```java
import java.util.Random;

public class Recuit {
    private static Random generateur = new Random();

    /*********************************************************/
    /* Parametres du recuit */
    private static final int nbTransitions = 2000;
    private static final double alpha = 0.995;
    private static final boolean minimisation = true;
    /*********************************************************/

    /* dimension du probleme */
    private static final int DIMENSION = 100;

    // principe d'acceptation en maximisation
    private boolean accept(double yi, double yj, double temp, boolean minimiser) {
        boolean res = false;
        double proba, tirage;
        if (minimiser) {
            if (yj < yi) {
                res = true;
            } else {
                proba = Math.exp((yi - yj) / temp);
                tirage = Math.random(); // 0-1
                if (tirage < proba)
                    res = true;
            }
        } else {
            if (yj > yi) {
                res = true;
            } else {
                proba = Math.exp((yj - yi) / temp);
                tirage = Math.random(); // 0-1
                if (tirage < proba)
                    res = true;
            }
        }
        return res;
    }

    // ************************************
    // Heat Up
    // ************************************
    public double heatUpLoop() { // HeatUp heat = new HeatUp();
        int acceptCount = 0;
        double yi = 0, yj;
        double T = 0.01, tauxAccept = 0.0;
        Etat xi = new Etat(DIMENSION);

        do {
            acceptCount = 0;
```

```
53            for (int i = 0; i < nbTransitions; i++) {
54
55                // generation d'un point de l'espace d'etat
56                xi.initAleatEtat();
57                yi = xi.calculCritere();
58
59                // generation d'un voisin
60                xi.genererVoisin();
61                yj = xi.calculCritere();
62
63                if (accept(yi, yj, T, minimisation))
64                    acceptCount++;
65                tauxAccept = (double) acceptCount / (double)
  nbTransitions;
66            }
67            T = T * 1.1;
68            System.out.println("T= " + T + " tauxAccept= " + tauxAccept +
  " currentCost= " + yi);
69        } while (tauxAccept < 0.8);
70        return T;
71    }
72
73    // ****************************************
74    // COOLING
75    // ****************************************
76
77    public Etat coolingLoop(double Tinit) { // HeatUp heat = new HeatUp();
78        double yi = 0.0, yj = 0.0; // , proba;
79        double T = Tinit;
80        Etat xi = new Etat(DIMENSION);
81
82        xi.initAleatEtat();
83        yi = xi.calculCritere();
84        do {
85            for (int i = 0; i < nbTransitions; i++) {
86                xi.genererVoisin();
87                yj = xi.calculCritere();
88                if (accept(yi, yj, T, minimisation)) {
89                    yi = yj;
90                } else {
91                    xi.comeBack();
92                }
93            }
94            T = T * alpha;
95            System.out.println("T= " + T + " valeur critere " + yi);
96            System.out.println(xi.afficherEtat());
97        } while (T > 0.0001 * Tinit);
98        return xi;
99    }
100
101   // ****************************************
102   // MAIN
103   // ****************************************
```

```
104     public static void main(String args[]) {
105 
106         double temperature;
107         Recuit monRecuit = new Recuit();
108         // generation des données
109         System.out.println("**********************Generation des donnee
    ***************");
110         // Data.genererObjets(DIMENSION);
111         Data.genererVillesCercle(DIMENSION);
112         // Data.genererVilles(DIMENSION);
113         // Data.genererAvions(DIMENSION);
114 
115         System.out.println("***************************Chauffage
    ****************");
116         temperature = monRecuit.heatUpLoop();
117         System.out.println("======================Refroidissement
    =============");
118         monRecuit.coolingLoop(temperature);
119     }// end main
120 }// End class HeatUp2
121 
```