

Réalisation d'exercices de programmation
auto-évalués sur une plateforme basée sur
Moodle et dédiée aux informaticiens et aux
mathématiciens : Caseine.



Retour d'expérience

Yvan Maillot (yvan.maillot@uha.fr)

- Enseignement de la programmation Java
- à la Faculté des Sciences et Techniques (FST)
- de l'Université de Haute-Alsace (UHA)
- en MIAGE (Méthodes Informatiques Appliquées à la Gestion des Entreprises)
 - en 3^{ème} année de licence (L3)
 - en 1^{ère} année de master (M1)
 - en 2^{ème} année de master (M2)

=> Plateforme d'apprentissage adaptée à des informaticiens débutants comme expérimentés

Pourquoi avoir choisi la classe inversée ?

Pourquoi avoir choisi la classe inversée ?

- Pas à cause de la COVID

Pourquoi avoir choisi la classe inversée ?

- Pas à cause de la COVID
- L'envie d'expérimenter

Pourquoi avoir choisi la classe inversée ?

- Pas à cause de la COVID
- L'envie d'expérimenter
- Augmentation des effectifs en L3

Pourquoi avoir choisi la classe inversée ?

- Pas à cause de la COVID
- L'envie d'expérimenter
- Augmentation des effectifs en L3
 - Jusqu'en 2018 cours et TD en salle machine

Pourquoi avoir choisi la classe inversée ?

-
- L'envie d'expérimenter
- Augmentation des effectifs en L3
 - Jusqu'en 2018 cours et TD en salle machine
 - Après 2018 cours en amphi et TD en salle machine (groupes)

Pourquoi avoir choisi la classe inversée ?

-
- L'envie d'expérimenter
- Augmentation des effectifs en L3
 - Jusqu'en 2018 cours et TD en salle machine
 - Après 2018 cours en amphi et TD en salle machine (groupes)
 - => Hybride

Pourquoi avoir choisi la classe inversée ?

-
- L'envie d'expérimenter
- Augmentation des effectifs en L3
 - Jusqu'en 2018 cours et TD en salle machine
 - Après 2018 cours en amphi et TD en salle machine (groupes)
 - => Hybride
 - Remplacer le cours en amphi par
 - des support et des ressources (à lire « à la maison »)

Pourquoi avoir choisi la classe inversée ?

-
- L'envie d'expérimenter
- Augmentation des effectifs en L3
 - Jusqu'en 2018 cours et TD en salle machine
 - Après 2018 cours en amphi et TD en salle machine (groupes)
 - => Hybride
 - Remplacer le cours en amphi par
 - des support et des ressources (à lire « à la maison »)
 - des exercices (à faire « à la maison »)

Pourquoi avoir choisi la classe inversée ?

-
- L'envie d'expérimenter
- Augmentation des effectifs en L3
 - Jusqu'en 2018 cours et TD en salle machine
 - Après 2018 cours en amphi et TD en salle machine (groupes)
 - => Hybride
 - Remplacer le cours en amphi par
 - des support et des ressources (à lire « à la maison »)
 - des exercices (à faire « à la maison »)
 - Continuer les TDs en salle machine et en groupe pour renforcer les acquis.

Pourquoi avoir choisi la classe inversée ?

- Remplacer le cours en amphi par
 - des support et des ressources (à lire « à la maison »)
 - des exercices (à faire « à la maison »)
- Continuer les TDs en salle machine et en groupe pour renforcer les acquis.

Ce sont bien les attributs de la classe inversée

La problématique (en vrac)

- Quels sont les bénéfices et les risques pour les étudiants ?

- Quels sont les bénéfices et les risques pour les étudiants ?
- Quels sont les bénéfices et les risques pour l'enseignant ?

- Quels sont les bénéfices et les risques pour les étudiants ?
- Quels sont les bénéfices et les risques pour l'enseignant ?
- Y-a-t-il des supports de cours auto-suffisants

- Quels sont les bénéfices et les risques pour les étudiants ?
- Quels sont les bénéfices et les risques pour l'enseignant ?
- Y-a-t-il des supports de cours auto-suffisants
- Comment s'assurer de la bonne compréhension du cours par les étudiants avant l'accès au TD ?

La problématique (en vrac)

- Quels sont les bénéfices et les risques pour les étudiants ?
 - Gain en autonomie
 - Risque de se laisser déborder
- Quels sont les bénéfices et les risques pour l'enseignant ?
- Y-a-t-il des supports de cours auto-suffisants
- Comment s'assurer de la bonne compréhension du cours par les étudiants avant l'accès au TD ?

La problématique (en vrac)

- Quels sont les bénéfices et les risques pour les étudiants ?
 - Gain en autonomie
 - Risque de se laisser déborder => Suivi
- Quels sont les bénéfices et les risques pour l'enseignant ?
- Y-a-t-il des supports de cours auto-suffisants
- Comment s'assurer de la bonne compréhension du cours par les étudiants avant l'accès au TD ?

La problématique (en vrac)

- Quels sont les bénéfices et les risques pour les étudiants ?
 - Gain en autonomie
 - Risque de se laisser déborder => Suivi
- Quels sont les bénéfices et les risques pour l'enseignant ?
 - Éviter ce cours magistral peu adapté
 - Risque de perdre le contact avec les étudiants
- Y-a-t-il des supports de cours auto-suffisants
- Comment s'assurer de la bonne compréhension du cours par les étudiants avant l'accès au TD ?

La problématique (en vrac)

- Quels sont les bénéfices et les risques pour les étudiants ?
 - Gain en autonomie
 - Risque de se laisser déborder => Suivi
- Quels sont les bénéfices et les risques pour l'enseignant ?
 - Éviter ce cours magistral peu adapté
 - Risque de perdre le contact avec les étudiants => On verra
- Y-a-t-il des supports de cours auto-suffisants
- Comment s'assurer de la bonne compréhension du cours par les étudiants avant l'accès au TD ?

La problématique (en vrac)

- Quels sont les bénéfices et les risques pour les étudiants ?
 - Gain en autonomie
 - Risque de se laisser déborder => Suivi
- Quels sont les bénéfices et les risques pour l'enseignant ?
 - Éviter ce cours magistral peu adapté
 - Risque de perdre le contact avec les étudiants => On verra
- Y-a-t-il des supports de cours auto-suffisants
- Comment s'assurer de la bonne compréhension du cours par les étudiants avant l'accès au TD ?

La problématique (en vrac)

- Quels sont les bénéfices et les risques pour les étudiants ?
 - Gain en autonomie
 - Risque de se laisser déborder => Suivi
- Quels sont les bénéfices et les risques pour l'enseignant ?
 - Éviter ce cours magistral peu adapté
 - Risque de perdre le contact avec les étudiants => On verra
- Les supports de cours sont auto-suffisants
- Comment s'assurer de la bonne compréhension du cours par les étudiants avant l'accès au TD ?

La problématique (en vrac)

- Quels sont les bénéfices et les risques pour les étudiants ?
 - Gain en autonomie
 - Risque de se laisser déborder => Suivi
- Quels sont les bénéfices et les risques pour l'enseignant ?
 - Éviter ce cours magistral peu adapté
 - Risque de perdre le contact avec les étudiants => On verra
- Les supports de cours sont auto-suffisants
- Comment s'assurer de la bonne compréhension du cours par les étudiants avant l'accès au TD ?

Pierre d'achoppement

Comment s'assurer de la bonne compréhension du cours avant l'accès au TD ?

Comment s'assurer de la bonne compréhension du cours avant l'accès au TD ?

- Comment « inciter » les étudiants à travailler chez eux ?
- Comment les suivre ?
- Comment m'assurer qu'ils ont compris ?
- Comment peuvent-ils s'assurer qu'ils ont compris ?

Pierre d'achoppement

Comment s'assurer de la bonne compréhension du cours avant l'accès au TD ?

- Comment « inciter » les étudiants à travailler chez eux ?
- Comment les suivre ?
- Comment m'assurer qu'ils ont compris ?
- Comment peuvent-ils s'assurer qu'ils ont compris ?

Auto-évaluation

Il faut un outil d'auto-évaluation et de suivi.

Pierre d'achoppement

Comment s'assurer de la bonne compréhension du cours avant l'accès au TD ?

- Comment « inciter » les étudiants à travailler chez eux ?
- Comment les suivre ?
- Comment m'assurer qu'ils ont compris ?
- Comment peuvent-ils s'assurer qu'ils ont compris ?

Auto-évaluation

Il faut un outil d'auto-évaluation et de suivi.

Pas trouvé tout de suite

Pierre d'achoppement

Comment s'assurer de la bonne compréhension du cours avant l'accès au TD ?

- Comment « inciter » les étudiants à travailler chez eux ?
- Comment les suivre ?
- Comment m'assurer qu'ils ont compris ?
- Comment peuvent-ils s'assurer qu'ils ont compris ?

Auto-évaluation

Il faut un outil d'auto-évaluation et de suivi.

Pas trouvé tout de suite

Le cours a été fait de façon « traditionnelle » en 2018-2019

Par manque d'un outil d'auto-évaluation et de suivi.

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.

VPL, the Virtual Programming lab for Moodle

`https://vpl.dis.ulpgc.es/`

VPL is the easy way to manage programming assignments in Moodle.

Its features of editing, running and evaluation of programs makes learning process for students, and the evaluation task for teachers, easier than ever.

Un exemple classique

The screenshot shows a web browser window with the address bar displaying `moodle.casine.org/mod/vpl/forms/edit.php?id=22834&userid=10826`. The browser's toolbar includes various icons for navigation and editing. The main content area is divided into two panels. The left panel, titled "IntegerTriplet.java", displays the following Java code:

```
1 package number;
2
3 /**
4  * A class representing a triplet of three integers
5  *
6  * @author hadrien cambazard
7  */
8 public class IntegerTriplet {
9
10     private final int a, b, c;
11
12     /**
13      * Constructor using three values
14      */
15     public IntegerTriplet(int a, int b, int c) {
16         this.a = a;
17         this.b = b;
18         this.c = c;
19     }
20
21     /**
22      * @return the sum of the elements in the set
23      */
24     public int sum() {
25         return 0;
26     }
27     // TODO
28
29     /**
30      * @return the average value of the set
31      */
```

The right panel shows the test results for the code. It includes a section for "Note proposée: 0 / 20" and a section for "Commentaires". The test results are as follows:

- Tests results**
- number.IntegerTripletTest.testSum (1/3)**
Incorrect program result
Message : java.lang.AssertionError: expected:<-1007120394> but was:<0>
- number.IntegerTripletTest.testAverage (2/3)**
Incorrect program result
Message : java.lang.AssertionError: expected:<-6.699248276666666E8> but was:<0>
- number.IntegerTripletTest.testConcatenate (3/3)**
Incorrect program result
Message : java.lang.AssertionError: expected:<5848488731181025995-200352> but was:<0>

Un exemple classique

Java L3 (UHA) TroisEntiers Éditer - Google Chrome

moodle.caseine.org/mod/vpl/forms/edit.php?id=22834&userid=10826

IntegerTriplet.java

```
16         this.a = a;
17         this.b = b;
18         this.c = c;
19     }
20
21     /**
22      * @return the sum of the elements in the set
23      */
24     public int sum() {
25         return 0;
26         // TODO
27     }
28
29     /**
30      * @return the average value of the set
31      */
32     public double average() {
33         return 0;
34         // TODO
35     }
36
37     /**
38      * @return the string obtained by concatenating th
39      */
40     public String concatenate() {
41         return null;
42         // TODO
43     }
44 }
45
```

► Note proposée: 0 / 20

▼ Commentaires

Tests results

number.IntegerTripletTest.testSum (1/3)
Incorrect program result
Message : java.lang.AssertionError: expected:<-1007120394> but was:<0>

number.IntegerTripletTest.testAverage (2/3)
Incorrect program result
Message : java.lang.AssertionError: expected:<-6.699248276666666E8> but w

number.IntegerTripletTest.testConcatenate (3/3)
Incorrect program result
Message : java.lang.AssertionError: expected:<5848488731181025995-200352

Un exemple classique

Java L3 (UHA) TroisEntiers Éditer - Google Chrome

moodle.caseine.org/mod/vpl/forms/edit.php?id=22834&userid=10826

IntegerTriplet.java

```
16     this.a = a;
17     this.b = b;
18     this.c = c;
19 }
20
21 /**
22  * @return the sum of the elements in the set
23  */
24 public int sum() {
25     return a + b + c;
26 // TODO
27 }
28
29 /**
30  * @return the average value of the set
31  */
32 public double average() {
33     return 0;
34 // TODO
35 }
36
37 /**
38  * @return the string obtained by concatenating th
39  */
40 public String concatenate() {
41     return null;
42 // TODO
43 }
44 }
45 }
```

► Note proposée: 0 / 20

▼ Commentaires

Tests results

number.IntegerTripletTest.testSum (1/3)
Incorrect program result
Message : java.lang.AssertionError: expected:<-1007120394> but was:<0>

number.IntegerTripletTest.testAverage (2/3)
Incorrect program result
Message : java.lang.AssertionError: expected:<-6.699248276666666E8> but was:<0>

number.IntegerTripletTest.testConcatenate (3/3)
Incorrect program result
Message : java.lang.AssertionError: expected:<5848488731181025995-200352>

► Exécution

Un exemple classique

Java L3 (UHA) TroisEntiers Éditer - Google Chrome

moodle.caseine.org/mod/vpl/forms/edit.php?id=22834&userid=10826

IntegerTriplet.java

```
16         this.a = a;
17         this.b = b;
18         this.c = c;
19     }
20
21     /**
22      * @return the sum of the elements in the set
23      */
24     public int sum() {
25         return a + b + c;
26     }
27
28
29     /**
30      * @return the average value of the set
31      */
32     public double average() {
33         return 0;
34     }
35
36
37     /**
38      * @return the string obtained by concatenating th
39      */
40     public String concatenate() {
41         return null;
42     }
43
44 }
45
```

► Note proposée: 5 / 20

▼ Commentaires

Tests results

number.IntegerTripletTest.testSum (1/3)

number.IntegerTripletTest.testAverage (2/3)

Incorrect program result

Message : java.lang.AssertionError: expected:<1.7352594266666666E8> but v

number.IntegerTripletTest.testConcatenate (3/3)

Incorrect program result

Message : java.lang.AssertionError: expected:<-769411308-987732418-15588<null>

Exécution

Un exemple classique

Java L3 (UHA) TroisEntiers Éditer - Google Chrome

moodle.caseine.org/mod/vpl/forms/edit.php?id=22834&userid=10826

IntegerTriplet.java

```
16     this.a = a;
17     this.b = b;
18     this.c = c;
19 }
20
21 /**
22  * @return the sum of the elements in the set
23  */
24 public int sum() {
25     return a + b + c;
26     // TODO
27 }
28
29 /**
30  * @return the average value of the set
31  */
32 public double average() {
33     return sum()/3;
34     // TODO
35 }
36
37 /**
38  * @return the string obtained by concatenating th
39  */
40 public String concatenate() {
41     return null;
42     // TODO
43 }
44 }
45 }
```

► Note proposée: 5 / 20

▼ Commentaires

Tests results

number.IntegerTripletTest.testSum (1/3)
Incorrect program result
Message : java.lang.AssertionError: expected:<-1.6234772333333334E7> but

number.IntegerTripletTest.testAverage (2/3)
Incorrect program result
Message : java.lang.AssertionError: expected:<-511935819-1557852427-6681

number.IntegerTripletTest.testConcatenate (3/3)
Incorrect program result
Message : java.lang.AssertionError: expected:<-511935819-1557852427-6681

Exécution

Realisation d'exercices de programmation auto-évalués sur une pl

Un exemple classique

Java L3 (UHA) TroisEntiers Éditer - Google Chrome

moodle.caseine.org/mod/vpl/forms/edit.php?id=22834&userid=10826

IntegerTriplet.java

```
16         this.a = a;
17         this.b = b;
18         this.c = c;
19     }
20
21     /**
22      * @return the sum of the elements in the set
23      */
24     public int sum() {
25         return a + b + c;
26     }
27
28
29     /**
30      * @return the average value of the set
31      */
32     public double average() {
33         return sum()/3.0;
34     }
35
36
37     /**
38      * @return the string obtained by concatenating th
39      */
40     public String concatenate() {
41         return null;
42     }
43
44 }
```

► Note proposée: 5 / 20

▼ Commentaires

Tests results

number.IntegerTripletTest.testSum (1/3)
Incorrect program result
Message : java.lang.AssertionError: expected:<-1.623477233333334E7> but

number.IntegerTripletTest.testAverage (2/3)
Incorrect program result
Message : java.lang.AssertionError: expected:<-511935819-1557852427-6681

number.IntegerTripletTest.testConcatenate (3/3)
Incorrect program result
Message : java.lang.AssertionError: expected:<-511935819-1557852427-6681

Exécution

Réalisation d'exercices de programmation auto-évalués sur une pl

Un exemple classique

Java L3 (UHA) TroisEntiers Éditer - Google Chrome

moodle.caseine.org/mod/vpl/forms/edit.php?id=22834&userid=10826

IntegerTriplet.java

```
16     this.a = a;
17     this.b = b;
18     this.c = c;
19 }
20
21 /**
22  * @return the sum of the elements in the set
23  */
24 public int sum() {
25     return a + b + c;
26 // TODO
27 }
28
29 /**
30  * @return the average value of the set
31  */
32 public double average() {
33     return sum()/3.0;
34 // TODO
35 }
36
37 /**
38  * @return the string obtained by concatenating th
39  */
40 public String concatenate() {
41     return null;
42 // TODO
43 }
44 }
45 }
```

► Note proposée: 12 / 20

▼ Commentaires

Tests results

- number.IntegerTripletTest.testSum (1/3)
- number.IntegerTripletTest.testAverage (2/3)
- number.IntegerTripletTest.testConcatenate (3/3)

Incorrect program result
Message : java.lang.AssertionError: expected:<86991145-145952888442282>

► Exécution

Realisation d'exercices de programmation auto-évalués sur une pl

Un exemple classique

Java L3 (UHA) TroisEntiers Éditer - Google Chrome

moodle.caseine.org/mod/vpl/forms/edit.php?id=22834&userid=10826

IntegerTriplet.java

```
16     this.a = a;
17     this.b = b;
18     this.c = c;
19 }
20
21 /**
22  * @return the sum of the elements in the set
23  */
24 public int sum() {
25     return a + b + c;
26     // TODO
27 }
28
29 /**
30  * @return the average value of the set
31  */
32 public double average() {
33     return sum()/3.0;
34     // TODO
35 }
36
37 /**
38  * @return the string obtained by concatenating th
39  */
40 public String concatenate() {
41     return "" + a + b + c;
42     // TODO
43 }
44 }
45 }
```

► Note proposée: 12 / 20

▼ Commentaires

Tests results

- number.IntegerTripletTest.testSum (1/3)
- number.IntegerTripletTest.testAverage (2/3)
- number.IntegerTripletTest.testConcatenate (3/3)

Incorrect program result
Message : java.lang.AssertionError: expected:<86991145-145952888442282>

► Exécution

Un exemple classique

Java L3 (UHA) TroisEntiers Éditer - Google Chrome

moodle.caseine.org/mod/vpl/forms/edit.php?id=22834&userid=10826

IntegerTriplet.java

```
16     this.a = a;
17     this.b = b;
18     this.c = c;
19 }
20
21 /**
22  * @return the sum of the elements in the set
23  */
24 public int sum() {
25     return a + b + c;
26     // TODO
27 }
28
29 /**
30  * @return the average value of the set
31  */
32 public double average() {
33     return sum()/3.0;
34     // TODO
35 }
36
37 /**
38  * @return the string obtained by concatenating th
39  */
40 public String concatenate() {
41     return "" + a + b + c;
42     // TODO
43 }
44 }
45 }
```

► Note proposée: 20 / 20

▼ Commentaires

Tests results

- number.IntegerTripletTest.testSum (1/3)
- number.IntegerTripletTest.testAverage (2/3)
- number.IntegerTripletTest.testConcatenate (3/3)

Exécution

Realisation d'exercices de programmation auto-évalués sur une pl

Une classe à écrire complètement

Java L3 (UHA) Programme Minimum Éditeur - Google Chrome

moodle.caseine.org/mod/vpl/forms/edit.php?id=16166&userid=10826

ClasseAutonome.java

```
1 /*
2  * @author : Yvan Maillot (yvan.maillot@uha.fr)
3  */
4 /**
5  * @author yvan
6  */
7 // TODO Écrire la classe autonome minimale.
8 //
9 //
10
```

Note proposée: 0 / 20

Commentaires

Tests results

GeneratedClasseAutonomeTest.p000001000_checkClassClasseAutonome (1/4)

Incorrect program result
Message : java.lang.AssertionError: Class ClasseAutonome not found

GeneratedClasseAutonomeTest.p000002000_checkMethodClasseAutonome (2/4)

Incorrect program result
Message : java.lang.AssertionError: Inexisting class java.lang.ClassNotFoundException: java.lang.ClasseAutonome

ClasseAutonomeTest.testMain (3/4)

Incorrect program result
Message : java.lang.AssertionError: Bien relire le cours et/ou l'ensemble des exercices

ClasseAutonomeTest.testOnlyOneMethod (4/4)

Incorrect program result
Message : java.lang.AssertionError: Bien relire le cours et/ou l'ensemble des exercices

Exécution

Réalisation d'exercices de programmation auto-évalués sur une plateforme

Une classe à écrire complètement

The screenshot shows a web-based IDE interface. The top bar indicates the editor is 'Java L3 (UHA) Programme Minimum Éditeur - Google Chrome'. The browser address bar shows the URL 'moodle.caseine.org/mod/vpl/forms/edit.php?id=16166&userid=10826'. The IDE has a toolbar with various icons, including a checkmark with the number '12'. The main editor area displays the code for 'ClasseAutonome.java':

```
1 /*
2  * @author : Yvan Maillot (yvan.maillot@uha.fr)
3  */
4 /**
5  * @author yvan
6  */
7 // TODO Écrire la classe autonome minimale.
8 //
9 //
10 public class ClasseAutonome {
11
12 }
```

On the right side, the 'Note proposée: 0 / 20' and 'Commentaires' section shows the test results:

- Tests results**
- GeneratedClasseAutonomeTest.p000001000_checkClassClasseAutonome (1/4)**
Incorrect program result
Message : java.lang.AssertionError: Class ClasseAutonome not found
- GeneratedClasseAutonomeTest.p000002000_checkMethodClasseAutonome (2/4)**
Incorrect program result
Message : java.lang.AssertionError: Inexisting class java.lang.ClassNotFoundException: java.lang.ClassClasseAutonomeTest.testMain (3/4)
- ClasseAutonomeTest.testOnlyOneMethod (4/4)**
Incorrect program result
Message : java.lang.AssertionError: Bien relire le cours et/ou l'ensemble des exercices de programmation auto-évalués sur une plateforme d'exercices

The 'Exécution' section is also visible at the bottom of the right panel.

Réalisation d'exercices de programmation auto-évalués sur une plateforme d'exercices

Une classe à écrire complètement

Java L3 (UHA) Programme Minimum Éditeur - Google Chrome

moodle.caseine.org/mod/vpl/forms/edit.php?id=16166&userid=10826

ClasseAutonome.java

```
1 /*
2  * @author : Yvan Maillot (yvan.maillot@uha.fr)
3  */
4 /**
5  * @author yvan
6  */
7 // TODO Écrire la classe autonome minimale.
8 //
9 //
10 public class ClasseAutonome {
11
12 }
```

► Note proposée: 3,33 / 20

▼ Commentaires

Tests results

GeneratedClasseAutonomeTest.p000001000 checkClassClasseAutonome (1/4)

GeneratedClasseAutonomeTest.p000002000 checkMethodClasseAutonome (2/4)

Incorrect program result
Message : java.lang.AssertionError: Fix votre classe. Elle n'est pas compilable

ClasseAutonomeTest.testMain (3/4)

Incorrect program result
Message : java.lang.AssertionError: Bien relire le cours et/ou les exercices

ClasseAutonomeTest.testOnlyOneMethod (4/4)

Incorrect program result
Message : java.lang.AssertionError: Bien relire le cours et/ou les exercices

► Exécution

Réalisation d'exercices de programmation auto-évalués sur une plateforme

Une classe à écrire complètement

Java L3 (UHA) Programme Minimum Éditeur - Google Chrome

moodle.caseine.org/mod/vpl/forms/edit.php?id=16166&userid=10826

ClasseAutonome.java

```
1 /*
2  * @author : Yvan Maillot (yvan.maillot@uha.fr)
3  */
4 /**
5  * @author yvan
6  */
7 // TODO Écrire la classe autonome minimale.
8 //
9 //
10 public class ClasseAutonome {
11     public static void main(String[] arg) {
12
13     }
14 }
```

► Note proposée: 3,33 / 20

▼ Commentaires

Tests results

GeneratedClasseAutonomeTest.p000001000 checkClassClasseAutonome (1/4)

GeneratedClasseAutonomeTest.p000002000 checkMethodClasseAutonome (2/4)

Incorrect program result
Message : java.lang.AssertionError: Fix votre classe. Elle n'est pas compilable

ClasseAutonomeTest.testMain (3/4)

Incorrect program result
Message : java.lang.AssertionError: Bien relire le cours et/ou les exercices

ClasseAutonomeTest.testOnlyOneMethod (4/4)

Incorrect program result
Message : java.lang.AssertionError: Bien relire le cours et/ou les exercices

► Exécution

Réalisation d'exercices de programmation auto-évalués sur une plateforme en ligne

Une classe à écrire complètement

Java L3 (UHA) Programme Minimum Éditeur - Google Chrome

moodle.caseine.org/mod/vpl/forms/edit.php?id=16166&userid=10826

ClasseAutonome.java

```
1 /*
2  * @author : Yvan Maillot (yvan.maillot@uha.fr)
3  */
4 /**
5  * @author yvan
6  */
7 // TODO Écrire la classe autonome minimale.
8 //
9 //
10 public class ClasseAutonome {
11     public static void main(String[] arg) {
12
13     }
14 }
```

► Note proposée: 20 / 20

▼ Commentaires

Tests results

- GeneratedClasseAutonomeTest.p000001000 checkClassClasseAutonome (1/4)
- GeneratedClasseAutonomeTest.p000002000 checkMethodClasseAutonome (2/4)
- ClasseAutonomeTest.testMain (3/4)
- ClasseAutonomeTest.testOnlyOneMethod (4/4)

► Exécution

► Description:

Plusieurs classes et nombreux points d'évaluation

Java L3 (UHA) Des triangles p

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

Point.java

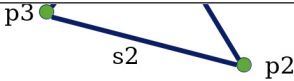
```
1- /*
2-  * @author : Y
3-  */
4- package edu.uh
5-
6- public class P
7-     // TODO 1.
8-
9-     // TODO 1.
10-
11-     // TODO 1.
12-
13-     // TODO 1.
14-
15-     // TODO 1.
16-
17-     // TODO 1.
18-
19-     // TODO 1.
20-
21- }
22
```

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 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 devrez implémenter en totalité.

Triangle

+ Triangle(in p1: Point, in p2: Point, in p3: Point) C0

+ getP1(): Point

+ getP2(): Point

+ getP3(): Point

+ setP1(in p1: Point)

+ setP2(in p1: Point)

+ setP3(in p1: Point)

+ getPerimetre(): double

+ getSurface(): double

+ getBaryCentre(): Point

+ toString(): String

- s1

1

- s2

1

- s3

1

Segment

+ Segment(in p1: Point, in p2: Point) C0

+ getP1(): Point

+ getP2(): Point

+ setP1(in p: Point)

+ setP2(in p: Point)

+ getLongueur(): double

+ toString(): String

- p1

1

- p2

1

Point

- x : double

- y : double

+ Point(in x: double, in y: double) C0

+ setX(in x: double)

+ setY(in y: double)

+ getX(): double

+ getY(): double

+ toString(): String

Réalisation d'exercices de programmation auto-évaluée sur une pl

Yvan Maillot (yvan.maillot@uha.fr)

9/23

Plusieurs classes et nombreux points d'évaluation



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

Java L3 (UHA) Des triangles p...
moodle.caseine.org/mod/vpl/forms/edit.php?id=16986&userid=10826

Main.java Segment.java Point.java Triangle.java

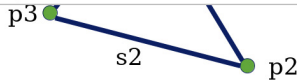
```
1- /*  
2-  * @author : Yvan Maillot (yvan.maillot@uha.fr)  
3-  */  
4- package edu.uha.miage;  
5-  
6- public class Point {  
7-     // TODO 1.01. Déclarer x et y, les coordonnées cartésiennes de ce point  
8-     private double x, y;  
9-     // TODO 1.02. Écrire le constructeur aux coordonnées 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



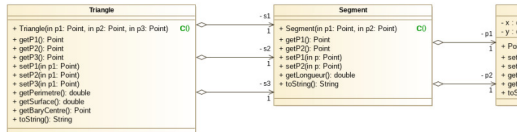
On pourrait traduire cette définition par plusieurs modélisations orientées objets en Java, par exemple en mettant en attribut de la classe Triangle 3 points. On aurait un triangle défini par p1, p2 et p3 en vert.

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 ses côtés s1, s2 et s3.

- 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.



Point

Réalisation d'exercices de programmation auto-évaluée sur une plateforme

Plusieurs classes et nombreux points d'évaluation

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

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

Main java U Segment.java U Point.java U Triangle.java U

Note proposée: 0,32 / 20

Commentaires

```
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 }
```

Incorrect program result
Message : java.lang.AssertionError: Fix setP1
[edu.uha.miage.GeneratedTriangleTest.p000033000_checkMethodTrianglesetP2Point \(41/57\)](#)
Incorrect program result
Message : java.lang.AssertionError: Fix setP2
[edu.uha.miage.GeneratedTriangleTest.p000033000_checkMethodTrianglesetP2Point \(42/57\)](#)
Incorrect program result
Message : java.lang.AssertionError: Fix setP3
[edu.uha.miage.GeneratedTriangleTest.p000034000_checkMethodTrianglesetPerimetre \(43/57\)](#)
Incorrect program result
Message : java.lang.AssertionError: Fix Signature of Triangle.getPerimetre()
[edu.uha.miage.GeneratedTriangleTest.p000035000_testMethodTrianglesetPerimetre \(44/57\)](#)
Incorrect program result
Message : java.lang.AssertionError: Fix Triangle.getPerimetre()
[edu.uha.miage.GeneratedTriangleTest.p000035000_checkMethodTrianglesetSurface \(45/57\)](#)
Incorrect program result
Message : java.lang.AssertionError: Fix Signature of Triangle.getSurface()
[edu.uha.miage.GeneratedTriangleTest.p000037000_testMethodTrianglesetSurface \(46/57\)](#)
Incorrect program result
Message : java.lang.AssertionError: Fix Triangle.getSurface()
[edu.uha.miage.GeneratedTriangleTest.p000038000_checkMethodTrianglesetBaryCentre \(47/57\)](#)
Incorrect program result
Message : java.lang.AssertionError: Fix Signature of Triangle.getBaryCentre()
[edu.uha.miage.GeneratedTriangleTest.p000040000_checkMethodTrianglesetToString \(48/57\)](#)
Incorrect program result
Message : java.lang.AssertionError: Fix toString
[edu.uha.miage.TriangleTest.testConstructor \(49/57\)](#)
Incorrect program result
Message : java.lang.AssertionError: Constructeur Point(double x, double y) absent ou mal signe
[edu.uha.miage.TriangleTest.testSetP1 \(50/57\)](#)
Incorrect program result
Message : java.lang.AssertionError: Constructeur Point(double x, double y) absent ou mal signe
[edu.uha.miage.TriangleTest.testSetP2 \(51/57\)](#)
Incorrect program result
Message : java.lang.AssertionError: Constructeur Point(double x, double y) absent ou mal signe
[edu.uha.miage.TriangleTest.testSetP3 \(52/57\)](#)
Incorrect program result
Message : java.lang.AssertionError: Constructeur Point(double x, double y) absent ou mal signe
[edu.uha.miage.TriangleTest.testGetP1 \(53/57\)](#)
Incorrect program result
Message : java.lang.AssertionError: Constructeur Point(double x, double y) absent ou mal signe
[edu.uha.miage.TriangleTest.testGetP2 \(54/57\)](#)
Incorrect program result
Message : java.lang.AssertionError: Constructeur Point(double x, double y) absent ou mal signe
[edu.uha.miage.TriangleTest.testGetP3 \(55/57\)](#)
Incorrect program result
Message : java.lang.AssertionError: Constructeur Point(double x, double y) absent ou mal signe
[edu.uha.miage.TriangleTest.testToString \(56/57\)](#)
Incorrect program result
Message : java.lang.AssertionError: Constructeur Point(double x, double y) absent ou mal signe
[edu.uha.miage.TriangleTest.testToBarycentre \(57/57\)](#)
Incorrect program result
Message : java.lang.AssertionError: Constructeur Point(double x, double y) absent ou mal signe

Exécution

Réalisation d'exercices de programmation auto-évalués sur une pl

Plusieurs classes et nombreux points d'évaluation

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

Java L3 (UHA) Des triangles plutôt classes

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

Main java U Segments java U Point.java U Triangle.java U

Note proposée: 2,88 / 20

Commentaires

Tests results

edu.uha.miage.GeneratedPointTest.p000001000 checkPointFieldX (1/57)
edu.uha.miage.GeneratedPointTest.p000001000 checkPointFieldY (2/57)
edu.uha.miage.GeneratedPointTest.p000002000 checkConstructorPointDoubledouble (4/57)
edu.uha.miage.GeneratedPointTest.p000003000 checkMethodPointGetX (4/57)
edu.uha.miage.GeneratedPointTest.p000003000 checkMethodPointGetY (5/57)
edu.uha.miage.GeneratedPointTest.p000004000 checkMethodPointSetXdouble (6/57)
edu.uha.miage.GeneratedPointTest.p000004000 checkMethodPointSetYdouble (7/57)
edu.uha.miage.GeneratedPointTest.p000005000 checkMethodPointToString (10/57)
edu.uha.miage.GeneratedPointTest.p000006000 checkGetterPointX (9/57)
edu.uha.miage.GeneratedPointTest.p000006000 checkGetterPointY (10/57)
edu.uha.miage.GeneratedPointTest.p000007000 checkSetterPointX (11/57)
edu.uha.miage.GeneratedPointTest.p000007000 checkSetterPointY (12/57)
edu.uha.miage.PointTest.fastConstructor (13/57)
edu.uha.miage.PointTest.toString (14/57)
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 checkConstructorSegmentPoint (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 checkMethodSegmentSetP1 (20/57)
Incorrect program result
Message : java.lang.AssertionError: Fix setP1
edu.uha.miage.GeneratedSegmentTest.p000023000 checkMethodSegmentSetP2 (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
edu.uha.miage.GeneratedSegmentTest.p000027000 checkSetterSegmentP1 (27/57)
Incorrect program result
Message : java.lang.AssertionError: Fix setP1

Exécution

```
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("(%f, %f)", x, y);
35-     }
36- }
37- }
```

Plusieurs classes et nombreux points d'évaluation

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

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

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

Segment.java | Point.java | Triangle.java

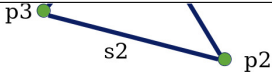
```
1- /*
2-  * @author : Yvan Maillot (yvan.maillot@uha.fr)
3-  */
4- package edu.uha.mlage;
5-
6- /**
7-  * @author yvan
8-  */
9- public class Segment {
10-
11-     private Point p1, p2;
12-
13-     public Segment(Point p1, Point p2) {
14-         this.p1 = p1;
15-         this.p2 = p2;
16-     }
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()) * (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-
```

Note proposée: 2,88 / 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 sommets 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 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.

Triangle

+ Triangle(n p1: Point, in p2: Point, in p3: Point) C0

+ getP1(): Point

+ getP2(): Point

+ getP3(): Point

+ setP1(in p1: Point)

+ setP2(in p1: Point)

+ setP3(in p1: Point)

+ getPerimetre(): double

+ getSurface(): double

+ getBaryCentre(): Point

+ toString(): String

Segment

+ Segment(n p1: Point, in p2: Point) C0

+ getP1(): Point

+ getP2(): Point

+ setP1(in p1: Point)

+ setP2(in p1: Point)

+ getLongueur(): double

+ toString(): String

Point

- x : double

- y : double

Yvan Maillot (yvan.maillot@uha.fr)

9/23

Plusieurs classes et 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/aml/forms/edit.php?id=16986&userid=10826

Main java U Segment java U Point java U Triangle java U

```
1- /*
2-  * @author : Yvan Maillot (yvan.maillot@uha.fr)
3-  */
4- package edu.uha.mlage;
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()) * (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- }
```

► Note proposée: 10,08 / 20

▼ Commentaires

Tests results

edu.uha.mlage.GeneratedPointTest.p000001000 checkPointFieldX (1/57)
edu.uha.mlage.GeneratedPointTest.p000001000 checkPointFieldY (2/57)
edu.uha.mlage.GeneratedPointTest.p000002000 checkConstructorPointdoubledouble (4/57)
edu.uha.mlage.GeneratedPointTest.p000003000 checkMethodPointgetX (4/57)
edu.uha.mlage.GeneratedPointTest.p000003000 checkMethodPointgetY (5/57)
edu.uha.mlage.GeneratedPointTest.p000004000 checkMethodPointsetXdouble (6/57)
edu.uha.mlage.GeneratedPointTest.p000004000 checkMethodPointsetYdouble (7/57)
edu.uha.mlage.GeneratedPointTest.p000005000 checkMethodPointtoString (10/57)
edu.uha.mlage.GeneratedPointTest.p000006000 checkGetterPointX (9/57)
edu.uha.mlage.GeneratedPointTest.p000006000 checkGetterPointY (10/57)
edu.uha.mlage.GeneratedPointTest.p000007000 checkSetterPointX (11/57)
edu.uha.mlage.GeneratedPointTest.p000007000 checkSetterPointY (12/57)
edu.uha.mlage.PointTest.fastConstructor (13/57)
edu.uha.mlage.PointTest.toString (14/57)
edu.uha.mlage.GeneratedSegmentTest.p000020000 checkSegmentFieldP1 (15/57)
edu.uha.mlage.GeneratedSegmentTest.p000020000 checkSegmentFieldP2 (16/57)
edu.uha.mlage.GeneratedSegmentTest.p000021000 checkConstructorSegmentPoint (17/57)
edu.uha.mlage.GeneratedSegmentTest.p000022000 checkMethodSegmentgetP1 (18/57)
edu.uha.mlage.GeneratedSegmentTest.p000022000 checkMethodSegmentgetP2 (19/57)
edu.uha.mlage.GeneratedSegmentTest.p000023000 checkMethodSegmentsetP1Point (20/57)
edu.uha.mlage.GeneratedSegmentTest.p000023000 checkMethodSegmentsetP2Point (21/57)
edu.uha.mlage.GeneratedSegmentTest.p000024000 checkMethodSegmentlength (22/57)
edu.uha.mlage.GeneratedSegmentTest.p000025000 checkMethodSegmentprojectionLengthOnX (23/57)
edu.uha.mlage.GeneratedSegmentTest.p000025000 checkMethodSegmenttoString (24/57)
edu.uha.mlage.GeneratedSegmentTest.p000025000 checkGetterSegmentP1 (25/57)
edu.uha.mlage.GeneratedSegmentTest.p000026000 checkGetterSegmentP2 (26/57)
edu.uha.mlage.GeneratedSegmentTest.p000027000 checkSetterSegmentP1 (27/57)
edu.uha.mlage.GeneratedSegmentTest.p000027000 checkSetterSegmentP2 (28/57)
edu.uha.mlage.GeneratedSegmentTest.p000028000 testMethodSegmentlength (29/57)
edu.uha.mlage.GeneratedSegmentTest.p000029000 testMethodSegmentprojectionLengthOnX (30/57)
edu.uha.mlage.SegmentTest.testToString (31/57)
edu.uha.mlage.GeneratedTriangleTest.p000030000 checkTriangleFieldS1 (33/57)
Incorrect program result
Message : java.lang.AssertionError:
edu.uha.mlage.GeneratedTriangleTest.p000030000 checkTriangleFieldS2 (34/57)
Incorrect program result
Message : java.lang.AssertionError:
edu.uha.mlage.GeneratedTriangleTest.p000030000 checkTriangleFieldS3 (35/57)
Incorrect program result
Message : java.lang.AssertionError:
edu.uha.mlage.GeneratedTriangleTest.p000031000 checkConstructorTrianglePointPoint (36/57)
Incorrect program result
Message : java.lang.AssertionError: Constructor of edu.uha.mlage.Triangle
edu.uha.mlage.GeneratedTriangleTest.p000032000 checkMethodTrianglegetP1 (37/57)
Incorrect program result
Message : java.lang.AssertionError: Fix getP1
edu.uha.mlage.GeneratedTriangleTest.p000032000 checkMethodTrianglegetP2 (38/57)
Incorrect program result
Message : java.lang.AssertionError: Fix getP2
edu.uha.mlage.GeneratedTriangleTest.p000032000 checkMethodTrianglegetP3 (39/57)
Incorrect program result
Message : java.lang.AssertionError: Fix getP3

► Execution

Réalisation d'exercices de programmation auto-évalués sur une pl

Plusieurs classes et nombreux points d'évaluation

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

Java L3 (UHA) Des triangles p x +

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

Main.java | Segment.java | Point.java | Triangle.java

```
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 public // @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())
61 }
62
63 @Override
64 public String toString() {
65     return "c" + s1 + ", " + s2 + ", " + s3 + ">";
66 }
67
68 }
```

► Note proposée: 20 / 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 checkConstructorPointDoubleDouble (6/57)
edu.uha.miage.GeneratedPointTest.p00003000 checkMethodPointGetX (4/57)
edu.uha.miage.GeneratedPointTest.p00003000 checkMethodPointGetY (5/57)
edu.uha.miage.GeneratedPointTest.p00004000 checkMethodPointGetXDouble (6/57)
edu.uha.miage.GeneratedPointTest.p00004000 checkMethodPointGetYDouble (7/57)
edu.uha.miage.GeneratedPointTest.p00005000 checkMethodPointToString (10/57)
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.PointTest.testToString (14/57)
edu.uha.miage.GeneratedSegmentTest.p00002000 checkSegmentFieldP1 (15/57)
edu.uha.miage.GeneratedSegmentTest.p00002000 checkSegmentFieldP2 (16/57)
edu.uha.miage.GeneratedSegmentTest.p00002000 checkConstructorSegmentPointPoint (17/57)
edu.uha.miage.GeneratedSegmentTest.p00002000 checkMethodSegmentGetP1 (18/57)
edu.uha.miage.GeneratedSegmentTest.p00002000 checkMethodSegmentGetP2 (19/57)
edu.uha.miage.GeneratedSegmentTest.p00002000 checkMethodSegmentGetP1 (20/57)
edu.uha.miage.GeneratedSegmentTest.p00002000 checkMethodSegmentGetP2 (21/57)
edu.uha.miage.GeneratedSegmentTest.p00002000 checkMethodSegmentLength (22/57)
edu.uha.miage.GeneratedSegmentTest.p00002000 checkMethodSegmentProjectionLengthOnX (23/57)
edu.uha.miage.GeneratedSegmentTest.p00002000 checkMethodSegmentToString (24/57)
edu.uha.miage.GeneratedSegmentTest.p00002000 checkGetterSegmentP1 (25/57)
edu.uha.miage.GeneratedSegmentTest.p00002000 checkGetterSegmentP2 (26/57)
edu.uha.miage.GeneratedSegmentTest.p00002000 checkSetterSegmentP1 (27/57)
edu.uha.miage.GeneratedSegmentTest.p00002000 checkSetterSegmentP2 (28/57)
edu.uha.miage.GeneratedSegmentTest.p00002000 testMethodSegmentLength (29/57)
edu.uha.miage.GeneratedSegmentTest.p00002000 testMethodSegmentProjectionLengthOnX (30/57)
edu.uha.miage.SegmentTest.testConstructor (31/57)
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.p00003000 checkConstructorTrianglePointPointPoint (36/57)
edu.uha.miage.GeneratedTriangleTest.p00003000 checkMethodTriangleGetP1 (37/57)
edu.uha.miage.GeneratedTriangleTest.p00003000 checkMethodTriangleGetP2 (38/57)
edu.uha.miage.GeneratedTriangleTest.p00003000 checkMethodTriangleGetP3 (39/57)
edu.uha.miage.GeneratedTriangleTest.p00003000 checkMethodTriangleGetP1Point (40/57)
edu.uha.miage.GeneratedTriangleTest.p00003000 checkMethodTriangleGetP2Point (41/57)
edu.uha.miage.GeneratedTriangleTest.p00003000 checkMethodTriangleGetP3Point (42/57)
edu.uha.miage.GeneratedTriangleTest.p00003000 checkMethodTriangleGetPerimetre (43/57)
edu.uha.miage.GeneratedTriangleTest.p00003000 testMethodTriangleGetPerimetre (44/57)
edu.uha.miage.GeneratedTriangleTest.p00003000 checkMethodTriangleGetSurface (45/57)
edu.uha.miage.GeneratedTriangleTest.p00003000 testMethodTriangleGetSurface (46/57)
edu.uha.miage.GeneratedTriangleTest.p00003000 checkMethodTriangleGetBaryCentre (47/57)
edu.uha.miage.GeneratedTriangleTest.p00004000 checkMethodTriangleToString (48/57)
edu.uha.miage.TriangleTest.testConstructor (49/57)
edu.uha.miage.TriangleTest.testSetP1 (50/57)
edu.uha.miage.TriangleTest.testSetP2 (51/57)
edu.uha.miage.TriangleTest.testSetP3 (52/57)
edu.uha.miage.TriangleTest.testGetP1 (53/57)

► Execution

Realisation d'exercices de programmation auto-évalués sur une pl

Des classes absentes

Java L3 (UHA) Fraction et équation Éditer - Google Chrome

moodle.caseine.org/mod/vpl/forms/edit.php?id=16692&userid=10826

File list

- edu
 - miage
 - fractions
 - Fraction.java
 - main
 - Main.java
 - equations
 - EquationDuPremierDeg
 - exceptions
 - DenominateurNulExcep

Fraction.java

```
1  /*
2  * @author : Yvan Maillot (yvan.maillot@uha.fr)
3  */
4  package edu.uha.miage.fractions;
5
6  import edu.uha.miage.exceptions.*;
7
8  /**
9   * Dans cet exercice, vous allez réaliser une classe immuable
10  * de cette classe est de permettre le calcul fractionnaire. Elle
11  * encapsuler le numerateur et le denominateur pour garantir le
12  *
13  * En particulier, leur denominateur ne doit jamais être nul,
14  * et elles doivent être réduites.
15  *
16  * Exception
17  *
18  * Afin de s'assurer qu'une fraction au dénominateur nul ne soit
19  * vous allez devoir utiliser les exceptions. La première à faire
20  * créer une exception contrôlée edu.uha.miage.exceptions.Denom
21  *
22  * Elle servira à interdire la survenue de fractions dont le de
23  *
24  * Classe
25  *
26  * La classe Fraction doit se trouver dans un paquetage fraction
27  * et ne pas pouvoir être dérivée.
28  *
29  * Attributs
30  *
31  * Elle possède deux attributs entiers, constants et publics :
32  * le denominateur.
33  *
34  * Exception
35  *
36  * Une exception contrôlée DenominateurNulException doit être
```

Note proposée: 1,79 / 20

Commentaires

Tests results

edu.uha.miage.exceptions.GeneratedDenominatorNulException (1/34)

Incorrect program result

Message : java.lang.AssertionError: Ne pas oublier de contrôler DenominateurNulException (Inheritance)

edu.uha.miage.fractions.GeneratedFractionTest.p (2/34)

Incorrect program result

Message : java.lang.AssertionError: les attributs numerateur et denominateur

edu.uha.miage.fractions.GeneratedFractionTest.p (3/34)

Incorrect program result

Message : java.lang.AssertionError: les attributs numerateur et denominateur

edu.uha.miage.fractions.GeneratedFractionTest.p (4/34)

Incorrect program result

Message : java.lang.AssertionError: Constructeur a de

edu.uha.miage.fractions.GeneratedFractionTest.p (5/34)

Incorrect program result

Message : java.lang.AssertionError: Constructeur a un

edu.uha.miage.fractions.GeneratedFractionTest.p (6/34)

edu.uha.miage.fractions.GeneratedFractionTest.p (7/34)

Incorrect program result

Message : java.lang.AssertionError: Fix methode oppo

edu.uha.miage.fractions.GeneratedFractionTest.p

Exécution

Réalisation d'exercices de programmation auto-évalués sur une pl

Évaluation de performance

Java L3 (UHA) Polynômes immuables de fractions immuables Éditer - Google Chrome

moodle.caseine.org/mod/vpl/forms/edit.php?id=17515&userid=10826

File list

- edu
- edu.uha
- edu.uha.miage
- edu.uha.miage.fractions
- edu.uha.miage.fractions.Polynome.java
- edu.uha.miage.fractions.Fraction.java
- edu.uha.miage.fractions.exceptions
- edu.uha.miage.fractions.PasUneEquationDu...
- edu.uha.miage.fractions.InversionFraction...
- edu.uha.miage.fractions.DenominateurNu...
- edu.uha.miage.fractions.PolynomeNulEx...
- edu.uha.miage.fractions.SituationImposs...
- edu.uha.miage.fractions.DivisionParUneF...
- edu.uha.miage.fractions.main
- edu.uha.miage.fractions.Main.java
- caseine
- format
- javajunit
- TimeLimit.java

```
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 getDegree() {
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(i);
69         }
70         return sb.toString();
71     }
72 }
```

Note proposée: 14,74 / 20

Commentaires

edu.uha.miage.fractions.FractionTest.p000000200 - Test (1/12)

edu.uha.miage.fractions.GeneratedPolynomeTest.p000000200 - Test (2/12)

edu.uha.miage.fractions.GeneratedPolynomeTest.p000000200 - Test (3/12)

edu.uha.miage.fractions.GeneratedPolynomeTest.p000000200 - Test (4/12)

edu.uha.miage.fractions.GeneratedPolynomeTest.p000000200 - Test (5/12)

edu.uha.miage.fractions.GeneratedPolynomeTest.p000000200 - Test (6/12)

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

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

edu.uha.miage.fractions.PolynomeTest.testGetDegree (5/12)

edu.uha.miage.fractions.PolynomeTest.testEvaluer (10/12)

Incorrect program result

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

edu.uha.miage.fractions.PolynomeTest.performanceEvaluation (12/12)

Tests results

evaluationTime (1/1)

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

Method : evaluationTime

Exécution

Description

Évaluation de performance

Java L3 (UHA) Polynômes immuables de fractions immuables Éditer - Google Chrome

moodle.caseine.org/mod/vpl/forms/edit.php?id=17515&userid=10826

File list

- uaha
 - miage
 - fractions
 - Polynome.java
 - Fraction.java
 - exceptions
 - PasUneEquationDu...
 - InversionFrac...
 - Denom...
 - main
 - Main.java
- caseine
 - format
 - javajunit
 - TimeLimit.java

Polynome.java

```
38 this.a = new Fraction[0];
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 PolynomeNullException();
69     }
70 }
71
72 }
```

Note proposée: 16,84 / 20

Commentaires

edu.uha.miage.fractions.GeneratedPolynomeTest.p0000000100_re (12/12)

edu.uha.miage.fractions.GeneratedPolynomeTest.p0000000100_re (2/12)

edu.uha.miage.fractions.GeneratedPolynomeTest.p0000000100_re (3/12)

edu.uha.miage.fractions.GeneratedPolynomeTest.p0000000100_re (4/12)

edu.uha.miage.fractions.GeneratedPolynomeTest.p0000000100_re (5/12)

edu.uha.miage.fractions.GeneratedPolynomeTest.p0000000100_re (6/12)

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

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

edu.uha.miage.fractions.PolynomeTest.testGetDegre (5/12)

edu.uha.miage.fractions.PolynomeTest.testEvaluer (10/12)

edu.uha.miage.fractions.PolynomeTest.performanceEv (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.p0000000100_re (12/12)

Tests results

evaluationTime (1/1)

Message : We are unable to guess the runtime complexity of this method accurately. It seems worse than the expected complexity but we will check with a teacher.

(Attempt of detection : $n^{(2.32293)}$)

Method : evaluationTime

Exécution

Description

Évaluation de performance

Java L3 (UHA) Polynômes immuables de fractions immuables Éditer - Google Chrome

moodle.casine.org/mod/vpl/forms/edit.php?id=17515&userid=10826

File list

- edu
 - uha
 - miage
 - fractions
 - Polynome.java
 - Fraction.java
 - exceptions
 - PasUneEquation
 - InversionFraction
 - DenominateurNul
 - PolynomeNulException
 - SituationImpossible
 - DivisionParZero
 - main
 - Main.java

casine

 - format
 - javajunit
 - TimeLimit.java

Polynome.java

```
39 for (int i = 0; i < this.a.length; ++i) {
40     this.a[i] = a[i];
41 }
42 */
43 degree = as.length - 1;
44 while (degree >= 0 && as[degree].equals(new Fraction())) {
45     a = Arrays.copyOf(as, degree + 1);
46 }
47
48 public Fraction get(int i) {
49     return a[i];
50 }
51
52 public int getDegree() {
53     return degree;
54 }
55
56 public Fraction evaluer(Fraction x) {
57     try {
58         Fraction r = a[0];
59         Fraction xpi = new Fraction(1);
60         for (int i = 1; i < a.length; ++i) {
61             xpi = xpi.fois(x);
62             r = r.plus(a[i].fois(xpi));
63         }
64         return r;
65     } catch (ArrayIndexOutOfBoundsException ai) {
66         throw new PolynomeNulException();
67     }
68 }
69
70 @Override
71 public String toString() {
72     if (a.length == 0) {
73         return "Polynome nul";
74     }
75 }
```

Note proposée: 20 / 20

Commentaires

Tests results

edu.uha.miage.fractions.FractionTest.p000000100_red
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
edu.uha.miage.fractions.PolynomeTest.testGet (8/12)
edu.uha.miage.fractions.PolynomeTest.testGetDegree (5
edu.uha.miage.fractions.PolynomeTest.testEvaluer (10
edu.uha.miage.fractions.PolynomeTest.performanceEv
edu.uha.miage.fractions.PolynomeTest.p000000100_re
(12/12)

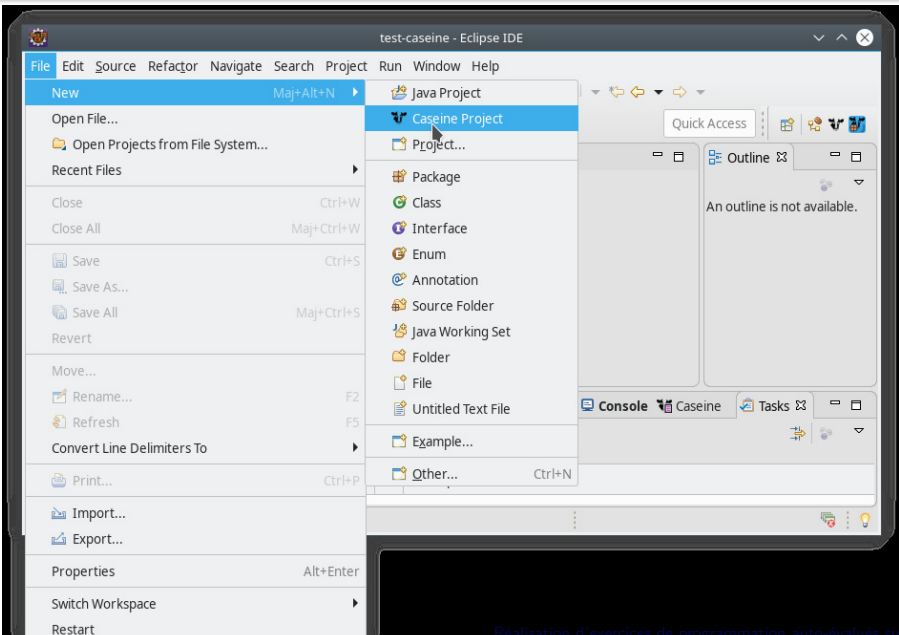
Tests results
evaluationTime (1/1)

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

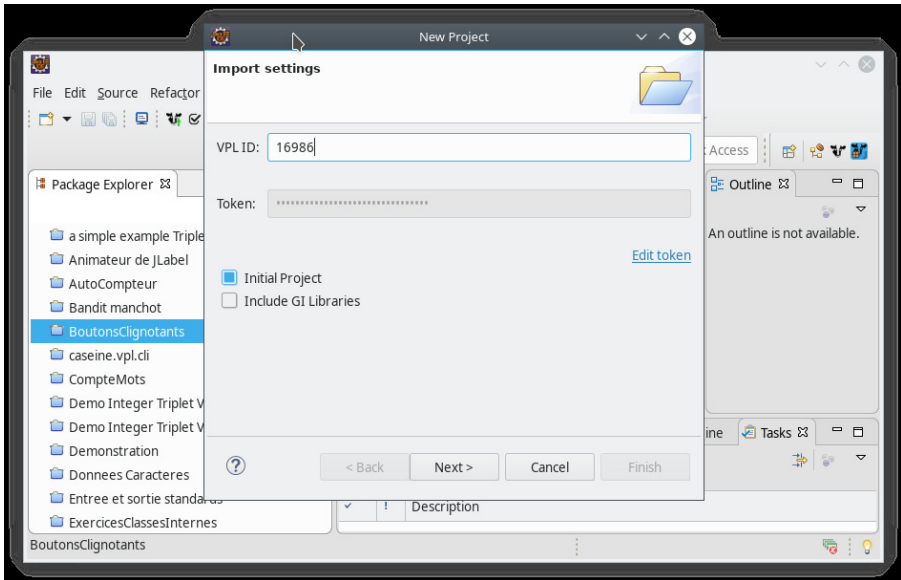
Exécution

Réalisation d'exercices de programmation auto-évalués sur une pl

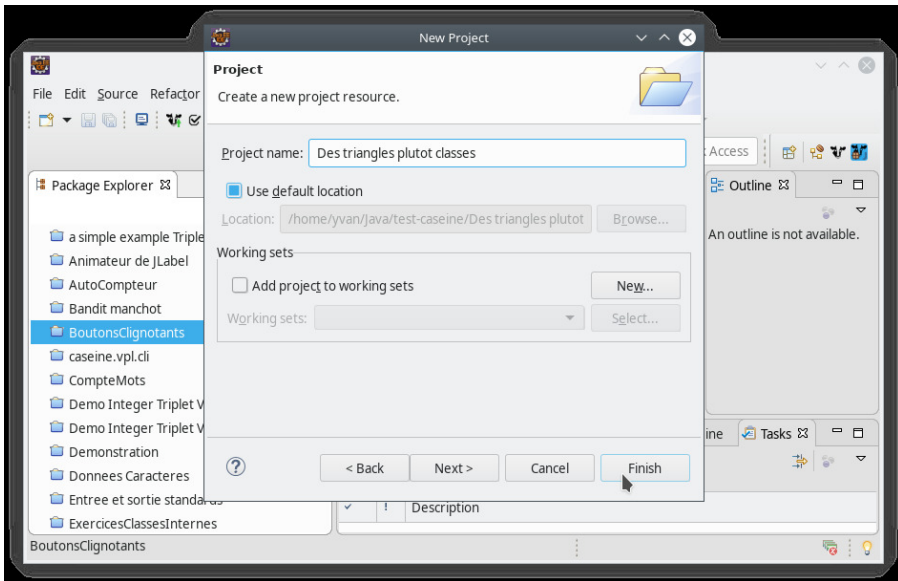
Le plugin Eclipse



Le plugin Eclipse



Le plugin Eclipse



Le plugin Eclipse

The screenshot displays the Eclipse IDE interface. The top menu bar includes File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, and Help. The Package Explorer on the left shows the project structure, with 'Des triangles plutoit classes' expanded to show 'Main.java', 'Point.java', 'Segment.java', and 'Triangle.java'. The main editor window shows the code for 'Point.java', which includes a package declaration and a class with several TODO comments. The Outline view on the right shows the class hierarchy, and the CaseView view displays a diagram of a triangle with vertices p1, p2, and p3, and segments s1, s2, and s3. The Problems view at the bottom shows a list of 26 items, including TODO comments and a warning about a missing resource.

```
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 coordonnées cartésiennes de
8
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,
20
21 }
22
```

Des triangles plutoit classes

Description Result

trois segments qui les relient, appelés côtés, délimitant un domaine du plan appelé intérieur."

p1 p2 p3 s1 s2 s3

Problems 26 items

Description	Resource	Path	Location	Type
TODO 1.01. Déclarer x et y, les coordonnées cartésiennes de ce point	Point.java	/Des triangles plutoit classes/src/edu/uha/miage/	line 7	Java Task
TODO 1.02. Écrire le constructeur aux coordonnées cartésiennes	Point.java	/Des triangles plutoit classes/src/edu/uha/miage/	line 9	Java Task
TODO 1.03. Écrire un getter pour x	Point.java	/Des triangles plutoit classes/src/edu/uha/miage/	line 11	Java Task
TODO 1.04. Écrire un getter pour y	Point.java	/Des triangles plutoit classes/src/edu/uha/miage/	line 13	Java Task
TODO 1.05. Écrire un setter pour x	Point.java	/Des triangles plutoit classes/src/edu/uha/miage/	line 15	Java Task
TODO 1.06. Écrire un setter pour y	Point.java	/Des triangles plutoit 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 plutoit 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 plutoit 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 plutoit classes/src/edu/uha/miage/	line 12	Java Task

Writable Smart Insert 2:1

Le plugin Eclipse

test-caseine - Des triangles plutôt classes/src/edu/uha/miage/Point.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer: a simple example Tr, Animateur de JLab, AutoCompteur, Bandit manchot, BoutonsClicnontants, caseine.vpl.ci, CompteMots, Correction Taquin, Demo Integer Tripl, Demo Integer Tripl, Demonstration, Des triangles plutôt classes, JRE System Libra, src, edu.uha.miage, Main.java, Point.java, Segment.java, Triangle.java, JUnit 5, Donnees Caractere, Entree et sortie stat, ExercicesClassesInt, Extreme, Fourmilliere, Fraction et equation, Fractions, Fractions_2, Genealogie, Horaire exceptionn, Illustrative Example, Kruskal, Kruskal_2

```
2 * @author : Yvan Maillot (yvan.maillot@uha.  
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
```

Outline Caseine View

Des triangles plutôt classes

Description Result

57 Tests Errors: 0 Failures: 57

Proposed grade: 0 / 20

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

Problems Javadoc Declaration Console Caseine Comments Tasks

✓	Description	Resource	Path	Location	Type
	TODO 1.01. Déclarer x et y, les coordonnées cartésiennes de ce point	Point.java	/Des triangles plutôt classes/src/edu/uha/miage/	line 7	Java Task
	TODO 1.02. Écrire le constructeur aux coordonnées cartésiennes	Point.java	/Des triangles plutôt classes/src/edu/uha/miage/	line 9	Java Task
	TODO 1.03. Écrire un getter pour x	Point.java	/Des triangles plutôt classes/src/edu/uha/miage/	line 11	Java Task
	TODO 1.04. Écrire un getter pour y	Point.java	/Des triangles plutôt classes/src/edu/uha/miage/	line 13	Java Task
	TODO 1.05. Écrire un setter pour x	Point.java	/Des triangles plutôt classes/src/edu/uha/miage/	line 15	Java Task
	TODO 1.06. Écrire un setter pour y	Point.java	/Des triangles plutôt 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 plutôt 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 plutôt classes/src/edu/uha/miage/	line 10	Java Task

Le plugin Eclipse

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 left sidebar shows the Project Explorer with a tree view of the project structure, including a package 'edu.uha.miage' and a class 'Point.java'. The main editor window displays the source code of 'Point.java', which includes a package declaration, a class declaration, and several TODO comments. The right sidebar shows the Outline view and the Case View, which displays a list of tests and their results. The bottom panel shows a table of tasks, including descriptions, resources, paths, locations, and types.

```
2 * @author : Yvan Maillot (yvan.maillot@uha.  
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
```

Description	Resource	Path	Location	Type
TODO 1.01. Déclarer x et y, les coordonnées cartésiennes de ce point	Point.java	/Des triangles plutôt classes/src/edu/uha/miage/ line 7		Java Task
TODO 1.02. Écrire le constructeur aux coordonnées cartésiennes	Point.java	/Des triangles plutôt classes/src/edu/uha/miage/ line 9		Java Task
TODO 1.03. Écrire un getter pour x	Point.java	/Des triangles plutôt classes/src/edu/uha/miage/ line 11		Java Task
TODO 1.04. Écrire un getter pour y	Point.java	/Des triangles plutôt classes/src/edu/uha/miage/ line 13		Java Task
TODO 1.05. Écrire un setter pour x	Point.java	/Des triangles plutôt classes/src/edu/uha/miage/ line 15		Java Task
TODO 1.06. Écrire un setter pour y	Point.java	/Des triangles plutôt 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 plutôt 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 plutôt classes/src/edu/uha/miage/ line 10		Java Task

Le plugin Eclipse

The screenshot shows the Eclipse IDE interface. The main editor displays the source code of the `Point` class in `Point.java`. The code includes package declarations, imports, and a class definition with several TODO comments for implementing methods like `getPointFieldX`, `getPointFieldY`, `getConstructorPointDoubleDouble`, `getMethodPointGetX`, `getMethodPointGetY`, `getMethodPointSetXDouble`, `getMethodPointSetYDouble`, `getMethodPointToString`, and `checkGetterPointX`.

The **Caseine View** window on the right shows the test results for the `Des triangles pluto` classes. It indicates 57 tests, 0 errors, and 55 failures. The proposed grade is 0.32 / 20. The tests are listed in a table below.

Description	Resource	Path	Location	Type
TODO 1.01. Déclarer x et y, les coordonnées cartésiennes de ce point	Point.java	/Des triangles pluto classes/src/edu/uha/miage/ line 7	Java Task	
TODO 1.02. Écrire le constructeur aux coordonnées cartésiennes	Point.java	/Des triangles pluto classes/src/edu/uha/miage/ line 9	Java Task	
TODO 1.03. Écrire un getter pour x	Point.java	/Des triangles pluto classes/src/edu/uha/miage/ line 11	Java Task	
TODO 1.04. Écrire un getter pour y	Point.java	/Des triangles pluto classes/src/edu/uha/miage/ line 13	Java Task	
TODO 1.05. Écrire un setter pour x	Point.java	/Des triangles pluto classes/src/edu/uha/miage/ line 15	Java Task	
TODO 1.06. Écrire un setter pour y	Point.java	/Des triangles pluto 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 pluto 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 pluto classes/src/edu/uha/miage/ line 10	Java Task	

Du Swing

test-caseine - Points Cardinaux/src/edu/uha/miage/frames/PointsCardinaux.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer

- Fractions
- Fractions_2
- Genealogie
- Horaire exceptionnel
- Illustrative Example
- Kruskal
- Kruskal_2
- Le duplicateur
- Lecture dans un fichier
- Manipulation Chainetique
- Manipulation de tableaux
- Nombres premiers
- Petite discussion entre
- Petites Discussions Entre
- Points Cardinaux
 - JRE System Library
 - src
 - edu.uha.miage
 - Main.java
 - edu.uha.miage.fr
 - PointsCardinaux
- JUnit 5
- Referenced Libraries
- lib
- Points IMMUALES et p
- Points MUTABLES et po
- Points Segments et Tri

```
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 de  
20  
21    */  
22
```

Problems Javadoc Declaration Console

2 items

Description
TODO 1.01 Ajouter au bouton lance une action
TODO 1.01 La classe PointsCardinaux est une

Démonstration de l'exo "Les points cardinaux"

Nord

Ouest Centre Est

Sud

Points Cardinaux

Description

Reload description

Path

/Bandit manchot/src/edu/u
/Points Cardinaux/src/edu/u

Du Swing et des ressources

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

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer

- Correction Taquin
 - JRE System Library
 - src
 - edu.uha.miage.d
 - 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
 - MAXIMU
 - MAXIMU
 - MINIMU
 - MINIMU
 - image
 - melange
 - piece
 - trou
 - Taquin(ir
 - clickPiece
 - deplacer
 - melange
 - positionr

Problems

1 items

Code Editor

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

Quick Access

Correction Taquin

Description

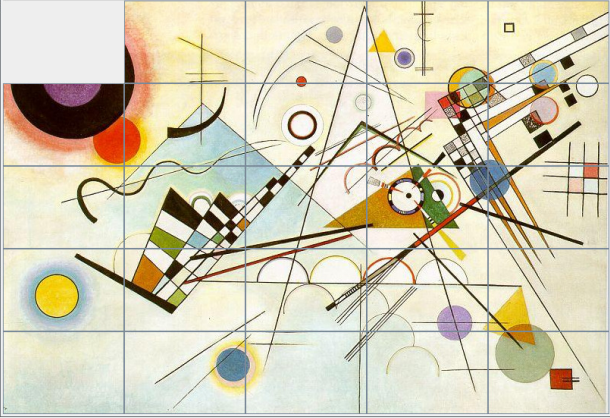
[Reload description](#)

Path

Bandit manchot/src/edu/u

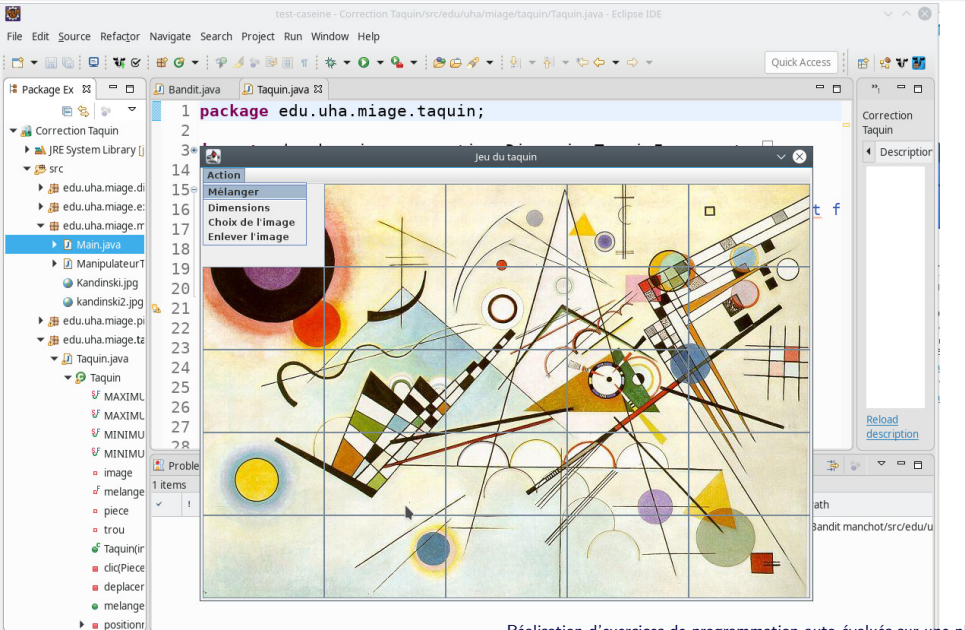
Jeu du taquin

Action



Réalisation d'exercices de programmation auto-évalués sur une pl

Du Swing et des ressources



Du Swing et des ressources

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
 - src
 - edu.uha.miage.d
 - 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
 - MAXIMU
 - MAXIMU
 - MINIMU
 - MINIMU
 - image
 - melange
 - piece
 - trou
 - Taquin(ir
 - clie(Piece
 - deplacer
 - melange
 - position

1 package edu.uha.miage.taquin;

2

3

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

Jeu du taquin

Action

Problem

1 items

ath

Bandit manchot/src/edu/u

Correction Taquin

Description

Reload description

Réalisation d'exercices de programmation auto-évalués sur une pl

Du Swing et des ressources

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

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer

- Correction Taquin
 - JRE System Library
 - src
 - edu.uha.miage.d
 - 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
 - MAXIMU
 - MAXIMU
 - MINIMU
 - MINIMU
 - image
 - melange
 - piece
 - trou
 - Taquin(ir
 - clit(Piece
 - deplacer
 - melange
 - positionr

1 package edu.uha.miage.taquin;

2

3

14

15

16

17

18

19

20

21

22

23

24

25

26

27

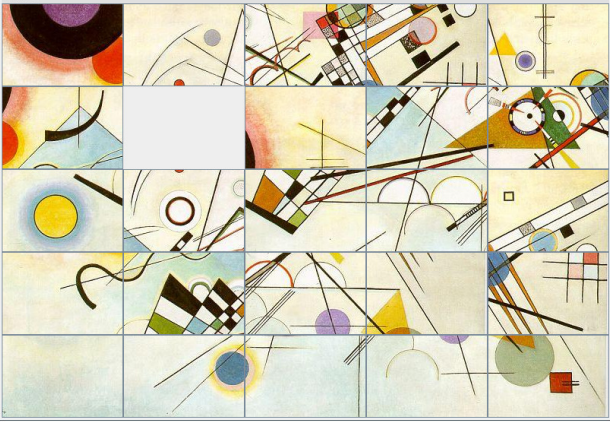
28

Probleme

1 items

jeu du taquin

Action



Correction Taquin

Description

[Reload](#)

[description](#)

ath

Bandit manchot/src/edu/u

Réalisation d'exercices de programmation auto-évalués sur une pl

Du Swing, des ressources et des threads

Animateur de JLabel

Bandit manchot (À FAIRE)

0 0 0

144

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  
    nt, par en m  
    ont cessées  
    (À FAIRE)");  
    demo = new
```

CE QU'IL FAUT FAIRE

3 2 0

Problems Javadoc Declaration Console Caseine Comments Tasks

1 items

- Moodle et VPL avec
 - auto-évaluation
 - auto-correction
 - un guide
- Pour l'écriture de code en
 - Java
 - C
 - C++
 - SQL
 - Python
 - R
 - ...
- Le plugin Eclipse (en Java)
- ...

Pour chaque activité

Types simples	 Régions du plan	 67%	 33%	 0%	6	+					
Objets	 Manipulation Chainétique	 67%	 22%	 11%	9	-					
		 	 	 							
		 	 								
		 									
		 									
		 									
		 									
Objets	 Manipulation de tableaux	 100%	 0%	 0%	7	+					
Classes et encapsulation	 AutoCompteur	 57%	 29%	 14%	7	+					

Globalement

Globalement

Voir toutes les réponses. **Tous les participants.** Ordre normal ⓘ Réponses: 39

Questionnaire et avis à propos du cours de Java en L3

1

Combien d'heures de travail personnel vous a demandé la préparation de chaque séance de 2 heures de TD en salle ?

Réponse	Moyenne	Total
Aucune	3%	1
Moins d'une heure	8%	3
De une à deux heures	32%	12
De deux à trois heures	37%	14
De trois à quatre heures	13%	5
Plus de quatre	8%	3
Total des réponses à la question		38/39

Globalement

2

Au lieu d'avoir à lire le cours chez vous, auriez-vous préféré des cours magistraux (**en amphî**) en plus des TD (en salle)

Réponse	Moyenne	Total
Non, pas du tout, c'était très bien comme ça.	38%	15
Non, mais il faudra que je m'organise mieux.	15%	6
Oui, quelques heures en plus pour avoir des explications.	33%	13
Oui, une heure de cours pour deux heures de TD	10%	4
Rien de tout ça, mais : Rajouter des questionnaires de fin de chapitre pour évaluer nos connaissances.	3%	1
Total des réponses à la question	100%	39/39

Globalement

3

Qu'avez-vous pensé du système des exercices auto-évalués à l'aide de Caseine ?

Réponse	Moyenne	Total
C'était trop bien	26%	10
C'était bien mais trop exigeant, difficile d'avoir 20 à tous les exercices	31%	12
C'était pas mal mais je préférerais avoir plus de liberté	38%	15
C'était trop difficile	3%	1
Rien de tout ça, mais : Une meilleure transparence sur les exigences à remplir pour avoir la note maximale (voir commentaire)	3%	1
Total des réponses à la question	100%	39/39

Globalement

4

Qu'avez-vous pensé du plugin Eclipse ?

Réponse	Moyenne	Total
Ça n'aurait pas été possible sans	8%	3
C'est vraiment pratique	82%	32
Pas mal	5%	2
Je préfère utiliser mon navigateur	5%	2
Total des réponses à la question	100%	39/39

Globalement

5

Pour chacun des éléments suivants, donnez votre niveau de satisfaction sur une échelle de 1 à 6, 1 : pas du tout satisfait, 6 tout à fait satisfait

	Rang moyen						↓	NSP
	1	2	3	4	5	6		
Les fascicules de cours					■		4.8	0
Les exercices à faire chez soi					■		4.8	0
Les exercices à faire en séances					■		4.7	1
Les vidéos					■		4.5	8
Le système d'auto-évaluation de Caseine					■		4.7	0
Le cours dans sa globalité					■		4.9	0

Caséine : vu de l'enseignant

Basée sur Moodle, caseine est librement accessible à tous les établissements du réseau mondial eduGAIN (dont Renater)
(Accessible par son identifiant institutionnel)

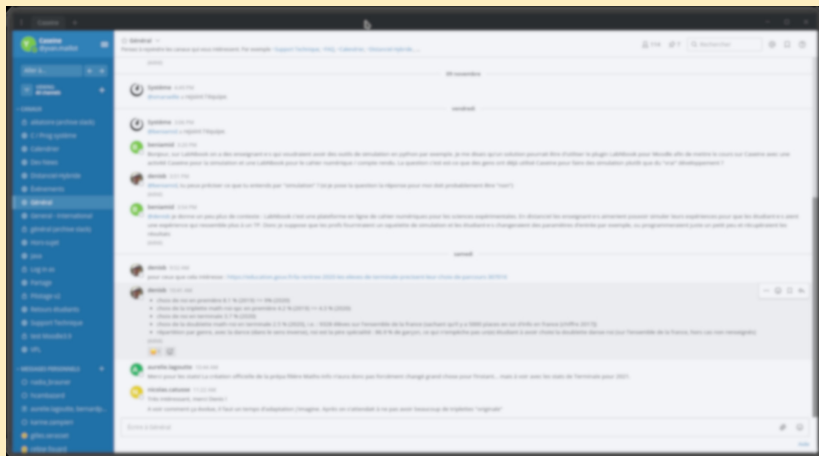
Un Moodle et son VPL, pour

- monter ses cours
- scénariser des exercices auto-évaluables, auto-corrigeable, guidés
- ...

Une communauté

- Utilisée par
 - 10 établissements universitaires français
 - 2 internationaux et
 - 6 lycées.
- Sa communauté regroupe 92 enseignants.

Un mattermost qui permet à la communauté de communiquer



Caséine : le partage

Une plateforme ouverte que chacun peut faire évoluer au profit des autres

Par la configuration d'activités VPL spécifiques

☐ Usable as base

DefaultActivity.JavaPackage

DefaultActivity.JavaPackage4Tests

Shared

DefaultActivity.JavaPackage

DefaultActivity.OplCustom

DefaultActivity.OplCustomCommunity

DefaultActivity-VPLPython3-Evaluate

DefaultActivity-VPLPython3-Run+Evaluate

DefaultActivity-VPLPython3-RunFenetreGraphique+Evaluate

Exercice 2 : conversion de durées

Exercice 5 : Fibonacci

Modèle TP

DefaultActivityPython3Unitest

DefaultActivityPython3Unitest_noscript

Conversions monétaires

DefaultActivity-SQLv2

Basic system programming default-activity

C programming default-activity

DefaultActivityPython3Unitest+CA

DefaultActivity-VPLPython3-Run+Evaluate+CA

DefaultActivity.JavaPackage+CA

Une plateforme ouverte que chacun peut faire évoluer au profit des autres

Par le développement de plugins

- Le plugin des « pistes de ski »
- Le plugin pour le retour des étudiants par activité
- Le QVPL
- Le plugin Eclipse
- Le plugin de conception Java
- ...

De nombreuses ressources partagées

Grace à l'espace de partage et le « panier »

The screenshot displays a web interface for sharing resources. On the left, a sidebar titled 'UHA' contains a list of categories: 'Java-M2-UHA', 'Intro-réseaux-L2-Info-UHA', 'InformatiqueLycee', 'UML open', 'Java-M1-UHA', 'Plus...', 'Espace de partage' (highlighted with a mouse cursor), and 'Ajouter un bloc'. The main area on the right shows the contents of the 'Espace de partage' folder. It starts with a folder icon and the title 'Classes Mutables vs Classes Immuables'. Below this, there is a sub-folder icon and the text 'Cours à lire avant de cocher (Classes Mutables vs Classes Immuables)'. Inside this sub-folder, two PDF files are listed: '03-classes-immuables-fascicule-paysage.pdf' and '03-classes-immuables-fascicule-portrait.pdf'. A button labeled 'Télécharger le dossier' is positioned below the files. Further down, several other resources are listed, each with a document icon and a title: 'Caractères UNICODE', 'Equation', 'Points MUTABLES et polygones MUTABLES', 'Observations à propos de l'exercice Points MUTABLES et Polygones MUTABLES', and 'Points IMMUABLES et polygones MUTABLES'. Each of these resources has a small icon indicating access restrictions. Below the 'Observations' resource, a message states: 'Accès restreint Non disponible à moins que : L'activité Points MUTABLES et po'. Below the 'Points IMMUABLES' resource, another message states: 'Accès restreint Non disponible à moins que l'une des conditions suivantes soit'.

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 po
- + Points IMMUABLES et polygones MUTABLES
 - Accès restreint Non disponible à moins que l'une des conditions suivantes soit

Comment profiter des ressources offertes par les collègues ?

En allant dans l'espace de partage avec son « panier »...

The screenshot shows a web browser window with the URL `moodle.caseine.org/local/sharedspaceh/index.php`. The page header includes a menu icon, the site name "Caseine", the language "Français (fr)", and a user profile for "Yvan Maillot". The main content area has a breadcrumb "Accueil / Espace de partage" and a message: "L'espace de partage de caseine est encore en développement. Il est néanmoins fonctionnel et en progrès !". There are two search sections: "Recherche par mots clefs" with a search box labeled "Rechercher" and a link "Recherche avancée"; and "Critères de recherche sur les identifiants moodle" with a search box labeled "Rechercher". A right sidebar titled "Panier d'activités" contains a list of activities: "Coquille vide", "Coquille vide avec Tests Graphiques", and "Coquille vide pour C++", each with a question mark icon and a close button.

Comment profiter des ressources offertes par les collègues ?

On cherche des exercices en Java, par exemple...

The screenshot shows a web browser window displaying the Moodle Caseine interface. The browser's address bar shows the URL `moodle.caseine.org/local/sharedspaceh/index.php?advsearch=1`. The page header includes the Caseine logo, the language 'Français (fr)', and a user profile for 'Yvan Maillot'. The main content area features a search bar at the top with the text 'Rechercher'. Below this, there are four filter categories, each with a search bar and a dropdown menu:

- Author(s)**: Search bar with 'Rechercher' and a dropdown menu.
- Programming Langage**: Search bar with 'Rechercher' and a dropdown menu. A blue tag labeled 'x Java' is visible next to the search bar.
- Pedagogical objectives**: Search bar with 'Rechercher' and a dropdown menu.
- Academic Level**: Search bar with 'Rechercher' and a dropdown menu.

The dropdown menu for 'Programming Langage' is open, showing a list of programming languages: Python, OPL, ProLog, R, gnu-r, Excel LP solver, C, Java8, and Erlang. The 'Java' option is highlighted in blue.

Comment profiter des ressources offertes par les collègues ?

Il y en a 235 ! Tiens, « Palindrome » m'intéresse => dans le panier

Shared Caseine Space - Google Chrome

moodle.caseine.org/local/sharedspaceh/index.php

Caseine Français (fr)

Resultat: 235

Télécharger les données au format Télécharger

1 2 3 4 5 6 7 8 »

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 processing, computer science
2	57	Couleur ⓘ		Informatique - python (GI, 1A-IPID)	1	vpl	Computer programming, programs & data	Data processing, computer science

Rechercher ▼

Activity type Aucune sélection

Rechercher ▼

Main Dewey classification Aucune sélection

Rechercher ▼

Disciplinary key concepts Aucune sélection

Rechercher ▼

Author(s) Aucune sélection

Rechercher ▼

Programming Language ☒ Java

Rechercher ▼

Panier d'activités + - ⚙

Comment profiter des ressources offertes par les collègues ?

Le Palindrome est dans mon panier maintenant.

The screenshot shows a Moodle LMS interface. The top navigation bar includes the Moodle logo, the course name 'Caseine', and the language 'Français (fr)'. The main content area displays a course page for 'Informatique - python (GI, 1A-IPID)' with a table of activities. The table has columns for activity ID, name, and description. The activities listed are 'Patient', 'MultiplesDeTroisEtCinq*', and 'Palindrome'. The 'Palindrome' activity is highlighted in the table. On the right side, there is a sidebar titled 'Panier d'activités' (Activities Basket) which contains a list of activities: 'Coquille vide', 'Coquille vide avec Tests Graphiques', 'Coquille vide pour C++', and 'Palindrome'. The 'Palindrome' activity is marked with a question mark icon, indicating it is the current focus.

ID	Nom	Description
3	Patient	Informatique - python (GI, 1A-IPID)
4	MultiplesDeTroisEtCinq*	Informatique - python (GI, 1A-IPID)
58	Palindrome	Informatique - python (GI, 1A-IPID)

Panier d'activités

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

Comment profiter des ressources offertes par les collègues ?

Je retourne dans mon cours...

The screenshot shows a web browser window displaying the Moodle LMS interface. The browser's address bar shows the URL `moodle.caseine.org/course/view.php?id=446`. The page title is "Compléments Java M2 (UHA)". The user is logged in as "Yvan Maillot". The left sidebar contains a menu with the following items: "Participants", "Compétences", "Notes", "Introduction", "Introspection", "Énumération", "Annotations", "Lambda-Expressions", "Internationalisation", "Client-Serveur", and "Synchronisation avancée". The main content area shows the course title "Compléments Java M2 (UHA)" and a button "Quitter le mode édition". Below the title, there is a section "Coller cet élément:" with a "Palindrome" resource. The "Annonces" resource is visible, and a button "Ajouter une activité ou ressource" is present. The "Panier d'activités" (Activity Basket) on the right contains the following items: "Coquille vide", "Coquille vide avec Tests", "Graphiques", "Coquille vide pour C++", and "Palindrome".

Comment profiter des ressources offertes par les collègues ?

pour l'y déposer.

The screenshot shows a Moodle course page for 'Compléments Java M2 (UHA)'. The left sidebar contains a menu with the following items: Participants, Compétences, Notes, Introduction, Introspection, Énumération, Annotations, Lambda-Expressions, Internationalisation, Client-Serveur, and Synchronisation avancée. The main content area displays a list of activities and resources. The first section is 'Annonces' (Announcements) with a 'Palindrome' resource, marked as 'Caché pour les étudiants' (Hidden from students). The second section is 'Introduction' with an 'IntroCMJAVA-article' resource. The third section is 'Introspection' with resources: 'introspection-fascicule-paysage', 'introspection-fascicule-portrait', 'Dumper', 'AnalyseurDeClassesARemplir', and 'Vidéo Cours "Introspection"'. Each resource has a 'Modifier' (Edit) button and a checkbox. A right sidebar titled 'Panier d'activités' (Activity Basket) lists: 'Coquille vide', 'Coquille vide avec Tests', 'Graphiques', 'Coquille vide pour C++', and 'Palindrome'. The user 'Yvan Maillot' is logged in.

Comment partager ses ressources ?

Je veux partager mon VPL « Dumper »

The screenshot shows a web browser window displaying a Moodle course page. The browser's address bar shows the URL `moodle.caseine.org/course/view.php?id=446§ion=0`. The page title is 'Cours : Compléments Java M2 (UHA) - Google Chrome'. The course name is 'Caseine'. The course category is 'Java-M2-UHA'. The course content is organized into sections: 'Participants', 'Compétences', 'Notes', 'Introduction', 'Introspection', 'Énumération', 'Annotations', 'Lambda-Expressions', 'Internationalisation', 'Client-Serveur', and 'Synchronisation avancée'. The 'Introspection' section is expanded, showing a list of resources: 'IntroCMJAVA-article', 'introspection-fascicule-paysage', 'introspection-fascicule-portrait', 'Dumper', 'AnalyseurDeClassesARemplir', and 'Vidéo Cours "Introspection"'. The resource 'Dumper' is highlighted. The password for the course is 'Le mot de passe est pRSzIFS3'. The user 'Yvan Maillot' is logged in.

Comment partager ses ressources ?

Le réglage se fait dans les paramètres.

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. **Dégager:** Oui. **Évaluer:** Oui

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

Fichiers requis

edu/uha/miage/ReflectUtil.java

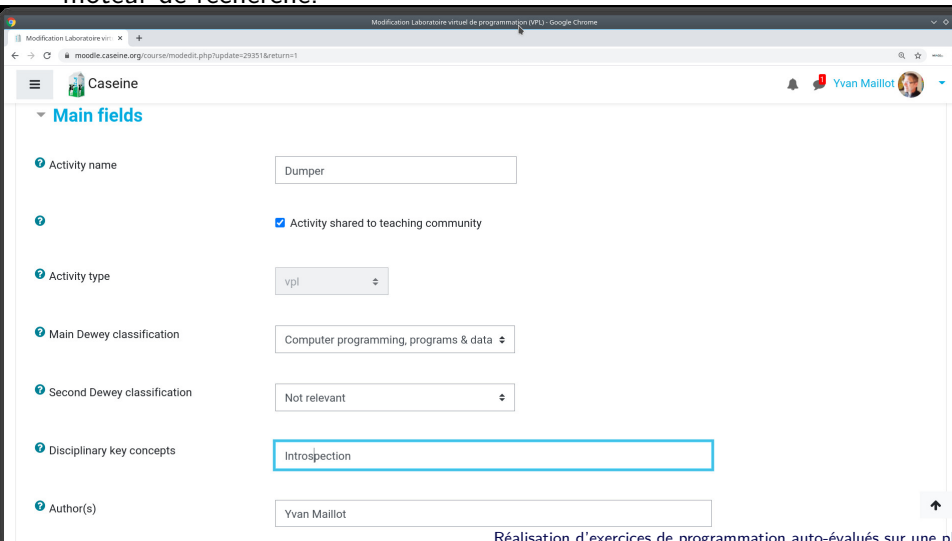
```
1 package edu.uha.miage;
2
3 import java.lang.reflect.InvocationTargetException;
4 import java.lang.reflect.Method;
5 import java.lang.reflect.Modifier;
6 import java.util.*;
7 import java.util.logging.Level;
8 import java.util.logging.Logger;
9
10 /**
11  * @author Yvan Maillot (yvan.maillot@uha.fr)
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 primitif 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

Comment partager ses ressources ?

Cocher la case « Activity shared to teaching community » suffit. Mais pour bien faire, il faut aussi renseigner les champs pour le moteur de recherche.



Modification Laboratoire virtuel de programmation (VPL) - Google Chrome

moodle.caseine.org/course/modedit.php?update=29351&return=1

Caseine

Yvan Maillot

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

Réalisation d'exercices de programmation auto-évalués sur une pl

Comment partager ses ressources ?

Définir la licence désirée.

Modification Laboratoire virtuel de programmation (VPL) - Google Chrome

moodle.caseine.org/course/modedit.php?update=29351&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

CC-BY-SA v4.0

Select one ...

CC-BY-SA v4.0

CC-BY-NC-SA v4.0

Public domain/BSD

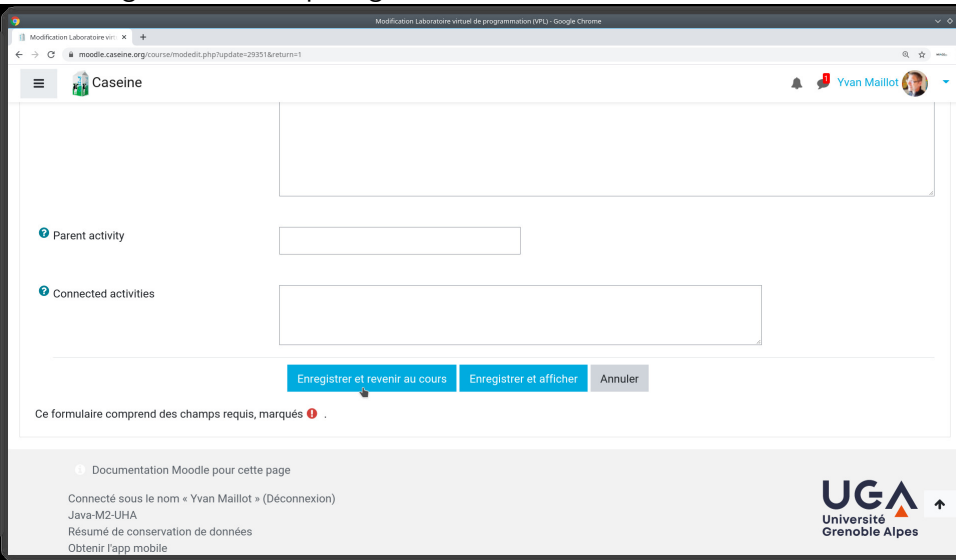
Licence Ouverte 2.0

Other

Strictly restricted

Comment partager ses ressources ?

Enregistrer et c'est partagé.



The screenshot shows a web browser window with the URL `moodle.caseine.org/course/modedit.php?update=29351&return=1`. The page header includes the Moodle logo, the course name 'Caseine', and the user 'Yvan Maillot'. The main content area contains two large text input fields, a 'Parent activity' field, and a 'Connected activities' field. At the bottom of the form, there are three buttons: 'Enregistrer et revenir au cours' (highlighted in blue), 'Enregistrer et afficher' (blue), and 'Annuler' (grey). Below the buttons, a message states: 'Ce formulaire comprend des champs requis, marqués !'. The footer of the page includes a link to 'Documentation Moodle pour cette page', the user's name 'Yvan Maillot', and the University of Grenoble Alpes logo (UGA).

Modification Laboratoire virtuel de programmation (VPL) - Google Chrome

moodle.caseine.org/course/modedit.php?update=29351&return=1

Caseine

Yvan Maillot

Parent activity

Connected activities

Enregistrer et revenir au cours Enregistrer et afficher Annuler

Ce formulaire comprend des champs requis, marqués !

Documentation Moodle pour cette page

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

UGA
Université
Grenoble Alpes

- Quels sont les bénéfices et les risques pour les étudiants ?
 - Gain en autonomie
 - Risque de se laisser déborder => **Suivi**
- Quels sont les bénéfices et les risques pour l'enseignant ?
 - Éviter ce cours magistral peu adapté
 - Risque de perdre le contact avec les étudiants => On verra
- Les supports de cours sont auto-suffisants
- Comment s'assurer de la bonne compréhension du cours par les étudiants avant l'accès au TD ?

Le suivi des étudiants

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

		dimanche 8 mars 2020, 14:45	2	2,21 / 20,00 Caché	Note automatique	dimanche 8 mars 2020, 14:45
		mercredi 19 février 2020, 16:33	8	7,05 / 20,00 Caché	Note automatique	mercredi 19 février 2020, 16:33
		lundi 9 mars 2020, 00:12	14	8,63 / 20,00 Caché	Note automatique	lundi 9 mars 2020, 00:12
		mercredi 19 février 2020, 12:40	27	11,53 / 20,00 Caché	Note automatique	mercredi 19 février 2020, 12:40
		mercredi 4 mars 2020, 18:44	27	14,00 / 20,00 Caché	Note automatique	mercredi 4 mars 2020, 18:44
		dimanche 8 mars 2020, 13:09	36	15,00 / 20,00 Caché	Note automatique	dimanche 8 mars 2020, 13:09
		samedi 7 mars 2020, 13:33	49	16,11 / 20,00 Caché	Note automatique	samedi 7 mars 2020, 13:33

- Quels sont les bénéfices et les risques pour les étudiants ?
 - Gain en autonomie
 - **Risque de se laisser déborder : limité => Grâce au suivi facilité par Moodle et Caséine**
- Quels sont les bénéfices et les risques pour l'enseignant ?
 - Éviter ce cours magistral peu adapté
 - Pas de perte de contact. Au contraire ! Un rapprochement => **Grâce au suivi facilité par Moodle et Caséine**
- Les supports de cours sont auto-suffisants
- Possible de s'assurer de la bonne compréhension du cours par les étudiants avant l'accès au TD ?
=> grâce aux exos dont l'évaluation est facilitée par le VPL de Caséine et au suivi

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.