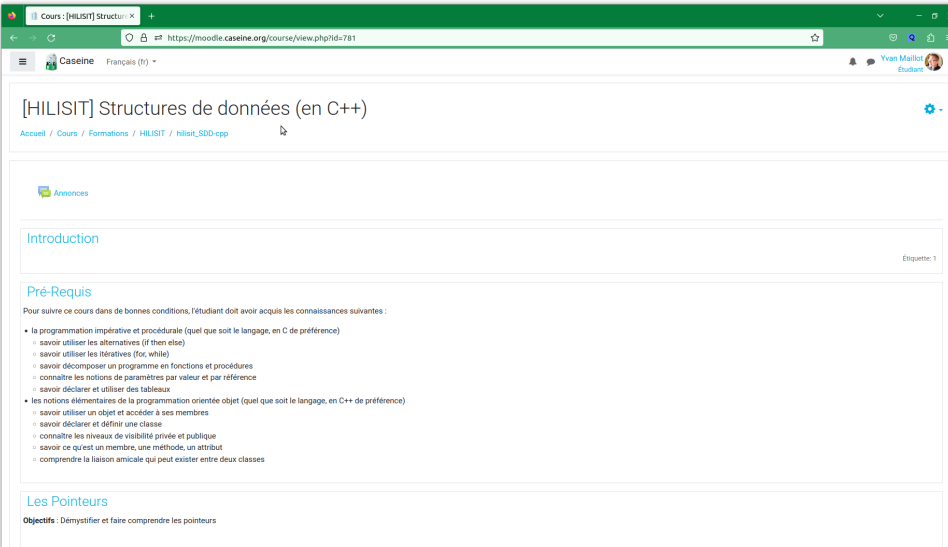
### Chaque semaine (4x)

- CM : 1h30 - amphi (tous)
- TD - salle info (groupe) : 2h (noté Caseine coef 0.25)
- Examen 1 sur table (coef 0.75)

### Chaque semaine (4x)

- CM : 1h30 - amphi (tous)
- TD - salle info (groupe) : 2h (noté Caseine coef 0.25)
- TD - salle info (groupe) : 2h (préparation au final)
- Examen 2 sur table (coef 1)

# SDD C++ en classe traditionnelle

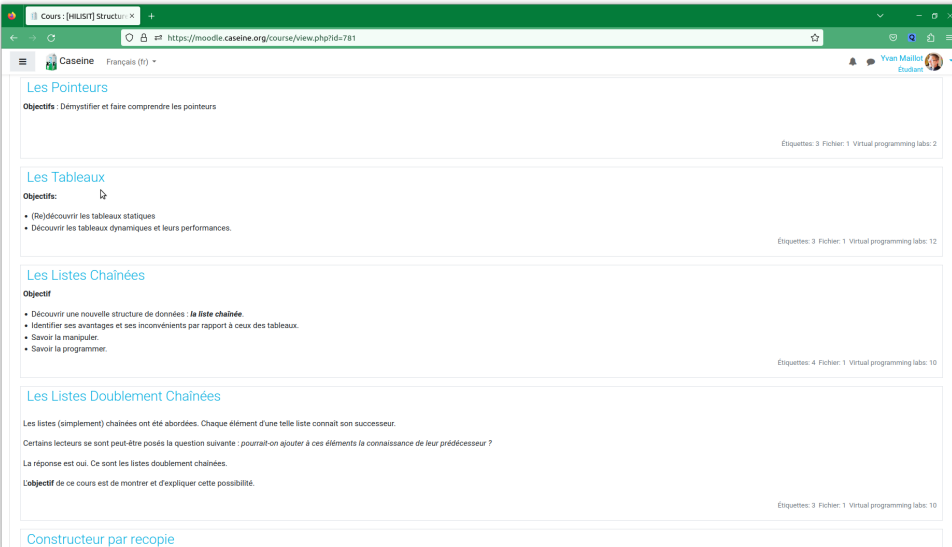


The screenshot shows a web browser window displaying a Moodle course page. The browser's address bar shows the URL: `https://moodle.caseine.org/course/view.php?id=781`. The page title is "[HILISIT] Structures de données (en C++)". The breadcrumb trail is "Accueil / Cours / Formations / HILISIT / hilisit\_SDD-cpp". The user is identified as "Yvan Maillot, Etudiant".

The page content includes:

- An "Annonces" (Announcements) section.
- An "Introduction" section with a label "Étiquette: 1".
- A "Pré-Requis" (Prerequisites) section with the text: "Pour suivre ce cours dans de bonnes conditions, l'étudiant doit avoir acquis les connaissances suivantes :".
  - la programmation impérative et procédurale (quel que soit le langage, en C de préférence)
    - savoir utiliser les alternatives (if then else)
    - savoir utiliser les itératives (for, while)
    - savoir décomposer un programme en fonctions et procédures
    - connaître les notions de paramètres par valeur et par référence
    - savoir déclarer et utiliser des tableaux
  - les notions élémentaires de la programmation orientée objet (quel que soit le langage, en C++ de préférence)
    - savoir utiliser un objet et accéder à ses membres
    - savoir déclarer et définir une classe
    - connaître les niveaux de visibilité privée et publique
    - savoir ce qu'est un membre, une méthode, un attribut
    - comprendre la liaison amicale qui peut exister entre deux classes
- A "Les Pointeurs" section with the text: "Objectifs : Démystifier et faire comprendre les pointeurs".

# SDD C++ en classe traditionnelle



The screenshot shows a Moodle course page for 'Cours : [HILISIT] Structures'. The browser address bar shows the URL: https://moodle.caseine.org/course/view.php?id=781. The course is in French (Français (fr)). The user is identified as Yvan Maillot, an Etudiant.

**Les Pointeurs**  
Objectifs : Démystifier et faire comprendre les pointeurs  
Étiquettes: 3 Fichier: 1 Virtual programming labs: 2

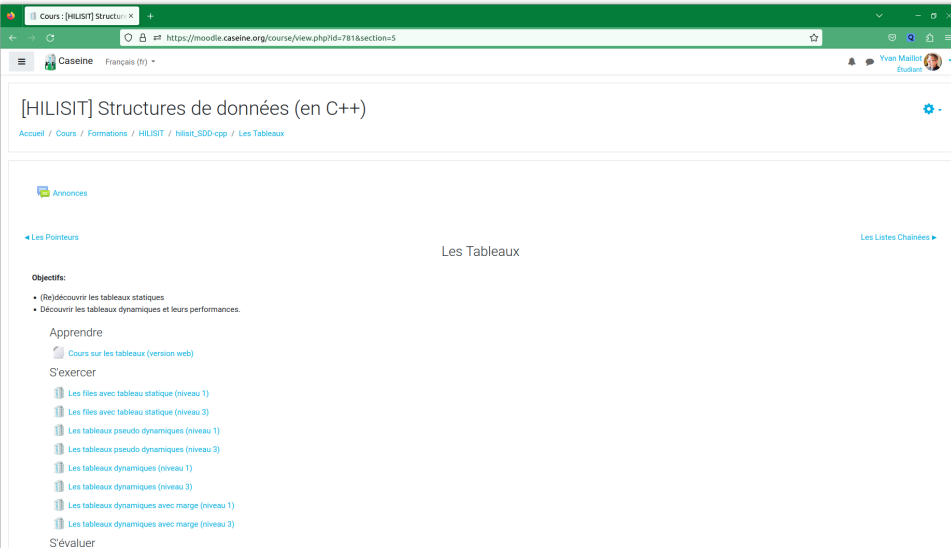
**Les Tableaux**  
Objectifs:  
• (Re)découvrir les tableaux statiques  
• Découvrir les tableaux dynamiques et leurs performances.  
Étiquettes: 3 Fichier: 1 Virtual programming labs: 12

**Les Listes Chainées**  
Objectif  
• Découvrir une nouvelle structure de données : la **liste chaînée**.  
• Identifier ses avantages et ses inconvénients par rapport à ceux des tableaux.  
• Savoir la manipuler.  
• Savoir la programmer.  
Étiquettes: 4 Fichier: 1 Virtual programming labs: 10

**Les Listes Doublement Chainées**  
Les listes (simplement) chaînées ont été abordées. Chaque élément d'une telle liste connaît son successeur.  
Certains lecteurs se sont peut-être posés la question suivante : *pourrait-on ajouter à ces éléments la connaissance de leur prédécesseur ?*  
La réponse est oui. Ce sont les listes doublement chaînées.  
L'objectif de ce cours est de montrer et d'expliquer cette possibilité.  
Étiquettes: 3 Fichier: 1 Virtual programming labs: 10

**Constructeur par recopie**

# SDD C++ en classe traditionnelle



Cours : [HILISIT] Structures de données

https://moodle.caseine.org/course/view.php?id=781&section=5

Caseine Français (fr)

Yvan Maillot  
Étudiant

## [HILISIT] Structures de données (en C++)

[Accueil](#) / [Cours](#) / [Formations](#) / [HILISIT](#) / [hilisit\\_SDD-cpp](#) / [Les Tableaux](#)

Annonces


← [Les Pointeurs](#) [Les Listes Chaînées](#) →

### Les Tableaux









**Objectifs:**

- (Re)découvrir les tableaux statiques
- Découvrir les tableaux dynamiques et leurs performances.

**Apprendre**

 [Cours sur les tableaux \(version web\)](#)

**S'exercer**

-  [Les files avec tableau statique \(niveau 1\)](#)
-  [Les files avec tableau statique \(niveau 3\)](#)
-  [Les tableaux pseudo dynamiques \(niveau 1\)](#)
-  [Les tableaux pseudo dynamiques \(niveau 3\)](#)
-  [Les tableaux dynamiques \(niveau 1\)](#)
-  [Les tableaux dynamiques \(niveau 3\)](#)
-  [Les tableaux dynamiques avec marge \(niveau 1\)](#)
-  [Les tableaux dynamiques avec marge \(niveau 3\)](#)

**S'évaluer**

# SDD C++ en classe traditionnelle

The screenshot shows a Moodle course page for 'Les Tableaux'. The browser address bar indicates the URL: `https://moodle.caseine.org/course/view.php?id=781&section=5`. The page header includes the course name 'Caseine', the language 'Français (fr)', and the user 'Yvan Maillot, Etudiant'. The main content area is titled 'Les Tableaux' and contains the following sections:

- Objectifs:**
  - (Re)découvrir les tableaux statiques
  - Découvrir les tableaux dynamiques et leurs performances.
- Apprendre**
  - Cours sur les tableaux (version web)
- S'exercer**
  - Les files avec tableau statique (niveau 1)
  - Les files avec tableau statique (niveau 3)
  - Les tableaux pseudo dynamiques (niveau 1)
  - Les tableaux pseudo dynamiques (niveau 3)
  - Les tableaux dynamiques (niveau 1)
  - Les tableaux dynamiques (niveau 3)
  - Les tableaux dynamiques avec marge (niveau 1)
  - Les tableaux dynamiques avec marge (niveau 3)
- S'évaluer**
  - Les files avec tableau statique (niveau 2)
  - Les tableaux pseudo dynamiques (niveau 2)
  - Les tableaux dynamiques (niveau 2)
  - Les tableaux dynamiques avec marge (niveau 2)

Navigation links include 'Les Listes Chainées' and 'Aller à...'.

# SDD C++ en classe traditionnelle

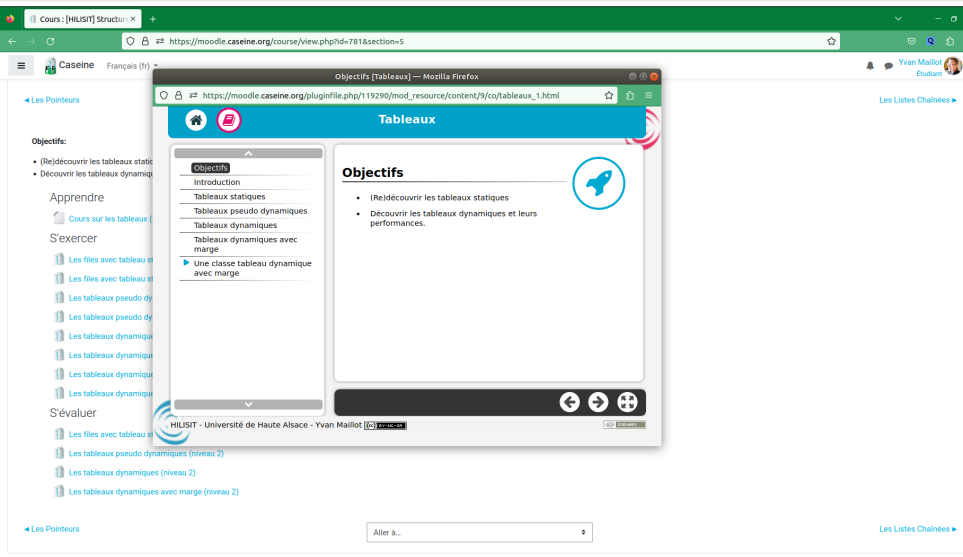
The screenshot shows a Moodle course page for 'Tableaux' (Tables) in the 'Caseine' system. The browser address bar shows the URL: <https://moodle.caseine.org/course/view.php?id=781&section=5>. The page content includes:

- Objectifs:**
  - (Re)découvrir les tableaux statiques
  - Découvrir les tableaux dynamiques
- Apprendre:**
  - Cours sur les tableaux (statiques)
- S'exercer:**
  - Les files avec tableaux statiques
  - Les files avec tableaux dynamiques
  - Les tableaux pseudo dynamiques
  - Les tableaux pseudo dynamiques
  - Les tableaux dynamiques
  - Les tableaux dynamiques
  - Les tableaux dynamiques
  - Les tableaux dynamiques
- S'évaluer:**
  - Les files avec tableaux statiques
  - Les tableaux pseudo dynamiques (niveau 2)
  - Les tableaux dynamiques (niveau 2)
  - Les tableaux dynamiques avec marge (niveau 2)

A Firefox window titled 'Tableaux - Mozilla Firefox' is overlaid on the page, showing the course content. The URL in the Firefox window is: [https://moodle.caseine.org/pluginfile.php/119290/mod\\_resource/content/9/ca/tableaux.html](https://moodle.caseine.org/pluginfile.php/119290/mod_resource/content/9/ca/tableaux.html). The content of the Firefox window includes:

- Logo for 'uniscle les sciences, l'essentielle'
- Section title: **Tableaux**
- Text: HILISIT - Université de Haute Alsace - Yvan Maillot
- Version: 1.0
- A pink button labeled 'Démarrer le module' with a right-pointing arrow.
- Footer: HILISIT - Université de Haute Alsace - Yvan Maillot

# SDD C++ en classe traditionnelle



The screenshot shows a Moodle course page for 'Tableaux' (Tables) in French. The page is viewed through Mozilla Firefox. The main content area is titled 'Objectifs' (Objectives) and contains a list of learning goals. A zoomed-in window highlights the 'Objectifs' section, showing a table of contents and a list of objectives.

**Objectifs**

- (Re)découvrir les tableaux statiques
- Découvrir les tableaux dynamiques

**Apprendre**

Cours sur les tableaux (1)

**S'exercer**

- Les files avec tableaux statiques (niveau 1)
- Les files avec tableaux dynamiques (niveau 1)
- Les tableaux pseudo dynamiques (niveau 1)
- Les tableaux pseudo dynamiques (niveau 2)
- Les tableaux dynamiques (niveau 1)
- Les tableaux dynamiques (niveau 2)
- Les tableaux dynamiques (niveau 3)
- Les tableaux dynamiques (niveau 4)
- Les tableaux dynamiques (niveau 5)
- Les tableaux dynamiques (niveau 6)
- Les tableaux dynamiques (niveau 7)
- Les tableaux dynamiques (niveau 8)
- Les tableaux dynamiques (niveau 9)
- Les tableaux dynamiques (niveau 10)
- Les tableaux dynamiques (niveau 11)
- Les tableaux dynamiques (niveau 12)
- Les tableaux dynamiques (niveau 13)
- Les tableaux dynamiques (niveau 14)
- Les tableaux dynamiques (niveau 15)
- Les tableaux dynamiques (niveau 16)
- Les tableaux dynamiques (niveau 17)
- Les tableaux dynamiques (niveau 18)
- Les tableaux dynamiques (niveau 19)
- Les tableaux dynamiques (niveau 20)
- Les tableaux dynamiques (niveau 21)
- Les tableaux dynamiques (niveau 22)
- Les tableaux dynamiques (niveau 23)
- Les tableaux dynamiques (niveau 24)
- Les tableaux dynamiques (niveau 25)
- Les tableaux dynamiques (niveau 26)
- Les tableaux dynamiques (niveau 27)
- Les tableaux dynamiques (niveau 28)
- Les tableaux dynamiques (niveau 29)
- Les tableaux dynamiques (niveau 30)
- Les tableaux dynamiques (niveau 31)
- Les tableaux dynamiques (niveau 32)
- Les tableaux dynamiques (niveau 33)
- Les tableaux dynamiques (niveau 34)
- Les tableaux dynamiques (niveau 35)
- Les tableaux dynamiques (niveau 36)
- Les tableaux dynamiques (niveau 37)
- Les tableaux dynamiques (niveau 38)
- Les tableaux dynamiques (niveau 39)
- Les tableaux dynamiques (niveau 40)
- Les tableaux dynamiques (niveau 41)
- Les tableaux dynamiques (niveau 42)
- Les tableaux dynamiques (niveau 43)
- Les tableaux dynamiques (niveau 44)
- Les tableaux dynamiques (niveau 45)
- Les tableaux dynamiques (niveau 46)
- Les tableaux dynamiques (niveau 47)
- Les tableaux dynamiques (niveau 48)
- Les tableaux dynamiques (niveau 49)
- Les tableaux dynamiques (niveau 50)
- Les tableaux dynamiques (niveau 51)
- Les tableaux dynamiques (niveau 52)
- Les tableaux dynamiques (niveau 53)
- Les tableaux dynamiques (niveau 54)
- Les tableaux dynamiques (niveau 55)
- Les tableaux dynamiques (niveau 56)
- Les tableaux dynamiques (niveau 57)
- Les tableaux dynamiques (niveau 58)
- Les tableaux dynamiques (niveau 59)
- Les tableaux dynamiques (niveau 60)
- Les tableaux dynamiques (niveau 61)
- Les tableaux dynamiques (niveau 62)
- Les tableaux dynamiques (niveau 63)
- Les tableaux dynamiques (niveau 64)
- Les tableaux dynamiques (niveau 65)
- Les tableaux dynamiques (niveau 66)
- Les tableaux dynamiques (niveau 67)
- Les tableaux dynamiques (niveau 68)
- Les tableaux dynamiques (niveau 69)
- Les tableaux dynamiques (niveau 70)
- Les tableaux dynamiques (niveau 71)
- Les tableaux dynamiques (niveau 72)
- Les tableaux dynamiques (niveau 73)
- Les tableaux dynamiques (niveau 74)
- Les tableaux dynamiques (niveau 75)
- Les tableaux dynamiques (niveau 76)
- Les tableaux dynamiques (niveau 77)
- Les tableaux dynamiques (niveau 78)
- Les tableaux dynamiques (niveau 79)
- Les tableaux dynamiques (niveau 80)
- Les tableaux dynamiques (niveau 81)
- Les tableaux dynamiques (niveau 82)
- Les tableaux dynamiques (niveau 83)
- Les tableaux dynamiques (niveau 84)
- Les tableaux dynamiques (niveau 85)
- Les tableaux dynamiques (niveau 86)
- Les tableaux dynamiques (niveau 87)
- Les tableaux dynamiques (niveau 88)
- Les tableaux dynamiques (niveau 89)
- Les tableaux dynamiques (niveau 90)
- Les tableaux dynamiques (niveau 91)
- Les tableaux dynamiques (niveau 92)
- Les tableaux dynamiques (niveau 93)
- Les tableaux dynamiques (niveau 94)
- Les tableaux dynamiques (niveau 95)
- Les tableaux dynamiques (niveau 96)
- Les tableaux dynamiques (niveau 97)
- Les tableaux dynamiques (niveau 98)
- Les tableaux dynamiques (niveau 99)
- Les tableaux dynamiques (niveau 100)

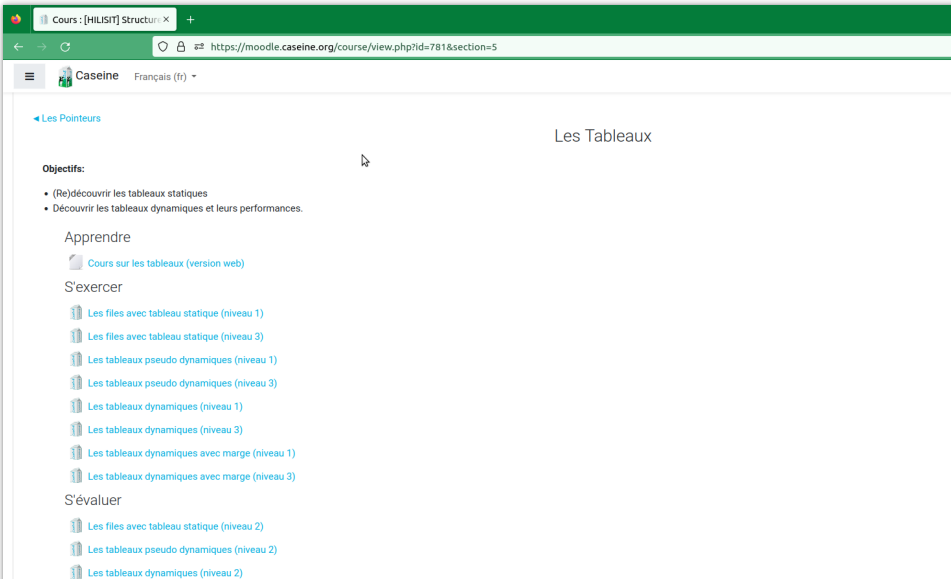
**S'évaluer**

- Les files avec tableaux statiques (niveau 1)
- Les tableaux pseudo dynamiques (niveau 2)
- Les tableaux dynamiques (niveau 2)
- Les tableaux dynamiques avec marge (niveau 2)

Navigation: Aller à... [dropdown]



# SDD C++ en classe traditionnelle



The screenshot shows a web browser window with the address bar displaying `https://moodle.caseine.org/course/view.php?id=781&section=5`. The page title is "Cours : [HILISIT] Structures". The user is logged in as "Caseine" and the language is set to "Français (fr)".


Navigation: [← Les Pointeurs](#)

## Les Tableaux









**Objectifs:**

- (Re)découvrir les tableaux statiques
- Découvrir les tableaux dynamiques et leurs performances.




**Apprendre**

-  [Cours sur les tableaux \(version web\)](#)

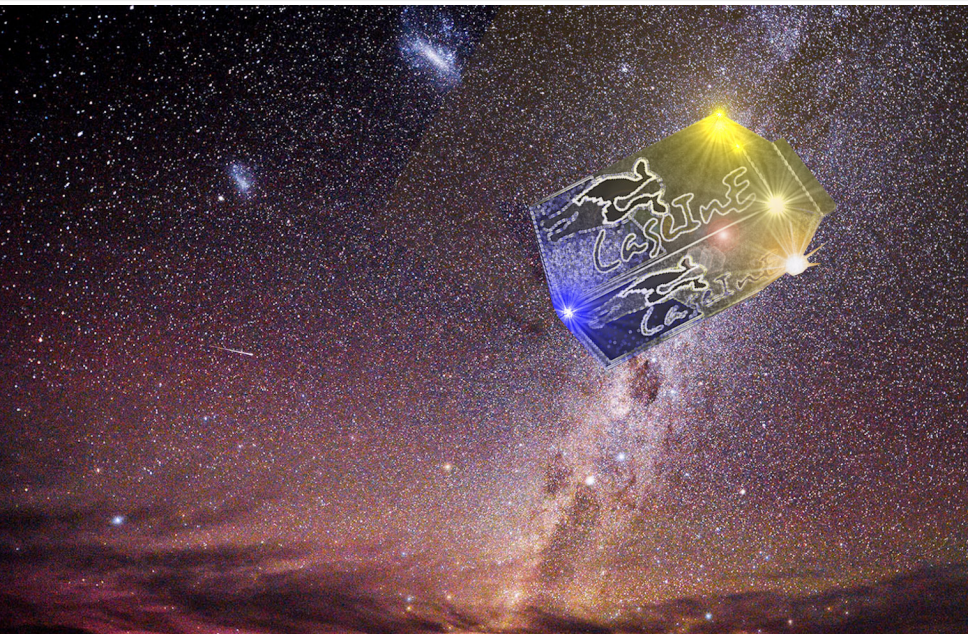
**S'exercer**

-  [Les files avec tableau statique \(niveau 1\)](#)
-  [Les files avec tableau statique \(niveau 3\)](#)
-  [Les tableaux pseudo dynamiques \(niveau 1\)](#)
-  [Les tableaux pseudo dynamiques \(niveau 3\)](#)
-  [Les tableaux dynamiques \(niveau 1\)](#)
-  [Les tableaux dynamiques \(niveau 3\)](#)
-  [Les tableaux dynamiques avec marge \(niveau 1\)](#)
-  [Les tableaux dynamiques avec marge \(niveau 3\)](#)

**S'évaluer**

-  [Les files avec tableau statique \(niveau 2\)](#)
-  [Les tableaux pseudo dynamiques \(niveau 2\)](#)
-  [Les tableaux dynamiques \(niveau 2\)](#)

# Suivez la voie lactée dans l' unisciel étoilé



Suivez la voie lactée dans l' étoilé



**PROJET**  
**HILISIT**  
Hybridation en Licence Scientifique

Suivez la voie lactée dans l' étoilé



Financé  
par



**GOUVERNEMENT**

*Liberté  
Égalité  
Fraternité*



Financé par  
l'Union Européenne

**PROJET HILISIT**  
Hybridation en Licence Scientifique

Yvan Maillot ([yvan.maillot@uha.fr](mailto:yvan.maillot@uha.fr))

Colloque final du projet HILISIT