

```

1  #include "chronique.h"
2
3  /* CONSTRUCTEURS */
4
5  // Constructeur simple
6  chronique::chronique(){
7      nom="";
8      event_verify=false;
9      nb_validation=0;
10     heure_validation.clear();
11     nb_contraintes_restantes = 0;
12     nb_contraintes_total = 0;
13     contraintes.clear();
14 }
15
16 // Constructeur évolué, créé la chronique à partir des données
17 chronique::chronique(string label, int nb_contraintes, vector<contrainte> vect_contraintes){
18     nom=label;
19     event_verify=false;
20     nb_validation=0;
21     heure_validation.clear();
22     nb_contraintes_restantes = nb_contraintes;
23     nb_contraintes_total = nb_contraintes;
24     contraintes = vect_contraintes;
25 }
26
27 /* ASSESSEURS */
28
29 // Assesseur, ressort le nom de la chronique
30 string chronique::get_nom(){
31     return nom;
32 }
33
34 // Assesseur, ressort l'état de la chronique (pour les chroniques de chroniques)
35 bool chronique::get_event(){
36     return event_verify;
37 }
38
39 // Assesseur, ressort le nombre de validation
40 int chronique::get_nb_valid(){
41     return nb_validation;
42 }
43
44 // Assesseur, ressort le nombre de contraintes restantes
45 int chronique::get_contraintes_restantes(){
46     return nb_contraintes_restantes ;
47 }
48
49
50 // Assesseur, ressort le nombre de contraintes totales
51 int chronique::get_contraintes_total(){
52     return nb_contraintes_total ;
53 }
54
55
56 // Assesseur, ressort le tableau des heures de validation
57 std::vector<int> chronique::get_h_valid(){
58     return heure_validation;
59 }
60
61 // Assesseur, ressort l'heure de la dernière validation
62 int chronique::get_last_h_valid(){
63     return heure_validation.back();
64 }
65
66 // Assesseur, ressort le tableau des contraintes

```

```

67     std::vector<contrainte> chronique::get_contraintes(){
68         return contraintes;
69     }
70
71     // Mutateur, change la valeur du nombre de contraintes restantes de la chronique
72     void chronique::set_contraintes_restantes(int nb){
73         nb_contraintes_restantes = nb;
74     }
75
76     // Mutateur, change le label de l'élément
77     void chronique::set_nom(string name){
78         nom = name;
79     }
80
81     // Mutateur, change la valeur d'évènement de la chronique
82     void chronique::set_event(bool event){
83         event_verify = event;
84     }
85
86     // Mutateur, change la valeur du nombre de validation de la chronique
87     void chronique::set_valid(int nb){
88         nb_validation = nb;
89     }
90
91     // Mutateur, ajoute une valeur dans les heures de validation de la chronique
92     void chronique::set_h_valid(int heure){
93         heure_validation.push_back(heure);
94     }
95
96     // Reinitialisateur, remet à 0 la valeur du nombre de validation
97     void chronique::reset_valid(){
98         nb_validation = 0;
99     }
100
101     // Reinitialisateur, remet à 0 le tableau des heures de validation
102     void chronique::reset_h_valid(){
103         heure_validation.clear();
104     }
105
106     // Reinitialisateur, remet au maximum la valeur des contraintes restantes à valider
107     void chronique::reset_nb_contraintes(){
108         nb_contraintes_restantes=nb_contraintes_total;
109     }
110
111     // Reinitialisateur, réinitialise tous les attributs
112     void chronique::reinit_chronique_all(){
113         nom="";
114         event_verify=false;
115         nb_validation=0;
116         heure_validation.clear();
117         nb_contraintes_restantes = nb_contraintes_total;
118         contraintes.clear();
119     }
120
121     /* METHODES */
122
123     // Affiche certains éléments dans la console
124     void chronique::afficher(){
125         cout<<"nom = "<<nom<<endl;
126         cout<<"nombre de validation = "<<nb_validation<<endl;
127         if (nb_validation != 0) {
128             cout<<"derniere validation = "<<heure_validation.back()<<endl;
129         }
130         cout<<"evenement detectable = "<<event_verify<<endl;
131         cout<<"nombre de contraintes total = "<<nb_contraintes_total<<endl;
132         cout<<"nombre de contraintes restantes = "<<nb_contraintes_restantes<<endl;

```

```

133     }
134
135     // Affiche les heures de validation de la chronique dans le terminal
136     void chronique::afficher_heures_validation(){
137         vector<int>::iterator it; // itérateur
138         for (it = heure_validation.begin(); it != heure_validation.end() ; it++)
139         {
140             cout<<"validation à "<<(*it)<<endl;
141         }
142     }
143
144     // Affiche le nombre de fois où la chronique a été terminée
145     void chronique::afficher_nb_validation(){
146         cout<<"nombre de validation = "<<nb_validation<<endl;
147     }
148
149     // Valide la chronique (affiche un message dans le terminal pour annoncer la validation de la chronique,
150     // incrémente nb_validation, met à jour heure_validation et passe nb_contraintes_restantes à
151     nb_contraintes_total)
152     void chronique::validation_chronique(){
153         cout<<"la chronique "<<nom<<" a ete validee"<<endl;
154         set_valid(nb_validation+1);
155         set_h_valid(HEURE_COURANTE);
156         set_event(true);
157         reset_nb_contraintes();
158     }

```