

COMPTE RENDU

POO - TP2 : Syntaxe, Structure itérative, Structure conditionnelle,
Tableau

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 :	3 octobre 2025

1 Exercice I : Opérateurs

Classe *Operateurs* en static contenant une méthode *main* permettant d'implémenter une calculatrice interactive.

```
1 package ex1;
2
3 import java.io.BufferedReader;
4 import java.io.IOException;
5 import java.io.InputStream;
6 import java.io.InputStreamReader;
7 import java.io.PrintStream;
8
9 public class Operateurs {
10     private static final PrintStream ps = System.out;
11     private static final InputStream is = System.in;
12
13     public static void main(String[] args) throws IOException {
14         String choix;
15
16         ps.println("=== CALCULATRICE INTERACTIVE ===");
17         ps.println("Operations disponibles: addition, division,
18             perimetre, quit");
19
20         do {
21             ps.print("\nQuelle op ration voulez-vous effectuer?
22                 ");
23             String[] input = readValues(is);
24             String ligne = String.join(" ", input).toLowerCase().
25                 trim();
26             choix = ligne.split("\\s+")[0]; // Premier mot non-
27                 vide
28
29             switch (choix) {
30                 case "addition":
31                     effectuerAddition();
32                     break;
33                 case "division":
34                     effectuerDivision();
35                     break;
36                 case "perimetre":
37                     effectuerPerimetre();
38                     break;
39                 case "quit":
40                     ps.println("Au revoir!");
41                     break;
42                 default:
43                     ps.println("Op ration non reconnue. Utilisez:
44                         addition, division, perimetre, ou quit");
45             }
46         } while (!choix.equals("quit"));
```

```

42     }
43
44     private static void effectuerAddition() throws IOException {
45         while (true) { // Boucle jusqu' à succès
46             try {
47                 ps.print("Entrez votre addition (format: nombre +
48                     nombre): ");
49                 String[] entree = readValues(is);
50                 String line = String.join(" ", entree);
51
52                 if (!line.contains("+")) {
53                     ps.println("Erreur: L'addition doit contenir
54                         le symbole '+'");
55                     continue; // Redemander au lieu de return
56                 }
57
58                 String[] parties = line.split("\\+");
59                 if (parties.length != 2) {
60                     ps.println("Erreur: Format invalide. Utilisez
61                         : nombre + nombre");
62                     continue;
63                 }
64
65                 String aStr = parties[0].trim();
66                 String bStr = parties[1].trim();
67
68                 if (aStr.isEmpty() || bStr.isEmpty()) {
69                     ps.println("Erreur: Les deux nombres doivent
70                         tre s'cifi s");
71                     continue;
72                 }
73
74                 // Essayer de parser les deux nombres
75                 double a = Double.parseDouble(aStr);
76                 double b = Double.parseDouble(bStr);
77
78                 // D terminer si le r sultat doit tre int ou
79                 // double
80                 // Un nombre est consid r comme int s'il n'a
81                 // pas de point ET si sa valeur est
82                 // enti re
83                 boolean aEstInt = !aStr.contains(".") && a == (
84                     int) a;
85                 boolean bEstInt = !bStr.contains(".") && b == (
86                     int) b;
87
88                 double resultat = calculerSomme(a, b);
89
90                 if (aEstInt && bEstInt) {
91                     // Si le r sultat est aussi entier, l'
92                     // afficher comme int

```

```

84         if (resultat == (int) resultat) {
85             ps.print((int) a + " + " + (int) b + " = " + (int) resultat + "\n");
86         } else {
87             ps.print((int) a + " + " + (int) b + " = " + resultat + "\n");
88         }
89     } else if (aEstInt) {
90         if (resultat == (int) resultat) {
91             ps.print((int) a + " + " + b + " = " + (int) resultat + "\n");
92         } else {
93             ps.print((int) a + " + " + b + " = " + resultat + "\n");
94         }
95     } else if (bEstInt) {
96         if (resultat == (int) resultat) {
97             ps.print(a + " + " + (int) b + " = " + (int) resultat + "\n");
98         } else {
99             ps.print(a + " + " + (int) b + " = " + resultat + "\n");
100         }
101     } else {
102         if (resultat == (int) resultat) {
103             ps.print(a + " + " + b + " = " + (int) resultat + "\n");
104         } else {
105             ps.print(a + " + " + b + " = " + resultat + "\n");
106         }
107     }
108     break;
109
110 } catch (NumberFormatException e) {
111     ps.println("Erreur: Veuillez entrer des nombres valides");
112 }
113
114 }
115
116 private static void effectuerDivision() throws IOException {
117     while (true) {
118         try {
119             ps.print("Entrez votre division (format: nombre / nombre ou nombre nombre): ");
120             String[] entree = readValues(is);
121             String line = String.join(" ", entree);
122
123             if (!line.contains("/") && !line.contains(" "))
124                 continue;

```

```

124         ps.println("Erreur: La division doit contenir
125             le symbole '/' ou ' ');
126         continue;
127     }
128     String[] parties;
129     char ope;
130     if (line.contains("/")) {
131         parties = line.split("/");
132         ope = '/';
133     } else {
134         parties = line.split(" ");
135         ope = ' ';
136     }
137
138     if (parties.length != 2) {
139         ps.println("Erreur: Format invalide. Utilisez
140             : nombre / nombre ou nombre nombre");
141         continue;
142     }
143
144     String aStr = parties[0].trim();
145     String bStr = parties[1].trim();
146
147     if (aStr.isEmpty() || bStr.isEmpty()) {
148         ps.println("Erreur: Les deux nombres doivent
149             tre sp cifi s");
150         continue;
151     }
152
153     double a = Double.parseDouble(aStr);
154     double b = Double.parseDouble(bStr);
155
156     if (b == 0) {
157         ps.println("Erreur: Division par z ro
158             impossible");
159         continue;
160     }
161
162     double resultat = calculerDivision(a, b);
163
164     boolean aEstInt = !aStr.contains(".") && a == (
165         int) a;
166     boolean bEstInt = !bStr.contains(".") && b == (
167         int) b;
168
169     // Si le r sultat est un nombre entier, l'
170     // afficher comme int
171     if (resultat == (int) resultat) {
172         if (aEstInt && bEstInt) {
173             ps.print((int) a + " " + ope + " " + (int)

```

```

168         ) b + " = " + (int) resultat + "\n");
169     } else if (aEstInt) {
170         ps.print((int) a + " " + ope + " " + b +
171             " = " + (int) resultat + "\n");
172     } else if (bEstInt) {
173         ps.print(a + " " + ope + " " + (int) b +
174             " = " + (int) resultat + "\n");
175     } else {
176         ps.print((int) a + " " + ope + " " + b +
177             " = " + (int) resultat + "\n");
178     }
179
180     } else {
181         if (aEstInt && bEstInt) {
182             ps.print((int) a + " " + ope + " " + (int)
183                 b + " = " + resultat + "\n");
184         } else if (aEstInt) {
185             ps.print((int) a + " " + ope + " " + b +
186                 " = " + resultat + "\n");
187         } else if (bEstInt) {
188             ps.print(a + " " + ope + " " + (int) b +
189                 " = " + resultat + "\n");
190         } else {
191             ps.print((int) a + " " + ope + " " + b +
192                 " = " + resultat + "\n");
193         }
194     }
195     break;
196
197     } catch (NumberFormatException e) {
198         ps.println("Erreur: Veuillez entrer des nombres
199             valides");
200     }
201 }
202
203 private static void effectuerPerimetre() throws IOException {
204     while (true) {
205         try {
206             ps.print("Entrez le diam tre ou le rayon (format
207                 : d=valeur ou r=valeur): ");
208             String[] input = readValues(is);
209             String ligne = String.join(" ", input).
210                 toLowerCase().trim().replaceAll("\\s+", ""); //
211                 Supprimer tous les espaces
212
213             if (ligne.startsWith("d=")) {
214                 String valeurStr = ligne.substring(2).trim();
215                 if (valeurStr.isEmpty()) {
216                     ps.println("Erreur: Veuillez sp cifier
217                         une valeur pour le diam tre (ex: d=5)"

```

```

206         );
207         continue;
208     }
209     double diametre = Double.parseDouble(
210         valeurStr);
211     if (diametre <= 0) {
212         ps.println("Erreur: Le diam tre doit
213             tre positif");
214         continue;
215     }
216     double perimetre =
217         calculerPerimetreAvecDiametre(diametre);
218     ps.print("P rim tre du cercle avec
219         diam tre " + diametre + " = " + String.
220         format("%.2f", perimetre)
221         + "\n");
222     break;
223
224 } else if (ligne.startsWith("r=")) {
225     String valeurStr = ligne.substring(2).trim();
226     if (valeurStr.isEmpty()) {
227         ps.println("Erreur: Veuillez sp cifier
228             une valeur pour le rayon (ex: r=5)");
229         continue;
230     }
231     double rayon = Double.parseDouble(valeurStr);
232     if (rayon <= 0) {
233         ps.println("Erreur: Le rayon doit tre
234             positif");
235         continue;
236     }
237     double perimetre = calculerPerimetreAvecRayon
238         (rayon);
239     ps.print("P rim tre du cercle avec rayon "
240         + rayon + " = " + String.format("%.2f",
241         perimetre)
242         + "\n");
243     break;
244
245 } else {
246     ps.println("Erreur: Format invalide. Utilisez
247         : d=valeur ou r=valeur");
248     continue;
249 }
250
251 } catch (NumberFormatException e) {
252     ps.println("Erreur: Veuillez entrer une valeur
253         numerique valide");
254 }
255
256 }

```

```
244
245     public static double calculerSomme(double a, double b) {
246         return a + b;
247     }
248
249     public static double calculerDivision(double a, double b) {
250         return a / b;
251     }
252
253     public static double calculerPerimetreAvecDiametre(double
254         diametre) {
255         return Math.PI * diametre;
256     }
257
258     public static double calculerPerimetreAvecRayon(double rayon)
259     {
260         return 2 * Math.PI * rayon;
261     }
262
263     private static String[] readValues(InputStream in) throws
264     IOException {
265         InputStreamReader r = new InputStreamReader(in);
266         BufferedReader br = new BufferedReader(r);
267         String line = br.readLine();
268         String[] values = line.split(" ");
269         return values;
270     }
271 }
```


2 Exercice II : Structures Itératives

Pour cette exercice j'ai voulu utiliser l'orienté objet comme il se doit, j'ai donc créé deux classes. Une classe *StructresItératives* contenant les constructeurs ainsi que l'implémentation des deux méthodes qui s'occuperont chacune d'une question de l'exercice. Et une classe *Main* qui contient l'exécution ainsi que la lecture de l'entrée par une méthode *readValues* (une sorte de *Scanner* personnel).

2.1 Classe *StructuresItératives* :

```
1 package ex2;
2
3 public class StructuresIteratives {
4     private int nombre;
5     @SuppressWarnings("unused")
6     private int taille;
7
8     public StructuresIteratives(int nombre) {
9         this.nombre = nombre;
10        this.taille = 0;
11    }
12
13    public int[] diviseurs() {
14        int taille = 0;
15        for (int i = 1 ; i <= nombre ; i++) {
16            if (nombre % i == 0) {
17                taille ++;
18            }
19        }
20
21        int[] diviseurs = new int[taille];
22        int index = 0 ;
23        for (int i = 1 ; i <= nombre ; i++) {
24            if (nombre % i == 0) {
25                diviseurs[index] = i;
26                index++ ;
27            }
28        }
29
30        return diviseurs;
31    }
32
33    public int somme() {
34        int somme = 0;
35        int i = 1;
36
37        while (i < nombre) {
38            if (i % 2 == 0) {
39                somme += i;
40            }
41            i++;
```

```

42     }
43
44     return somme;
45 }
46
47 public int getNombre() {
48     // TODO Auto-generated method stub
49     return this.nombre;
50 }
51
52 public boolean estPremier() {
53     // TODO Auto-generated method stub
54     if (diviseurs().length == 2) {
55         return true;
56     }
57     return false;
58 }
59 }

```

2.2 Classe *Main* :

```

1 package ex2;
2
3 import java.io.BufferedReader;
4 import java.io.IOException;
5 import java.io.InputStream;
6 import java.io.InputStreamReader;
7 import java.io.PrintStream;
8
9 public class Main {
10     private static PrintStream ps = System.out;
11     private static InputStream is = System.in;
12     private static StructuresIteratives si;
13
14     public static void main(String[] args) throws IOException {
15         // TODO Auto-generated method stub
16         String choix;
17
18         ps.println("=== RECHERCHE DE DIVISEURS INTERACTIVE ===");
19         ps.println("Operations disponibles :");
20         ps.println(" - diviseurs : afficher tous ses diviseurs");
21         ps.println(" - somme : calculer la somme des nombres pairs inferieurs");
22         ps.println(" - quit : quitter le programme");
23
24         do {
25             ps.print("\nQue voulez-vous faire ? ");
26             String[] input = readValues(is);
27             String ligne = String.join(" ", input).toLowerCase().trim();

```

```

28      choix = ligne.split("\\s+")[0];
29
30      switch (choix) {
31      case "diviseurs":
32          boolean nb = false;
33          int nombre1;
34          do {
35              ps.print("\nEntrez un nombre entier positif :
36                  ");
37              String[] ip1 = readValues(is);
38              nombre1 = Integer.parseInt(ip1[0]);
39              if (nombre1 <= 0) {
40                  ps.println("Erreur: Le nombre doit etre
41                      positif");
42              } else {
43                  nb = true;
44              }
45
46              } while (nb == false);
47
48          si = new StructuresIteratives(nombre1);
49          afficherDiviseurs();
50          break;
51
52      case "somme":
53          boolean nb2 = false;
54          int nombre2;
55          do {
56              ps.print("\nEntrez un nombre entier positif :
57                  ");
58              String[] ip2 = readValues(is);
59              nombre2 = Integer.parseInt(ip2[0]);
60              if (nombre2 <= 0) {
61                  ps.println("Erreur: Le nombre doit etre
62                      positif");
63              } else {
64                  nb2 = true;
65              }
66
67              } while (nb2 == false);
68
69          si = new StructuresIteratives(nombre2);
70          afficherSomme();
71          break;
72
73      case "quit":
74          ps.println("\nAu revoir !");
75          break;
76
77      default:

```

```

75         ps.println("\nOperation non reconnue. Utilisez:
76             diviseurs, somme, ou quit");
77         break;
78     }
79     } while (!choix.equals("quit"));
80 }
81
82 private static void afficherDiviseurs() {
83     if (si == null) {
84         ps.println("Erreur: objet non initialise");
85         return;
86     }
87
88     ps.println("\n===== RESULTAT =====");
89     ps.println("Nombre analyse: " + si.getNombre());
90
91     int[] diviseurs = si.diviseurs();
92     ps.print("Diviseurs: [");
93     for (int i = 0; i < diviseurs.length; i++) {
94         if (i == 0) {
95             ps.print(diviseurs[i]);
96         } else {
97             ps.print(", " + diviseurs[i]);
98         }
99     }
100     ps.println("]");
101
102     ps.println("Nombre total de diviseurs: " + diviseurs.
103         length);
104
105     if (si.estPremier()) {
106         ps.println("-> " + si.getNombre() + " est un nombre
107             PREMIER !");
108     }
109
110     ps.println("=====");
111 }
112
113 private static void afficherSomme() {
114     if (si == null) {
115         ps.println("Erreur: objet non initialise");
116         return;
117     }
118
119     ps.println("\n===== RESULTAT =====");
120     ps.println("Nombre limite: " + si.getNombre());
121
122     int somme = si.somme();
123     ps.println("Somme des nombres pairs inferieurs a " + si.
124         getNombre() + ": " + somme);

```

```

122     // Afficher les nombres pairs
123     ps.print("Nombres pairs: ");
124     for (int i = 2; i < si.getNombre(); i += 2) {
125         ps.print(i);
126         if ((i + 2) < si.getNombre()) {
127             ps.print(", ");
128         }
129     }
130     ps.println();
131
132     ps.println("=====");
133 }
134
135 private static String[] readValues(InputStream in) throws
    IOException {
136     InputStreamReader r = new InputStreamReader(in);
137     BufferedReader br = new BufferedReader(r);
138     String line = br.readLine();
139     String[] values = line.split(" ");
140     return values;
141 }
142
143 }

```

3 Exercice III : Manipulation des Tableaux

Pour cette exercice j'ai utilisé deux classes. Une classe *ManipulationTableau* contenant les constructeurs ainsi que l'implementation des deux méthodes qui s'occuperont chacune d'une question de l'exercice. Et une classe *Main* qui contient l'execution ainsi que la lecture de l'entrée par une methode *readValues* (une sorte de *Scanner* personnel).

3.1 Classe *ManipulationTableau* :

```
1 package ex3;
2
3 import java.util.Random;
4
5 public class ManipulationTableau {
6     public int[] tableau;
7     public int taille;
8     public int max;
9
10    public ManipulationTableau() {
11        this.taille = 20;
12        this.max = 10000;
13        this.tableau = generateArray(taille, max);
14    }
15
16    public ManipulationTableau(int[] tableau) {
17        this.tableau = tableau;
18        this.taille = tableau.length;
19        this.max = 0;
20    }
21
22    public ManipulationTableau(int taille) {
23        this.taille = taille;
24        this.max = 10000;
25        this.tableau = generateArray(taille, max);
26    }
27
28    public ManipulationTableau(int taille, int max) {
29        this.taille = taille;
30        this.max = max;
31        this.tableau = generateArray(taille, max);
32    }
33
34    private int[] generateArray(int taille, int max) {
35        // TODO Auto-generated method stub
36        int[] tab = new int[taille];
37        Random rand = new Random();
38
39        for (int i = 0 ; i < taille ; i++) {
40            tab[i] = rand.nextInt(max + 1);
41        }
42    }
```

```

43         return tab;
44     }
45
46     public int sommeValeurTableau() {
47         int val = 0;
48
49         for (int i = 0 ; i < taille ; i++ ) {
50             val += tableau[i];
51         }
52
53         return val;
54     }
55
56     public int[] getTableau() {
57         return this.tableau;
58     }
59
60     public int getTaille() {
61         return this.taille;
62     }
63
64     public int getMax() {
65         return this.max;
66     }
67
68 }

```

3.2 Classe *Main* :

```

1 package ex3;
2
3 import java.io.BufferedReader;
4 import java.io.IOException;
5 import java.io.InputStream;
6 import java.io.InputStreamReader;
7 import java.io.PrintStream;
8
9 public class Main {
10     private static final PrintStream ps = System.out;
11     private static final InputStream is = System.in;
12     private static ManipulationTableau mt;
13
14     public static void main(String[] args) throws IOException {
15         // TODO Auto-generated method stub
16         String choix;
17
18         ps.println("=== MANIPULATION INTERACTIVE DE TABLEAU ===");
19         ;
20         ps.println("Operations disponibles :");
21         ps.println("  - taille : choisir uniquement la taille");

```

```

21     ps.println("    - taille_max : choisir la taille et la
        valeur maximale");
22     ps.println("    - tableau : entrer votre propre tableau");
23     ps.println("    - aleatoire : generer aleatoirement un
        tableau");
24     ps.println("    - quit : quitter le programme");
25
26     do {
27         ps.print("\nQue voulez-vous faire ? ");
28         String[] input = readValues(is);
29         String ligne = String.join(" ", input).toLowerCase().
            trim();
30         choix = ligne.split("\\s+")[0]; // Premier mot non-
            vide
31
32         switch (choix) {
33             case "taille":
34                 ps.print("\nEntrez la taille : ");
35
36                 String[] ip1 = readValues(is);
37                 int taille = Integer.parseInt(ip1[0]);
38
39                 mt = new ManipulationTableau(taille);
40                 afficherTableauetSomme();
41                 break;
42
43             case "taille_max":
44                 ps.print("\nEntrez la taille : ");
45
46                 String[] ip2 = readValues(is);
47                 int taille2 = Integer.parseInt(ip2[0]);
48
49                 ps.print("Entrez la valeur maximale : ");
50                 String[] ip3 = readValues(is);
51                 int max = Integer.parseInt(ip3[0]);
52
53                 mt = new ManipulationTableau(taille2, max);
54                 afficherTableauetSomme();
55                 break;
56
57             case "tableau":
58                 ps.print("\nEntrez les valeurs separees par des
                    espaces : ");
59                 String[] ip4 = readValues(is);
60                 int[] tableau = new int[ip4.length];
61
62                 for (int i = 0; i < ip4.length; i++) {
63                     tableau[i] = Integer.parseInt(ip4[i]);
64                 }
65
66                 mt = new ManipulationTableau(tableau);

```



```

67         afficherTableauetSomme();
68         break;
69
70     case "aleatoire":
71         mt = new ManipulationTableau();
72         afficherTableauetSomme();
73         break;
74
75     case "quit":
76         ps.println("\nAu revoir !");
77         break;
78
79     default:
80         ps.println("\nOperation non reconnue. Utilisez:
81             taille, taille_max, tableau, aleatoire, ou quit
82             ");
83         break;
84     }
85 } while (!choix.equals("quit"));
86
87 private static void afficherTableauetSomme() {
88     // TODO Auto-generated method stub
89     if (mt == null) {
90         ps.println("Erreur: tableau non initialis ");
91         return;
92     }
93
94     ps.println("\n===== RESULTAT =====");
95     ps.println("Taille du tableau: " + mt.getTaille());
96
97     if (mt.getMax() > 0) {
98         ps.println("Valeur maximale: " + mt.getMax());
99     }
100
101     if (mt.getTaille() <= 20) {
102         ps.print("Contenu: [");
103         int[] tab = mt.getTableau();
104         for (int i = 0; i < tab.length; i++) {
105             ps.print(tab[i]);
106             if (i < tab.length - 1) {
107                 ps.print(", ");
108             }
109         }
110         ps.println("]");
111     } else {
112         ps.println("Tableau trop grand pour etre affiche (
113             taille > 20)");
114         ps.println("Apercu des 5 premieres valeurs: ");
115         int[] tab = mt.getTableau();
116         ps.print("[");

```

```

115         for (int i = 0; i < 5; i++) {
116             ps.print(tab[i]);
117             if (i < 4) ps.print(", ");
118         }
119         ps.println(", ...]");
120     }
121
122     int somme = mt.sommeValeurTableau();
123     ps.println("Somme des valeurs: " + somme);
124     ps.println("=====");
125 }
126
127 private static String[] readValues(InputStream in) throws
IOException {
128     InputStreamReader r = new InputStreamReader(in);
129     BufferedReader br = new BufferedReader(r);
130     String line = br.readLine();
131     String[] values = line.split(" ");
132     return values;
133 }
134
135 }

```

4 Exercice IV : Switch Case

Pour cet exercice j'ai utilisé deux classes. Une classe *SwitchCase* contenant les constructeurs ainsi que l'implementation des méthodes de conversion de date et de validation. Et une classe *Main* qui contient l'exécution ainsi que la lecture de l'entrée par une méthode *readValues* (une sorte de Scanner personnel).

4.1 Classe *SwitchCase* :

```
1 package ex4;
2
3 public class SwitchCase {
4     private int jour;
5     private int mois;
6     private int annee;
7
8     public SwitchCase(int jour, int mois, int annee) {
9         this.jour = jour;
10        this.mois = mois;
11        this.annee = annee;
12    }
13
14    public String convertirDate() {
15        String moisEnLettres = "";
16
17        switch (mois) {
18            case 1:
19                moisEnLettres = "Janvier";
20                break;
21            case 2:
22                moisEnLettres = "Fvrier";
23                break;
24            case 3:
25                moisEnLettres = "Mars";
26                break;
27            case 4:
28                moisEnLettres = "Avril";
29                break;
30            case 5:
31                moisEnLettres = "Mai";
32                break;
33            case 6:
34                moisEnLettres = "Juin";
35                break;
36            case 7:
37                moisEnLettres = "Juillet";
38                break;
39            case 8:
40                moisEnLettres = "Ao t";
41                break;
42            case 9:
```

```

43         moisEnLettres = "Septembre";
44         break;
45     case 10:
46         moisEnLettres = "Octobre";
47         break;
48     case 11:
49         moisEnLettres = "Novembre";
50         break;
51     case 12:
52         moisEnLettres = "D cembre";
53         break;
54     default:
55         moisEnLettres = "Mois invalide";
56         break;
57     }
58
59     String affichageAnnee = new String();
60     if (annee < 0) {
61         affichageAnnee = (-annee) + " Avant J sus-Christ";
62     } else {
63         affichageAnnee = String.valueOf(annee);
64     }
65
66     if (jour == 1) {
67         return "1er " + moisEnLettres + " " + affichageAnnee;
68     }
69
70     return jour + " " + moisEnLettres + " " + affichageAnnee;
71 }
72
73 private static boolean estBissextile(int annee) {
74     if (annee < 0) {
75         annee = -annee; // g rer ann es n gatives
76     }
77     return (annee % 4 == 0 && annee % 100 != 0) || (annee %
78         400 == 0);
79 }
80
81 public static boolean dateValide(int jour, int mois, int
82     annee) {
83     if (jour <= 0 || mois <= 0) {
84         return false;
85     }
86     if (mois < 1 || mois > 12) {
87         return false;
88     }
89
90     int maxJour;
91
92     switch (mois) {
93         case 4:

```

```

92     case 6:
93     case 9:
94     case 11:
95         maxJour = 30;
96         break;
97     case 2:
98         maxJour = estBissextile(annee) ? 29 : 28;
99         break;
100    default:
101        maxJour = 31;
102        break;
103    }
104
105    return jour >= 1 && jour <= maxJour;
106 }
107 }

```

4.2 Classe *Main* :

```

1 package ex4;
2
3 import java.io.BufferedReader;
4 import java.io.IOException;
5 import java.io.InputStream;
6 import java.io.InputStreamReader;
7 import java.io.PrintStream;
8
9 public class Main {
10     private static PrintStream ps = System.out;
11     private static InputStream is = System.in;
12     private static SwitchCase sc;
13
14     public static void main(String[] args) throws IOException {
15         // TODO Auto-generated method stub
16         String choix;
17
18         ps.println("=== CONVERTISSEUR DE DATE INTERACTIF ===");
19         ps.println("Operations disponibles :");
20         ps.println(" - date : convertir une date jj/mm/aaaa en
21             texte");
22         ps.println(" - quit : quitter le programme");
23
24         do {
25             ps.print("\nQue voulez-vous faire ? ");
26             String[] input = readValues(is);
27             String ligne = String.join(" ", input).toLowerCase().
28                 trim();
29             choix = ligne.split("\\s+")[0];
30
31             switch (choix) {

```

```

30     case "date":
31         boolean date = false;
32         int jour;
33         int mois;
34         int annee;
35         String[] parts;
36         do {
37             ps.print("\nEntrez une date au format jj/mm/
38                 aaaa : ");
39             String[] ip = readValues(is);
40             parts = ip[0].split("/");
41
42             if (parts.length != 3) {
43                 ps.println("Erreur: format invalide (
44                     attendu jj/mm/aaaa)");
45                 continue;
46             }
47
48             try {
49                 jour = Integer.parseInt(parts[0]);
50                 mois = Integer.parseInt(parts[1]);
51                 annee = Integer.parseInt(parts[2]);
52
53                 if (!SwitchCase.dateValide(jour, mois,
54                     annee)) {
55                     ps.println("Erreur: date invalide");
56                 } else {
57                     date = true;
58                 }
59             } catch (NumberFormatException e) {
60                 ps.println("Erreur: entrez uniquement des
61                     nombres !");
62             }
63
64             } while (!date);
65
66         jour = Integer.parseInt(parts[0]);
67         mois = Integer.parseInt(parts[1]);
68         annee = Integer.parseInt(parts[2]);
69
70         sc = new SwitchCase(jour, mois, annee);
71         ps.println("\n===== RESULTAT =====");
72         ps.println("Date convertie : " + sc.convertirDate
73             ());
74         ps.println("=====");
75         break;
76
77     case "quit":
78         ps.println("\nAu revoir !");
79         break;

```

```

76
77         default:
78             ps.println("\nOperation non reconnue. Utilisez:
79                 date ou quit");
80             break;
81         }
82     } while (!choix.equals("quit"));
83
84     private static String[] readValues(InputStream in) throws
85         IOException {
86         InputStreamReader r = new InputStreamReader(in);
87         BufferedReader br = new BufferedReader(r);
88         String line = br.readLine();
89         String[] values = line.split(" ");
90         return values;
91     }
92 }

```

5 Exercice V : La Devinette

Pour cet exercice j'ai utilisé trois classes. Une classe *Guess* contenant le nombre à deviner et le nombre de tentatives maximales. Une classe *Game* qui gère la logique du jeu et les propositions. Et une classe *Main* qui contient l'exécution ainsi que la lecture de l'entrée par une méthode `readValues` (une sorte de *Scanner* personnel).

5.1 Classe *Guess* :

```
1 package ex5;
2
3 public class Guess {
4     private int nombreADeviner;
5     private int tentativesMax;
6
7     public Guess(int nombreADeviner, int tentativesMax) {
8         // TODO Auto-generated constructor stub
9         this.nombreADeviner = nombreADeviner;
10        this.tentativesMax = tentativesMax;
11    }
12
13    public int getNombreADeviner() {
14        return this.nombreADeviner;
15    }
16
17    public int getTentativesMax() {
18        return this.tentativesMax;
19    }
20
21 }
```

5.2 Classe *Game* :

```
1 package ex5;
2
3 public class Game {
4     private Guess guess;
5     private int tentativesRestantes;
6
7     public Game(Guess guess) {
8         // TODO Auto-generated constructor stub
9         this.guess = guess;
10        this.tentativesRestantes = guess.getTentativesMax();
11    }
12
13    public String proposition(int proposition) {
14        tentativesRestantes--;
15
16        if (proposition == guess.getNombreADeviner()) {
```



```

17         return "F licitation ! vous venez de trouver le
           nombre";
18     } else if (proposition < guess.getNombreADeviner()) {
19         return "La valeur recherch e est plus grande";
20     } else {
21         return "La valeur recherch e est plus petite";
22     }
23 }
24
25 public boolean encoreDesTentatives() {
26     return tentativesRestantes > 0;
27 }
28
29 public int getTentativesRestantes() {
30     return tentativesRestantes;
31 }
32
33 }

```

5.3 Classe *Main* :

```

1 package ex5;
2
3 import java.io.BufferedReader;
4 import java.io.IOException;
5 import java.io.InputStream;
6 import java.io.InputStreamReader;
7 import java.io.PrintStream;
8 import java.util.Random;
9
10 public class Main {
11     private static PrintStream ps = System.out;
12     private static InputStream is = System.in;
13
14     public static void main(String[] args) throws IOException {
15         // TODO Auto-generated method stub
16         ps.println("=== JEU DU NOMBRE      DEVINER ===");
17         ps.println("Vous avez 3 tentatives pour deviner le nombre
18             .");
19         ps.println("Attention : vous devez entrer un seul nombre
20             entier      chaque fois.\n");
21
22         boolean rejouer = true;
23
24         while (rejouer) {
25             Random rand = new Random();
26             Guess guess = new Guess(rand.nextInt(101), 3);
27             Game game = new Game(guess);
28
29             int erreursConsecutives = 0;

```

```

28 //BufferedReader br = new BufferedReader(new
29     InputStreamReader(is));
30
31 while (game.encoreDesTentatives()) {
32     ps.print("\nEntrez un nombre : ");
33     String[] input = readValues(is);
34
35     if (input.length == 1 && input[0].toLowerCase().
36         equals("quit")) {
37         ps.println("Vous avez quitt le jeu.
38             Retourne jouer s rieusement !");
39         return;
40     }
41
42     if (input.length != 1) {
43         ps.println("Erreur : entrez exactement UN
44             seul nombre !");
45         erreursConsecutives++;
46     } else {
47         try {
48             int proposition = Integer.parseInt(input
49                 [0]);
50             erreursConsecutives = 0;
51
52             String resultat = game.proposition(
53                 proposition);
54             ps.println(resultat);
55
56             if (proposition == guess.
57                 getNombreADeviner()) {
58                 break;
59             }
60
61             ps.println("Tentatives restantes : " +
62                 game.getTentativesRestantes());
63
64             } catch (NumberFormatException e) {
65                 ps.println("Erreur : ce n'est pas un
66                     nombre entier !");
67                 erreursConsecutives++;
68             }
69         }
70     }
71
72     if (erreursConsecutives >= 3) {
73         ps.println("\nTrop de b tises ! Le jeu se
74             termine. Retourne jouer s rieusement !");
75         return;
76     }
77 }
78
79 if (!game.encoreDesTentatives())

```

```

69         && game.proposition(guess.getNombreADeviner()
70             ) != "Bravo ! Vous avez trouv le nombre."
71             ) {
72             ps.println("\nDommage ! Vous avez puis vos
73                 tentatives.");
74             ps.println("Le nombre deviner tait : " +
75                 guess.getNombreADeviner());
76         }
77
78         ps.print("\nVoulez-vous rejouer ? (oui/non) : ");
79         String[] replayInput = readValues(is);
80         if (replayInput.length == 0 || !replayInput[0].
81             toLowerCase().equals("oui")) {
82             rejouer = false;
83             ps.println("\nMerci d'avoir jou ! bient t !
84                 ");
85         }
86     }
87
88     ps.println("\n=== FIN DU JEU ===");
89 }
90
91 private static String[] readValues(InputStream in) throws
92     IOException {
93     InputStreamReader r = new InputStreamReader(in);
94     BufferedReader br = new BufferedReader(r);
95     String line = br.readLine();
96     String[] values = line.split(" ");
97     return values;
98 }

```