

# COMPTE RENDU

POO - TP3 : Manipulation des classes en JAVA, les bases  
3e année Cybersécurité - École Supérieure d'Informatique et du  
Numérique (ESIN)  
Collège d'Ingénierie & d'Architecture (CIA)

**Étudiant :** HATHOUTI Mohammed Taha  
**Filière :** Cybersecurité  
**Année :** 2025/2026  
**Enseignants :** M.NAJIB  
**Date :** 17 octobre 2025

## Table des matières

<b>1</b>	<b>Partie 1</b>	<b>2</b>
1.1	Classe <i>ResultatModule</i> . . . . .	2
<b>2</b>	<b>Partie 2</b>	<b>4</b>
2.1	Classe <i>AnneeEtudiant</i> . . . . .	4
2.2	Classe <i>Main</i> : . . . . .	5

# 1 Partie 1

On se propose de définir une classe *ResultatModule* qui mémorise les notes obtenues dans un module. Ses attributs sont l'identifiant du module, une note de CC, une note de TP et une note d'examen. Tous les attributs sont déclarés privés.

## 1.1 Classe *ResultatModule*

```
1 package ex1;
2
3 public class ResultatModule {
4     private static int cmpt = 1;
5
6     private int idModule;
7     private double cc;
8     private double tp;
9     private double exam;
10
11     private static final double COEF_CC = 0.3;
12     private static final double COEF_TP = 0.2;
13     private static final double COEF_EXAM = 0.5;
14
15     public ResultatModule(double cc, double tp, double exam) {
16         // TODO Auto-generated constructor stub
17         this.idModule = cmpt++;
18         this.cc = cc;
19         this.tp = tp;
20         this.exam = exam;
21     }
22
23     public int getIdModule() {
24         return idModule;
25     }
26
27     public void setIdModule(int idModule) {
28         this.idModule = idModule;
29     }
30
31     public double getCc() {
32         return cc;
33     }
34
35     public void setCc(double cc) {
36         this.cc = cc;
37     }
38
39     public double getTp() {
40         return tp;
41     }
42
43     public void setTp(double tp) {
```

```
44         this.tp = tp;
45     }
46
47     public double getExam() {
48         return exam;
49     }
50
51     public void setExam(double exam) {
52         this.exam = exam;
53     }
54
55     public double calculeMoyenne() {
56         return (cc * COEF_CC) + (tp * COEF_TP) + (exam *
57             COEF_EXAM);
58     }
59
60     public boolean valideModule() {
61         return calculeMoyenne() >= 10;
62     }
63 }
```

## 2 Partie 2

On veut maintenant définir une classe *AnneeEtudiant* dont les attributs sont l'identifiant d'un étudiant, le nombre de modules auxquels il est inscrit et un tableau d'objets *ResultatModule*, qui représente les résultats obtenus aux différents modules composant son année. On suppose qu'un étudiant ne suit pas plus de 20 modules par an.

### 2.1 Classe *AnneeEtudiant*

```
1 package ex1;
2
3 public class AnneeEtudiant {
4     private int id_etudiant;
5     private int nb_modules;
6     private ResultatModule[] modules;
7
8     private static final int MAX_MODULES = 20;
9
10    public AnneeEtudiant(int id_etudiant) {
11        // TODO Auto-generated constructor stub
12        this.id_etudiant = id_etudiant;
13        this.nb_modules = 0;
14        this.modules = new ResultatModule[MAX_MODULES];
15    }
16
17    public int getIdEtudiant() {
18        return id_etudiant;
19    }
20
21    public void setIdEtudiant(int id_etudiant) {
22        this.id_etudiant = id_etudiant;
23    }
24
25    public int getNbModules() {
26        return nb_modules;
27    }
28
29    public void setNbModules(int nb_modules) {
30        this.nb_modules = nb_modules;
31    }
32
33    public ResultatModule[] getModules() {
34        return modules;
35    }
36
37    public void setModules(ResultatModule[] modules) {
38        this.modules = modules;
39    }
40
41    public double moyenneAnnee() {
42        if (nb_modules == 0) {
```

```

43         return 0;
44     }
45
46     double somme = 0;
47     int cmpt = 0;
48
49     for (int i = 0 ; i < modules.length ; i++) {
50         if (modules[i] != null) {
51             somme += modules[i].calculeMoyenne();
52             cmpt++;
53         }
54     }
55
56     return somme / cmpt;
57 }
58
59 public int nombreValides() {
60     int cmpt = 0;
61
62     for (int i = 0 ; i < modules.length ; i++) {
63         if (modules[i] != null) {
64             if (modules[i].valideModule()) {
65                 cmpt++;
66             }
67         }
68     }
69
70     return cmpt;
71 }
72
73 }

```

## 2.2 Classe *Main* :

```

1 package ex1;
2
3 import java.io.IOException;
4 import java.io.InputStream;
5 import java.io.PrintStream;
6
7 public class Main {
8     private static PrintStream ps = System.out;
9     private static InputStream is = System.in;
10    private static AnneeEtudiant[] etudiants = new AnneeEtudiant
        [100];
11    private static int nbEtudiants = 0;
12    private static AnneeEtudiant etudiantActuel;
13
14    public static void main(String[] args) throws IOException {
15        String choix;

```

```

16     boolean etudiantCree = false;
17
18     ps.println("== GESTION DES RESULTATS ETUDIANT ==");
19     ps.println("Operations disponibles :");
20     ps.println(" - creer : creer un nouvel etudiant");
21     ps.println(" - ajouter : ajouter un module");
22     ps.println(" - moyenne : afficher la moyenne de l'annee");
23     ;
24     ps.println(" - valides : afficher le nombre de modules valides");
25     ps.println(" - afficher : afficher tous les resultats");
26     ps.println(" - quit : quitter le programme");
27
28     do {
29         ps.print("\nQue voulez-vous faire ? ");
30         String[] input = Keyboard.readValues(is);
31         String ligne = String.join(" ", input).toLowerCase().trim();
32         choix = ligne.split("\\s+")[0];
33
34         switch (choix) {
35             case "creer":
36                 int idEtudiant = 0;
37                 boolean idValide = false;
38
39                 do {
40                     ps.print("\nEntrez l'ID de l'etudiant : ");
41                     String[] idInput = Keyboard.readValues(is);
42
43                     try {
44                         idEtudiant = Integer.parseInt(idInput[0]);
45                     }
46                     if (idEtudiant > 0) {
47                         idValide = true;
48                     } else {
49                         ps.println("Erreur: l'ID doit etre un nombre positif");
50                     }
51                 } catch (NumberFormatException e) {
52                     ps.println("Erreur: entrez uniquement des nombres !");
53                 }
54             } while (!idValide);
55
56             etudiantActuel = new AnneeEtudiant(idEtudiant);
57             etudiants[nbEtudiants] = etudiantActuel;
58             nbEtudiants++;
59             etudiantCree = true;
60             ps.println("\n===== RESULTAT =====");
61             ps.println("Etudiant cree avec l'ID : " + idEtudiant);

```

```

60         ps.println("=====");
61         break;
62
63     case "ajouter":
64         if (nbEtudiants == 0) {
65             ps.println("\nErreur: veuillez d'abord creer
66                 un etudiant avec la commande 'creer'");
67             break;
68         }
69
70         ps.println("\n===== LISTE DES ETUDIANTS
71             =====");
72         for (int i = 0; i < nbEtudiants; i++) {
73             ps.println((i + 1) + ". ID: " + etudiants[i].
74                 getIdEtudiant() +
75                 " | Modules: " + etudiants[i].
76                     getNbModules() +
77                 " | Moyenne: " + (etudiants[i].
78                     getNbModules() > 0 ?
79                     String.format("%.2f", etudiants[i]
80                         .moyenneAnnee()) : "N/A"));
81         }
82         ps.println("
83             =====");
84
85         int choixEtudiantAjout = 0;
86         boolean choixValideAjout = false;
87
88         do {
89             ps.print("\nChoisissez un etudiant (1-" +
90                 nbEtudiants + ") : ");
91             String[] choixInput = Keyboard.readValues(is)
92                 ;
93
94             try {
95                 choixEtudiantAjout = Integer.parseInt(
96                     choixInput[0]);
97                 if (choixEtudiantAjout >= 1 &&
98                     choixEtudiantAjout <= nbEtudiants) {
99                     choixValideAjout = true;
100                 } else {
101                     ps.println("Erreur: choisissez un
102                         numero entre 1 et " + nbEtudiants);
103                 }
104             } catch (NumberFormatException e) {
105                 ps.println("Erreur: entrez uniquement des
106                     nombres !");
107             }
108         } while (!choixValideAjout);

```



```

97      etudiantActuel = etudiants[choixEtudiantAjout -
98          1];
99      etudiantCree = true;
100
101      if (etudiantActuel.getNbModules() >= 20) {
102          ps.println("\nErreur: nombre maximum de
103              modules atteint (20)");
104          break;
105      }
106
107      double cc = 0, tp = 0, exam = 0;
108      boolean notesValides = false;
109
110      do {
111          ps.print("\nEntrez la note CC (sur 20) : ");
112          String[] ccInput = Keyboard.readValues(is);
113
114          ps.print("Entrez la note TP (sur 20) : ");
115          String[] tpInput = Keyboard.readValues(is);
116
117          ps.print("Entrez la note Exam (sur 20) : ");
118          String[] examInput = Keyboard.readValues(is);
119
120          try {
121              cc = Double.parseDouble(ccInput[0]);
122              tp = Double.parseDouble(tpInput[0]);
123              exam = Double.parseDouble(examInput[0]);
124
125              if (cc >= 0 && cc <= 20 && tp >= 0 && tp
126                  <= 20 && exam >= 0 && exam <= 20) {
127                  notesValides = true;
128              } else {
129                  ps.println("Erreur: les notes doivent
130                      etre entre 0 et 20");
131              }
132          } catch (NumberFormatException e) {
133              ps.println("Erreur: entrez uniquement des
134                  nombres !");
135          }
136      } while (!notesValides);
137
138      ResultatModule module = new ResultatModule(cc, tp
139          , exam);
140      etudiantActuel.getModules()[etudiantActuel.
141          getNbModules()] = module;
142      etudiantActuel.setNbModules(etudiantActuel.
143          getNbModules() + 1);
144
145      ps.println("\n===== RESULTAT =====");
146      ps.println("Module " + module.getIdModule() + "
147          ajoute avec succes pour l'etudiant " +

```

```

139         etudiantActuel.getIdEtudiant());
140     ps.println("Moyenne du module : " + String.format
141         ("%2f", module.calculerMoyenne()));
142     ps.println("Module valide : " + (module.
143         valideModule() ? "OUI" : "NON"));
144     ps.println("=====");
145     break;
146
147     case "moyenne":
148         if (!etudiantCree) {
149             ps.println("\nErreur: veuillez d'abord creer
150             un etudiant avec la commande 'creer'");
151             break;
152         }
153
154         if (etudiantActuel.getNbModules() == 0) {
155             ps.println("\nErreur: aucun module n'a ete
156             ajoute");
157             break;
158         }
159
160         ps.println("\n===== RESULTAT =====");
161         ps.println("Moyenne de l'annee : " + String.
162             format("%2f", etudiantActuel.moyenneAnnee()));
163         ps.println("=====");
164         break;
165
166     case "valides":
167         if (!etudiantCree) {
168             ps.println("\nErreur: veuillez d'abord creer
169             un etudiant avec la commande 'creer'");
170             break;
171         }
172
173         if (etudiantActuel.getNbModules() == 0) {
174             ps.println("\nErreur: aucun module n'a ete
175             ajoute");
176             break;
177         }
178
179         ps.println("\n===== RESULTAT =====");
180         ps.println("Nombre de modules valides : " +
181             etudiantActuel.nombreValides() + "/" +
182             etudiantActuel.getNbModules());
183         ps.println("=====");
184         break;
185
186     case "afficher":
187         if (nbEtudiants == 0) {
188             ps.println("\nErreur: aucun etudiant n'a ete
189             cree");
190         }

```

```

179         break;
180     }
181
182     ps.println("\n===== LISTE DES ETUDIANTS
183     =====");
184     for (int i = 0; i < nbEtudiants; i++) {
185         ps.println((i + 1) + ". ID: " + etudiants[i].
186             getIdEtudiant() +
187             " | Modules: " + etudiants[i].
188                 getNbModules() +
189             " | Moyenne: " + (etudiants[i].
190                 getNbModules() > 0 ?
191                 String.format("%.2f", etudiants[i]
192                     .moyenneAnnee()) : "N/A"));
193     }
194     ps.println("
195     =====");
196
197     int choixEtudiant = 0;
198     boolean choixValide = false;
199
200     do {
201         ps.print("\nChoisissez un etudiant (1- " +
202             nbEtudiants + ") : ");
203         String[] choixInput = Keyboard.readValues(is)
204             ;
205
206         try {
207             choixEtudiant = Integer.parseInt(
208                 choixInput[0]);
209             if (choixEtudiant >= 1 && choixEtudiant
210                 <= nbEtudiants) {
211                 choixValide = true;
212             } else {
213                 ps.println("Erreur: choisissez un
214                     numero entre 1 et " + nbEtudiants);
215             }
216         } catch (NumberFormatException e) {
217             ps.println("Erreur: entrez uniquement des
218                 nombres !");
219         }
220     } while (!choixValide);
221
222     AnneeEtudiant etudiantChoisi = etudiants[
223         choixEtudiant - 1];
224
225     if (etudiantChoisi.getNbModules() == 0) {
226         ps.println("\n===== RESULTAT =====");
227         ps.println("Cet etudiant n'a aucun module");
228         ps.println("=====");
229     }

```

```

216         break;
217     }
218
219     ps.println("\n===== RESULTATS DE L'ETUDIANT
220     =====");
221     ps.println("ID Etudiant : " + etudiantChoisi.
222     getIdEtudiant());
223     ps.println("Nombre de modules : " +
224     etudiantChoisi.getNbModules());
225     ps.println();
226
227     for (int i = 0; i < etudiantChoisi.getModules().
228     length; i++) {
229         if (etudiantChoisi.getModules()[i] != null) {
230             ResultatModule m = etudiantChoisi.
231             getModules()[i];
232             ps.println("Module " + m.getIdModule() +
233             " :");
234             ps.println("  CC: " + String.format("%.2f",
235             m.getCc()) +
236             " | TP: " + String.format("%.2f",
237             m.getTp()) +
238             " | Exam: " + String.format("%.2f",
239             m.getExam()));
240             ps.println("  Moyenne: " + String.format(
241             "%.2f", m.calculerMoyenne()) +
242             " | Valide: " + (m.
243             valideModule() ? "OUI" : "
244             NON"));
245             ps.println();
246         }
247     }
248
249     ps.println("Moyenne de l'annee : " + String.
250     format("%.2f", etudiantChoisi.moyenneAnnee()));
251     ps.println("Modules valides : " + etudiantChoisi.
252     nombreValides() + "/" + etudiantChoisi.
253     getNbModules());
254     ps.println("
255     =====")
256     ;
257     break;
258
259     case "quit":
260         ps.println("\nAu revoir !");
261         break;
262
263     default:
264         ps.println("\nOperation non reconnue. Utilisez:
265         creer, ajouter, moyenne, valides, afficher ou
266         quit");

```

```
248         break;
249     }
250 } while (!choix.equals("quit"));
251 }
252 }
```