

Utils.java

```

1 package Serveur.maths;
2
3 public abstract class Utils {
4     /**
5      * réalise un mappage de valeur entre min incluse et max exclue c'est une
6      * sorte de modulo amélioré
7      *
8      * exemple: <br>
9      * map(angle, -PI, PI) réalise ramène un angle entre -PI et PI
10    */
11    public static double map(double value, double min, double max) {
12        double modulo = max - min;
13
14        if (value < min)
15            return ((value - min) % modulo + modulo) % modulo + min;
16
17        return (value - min) % modulo + min;
18    }
19
20    public static double map(double x, double xmin, double xmax, double ymin,
21        double ymax) {
22        return (x - xmin) / (xmax - xmin) * (ymax - ymin) + ymin;
23    }
24
25    /**
26     * Créer une liste avec un nombre de dimensions spécifiable. <br>
27     * Exemple avec sizes = {4, 2, 3, 1} <br>
28     * La fonction va renvoyer : new Object[4][2][3][1]
29     */
30    public static Object[] createList(final int[] sizes) {
31        Object[] res = new Object[sizes[0]];
32
33        if (sizes.length > 1) {
34            int[] subSizes = new int[sizes.length - 1];
35
36            for (int i = 0; i < subSizes.length; i++)
37                subSizes[i] = sizes[i + 1];
38
39            for (int i = 0; i < res.length; i++)
40                res[i] = createList(subSizes);
41        } else {
42            for (int i = 0; i < res.length; i++)
43                res[i] = null;
44        }
45
46        return res;
47    }
48
49    public static int dimensions(Object liste) {
50        try {
51            return 1 + dimensions(((Object[]) liste)[0]);
52        } catch (ClassCastException | NullPointerException e) {
53            return 0;
54        }
55    }
56
57    public static String listStr(Object list, int nbDims) {
58        return listStr(list, nbDims, 1);
59    }
60
61    public static String listStr(Object list, int nbDims, int sousCouche) {
62        String res = "[";

```

Utils.java

```
63     int length = ((Object[]) list).length;
64
65     if (nbDims > 1) {
66         String spaces = "";
67
68         for (int i = 0; i < sousCouche; i++)
69             spaces += " ";
70
71         for (int i = 0; i < length; i++) {
72             res += listStr(((Object[]) list)[i], nbDims - 1,
73                 sousCouche + 1);
74
75             if (i < length - 1)
76                 res += "\n" + spaces;
77         }
78     } else {
79         for (int i = 0; i < length; i++) {
80             res += ((Object[]) list)[i];
81
82             if (i < length - 1)
83                 res += ", ";
84         }
85     }
86
87     return res + "];";
88 }
89 }
```