

```

1  #ifndef CHRONIQUE_H_INCLUDED
2  #define CHRONIQUE_H_INCLUDED
3
4  #include <iostream>
5  #include <fstream>
6  #include <vector>
7  #include <string>
8  #include "contrainte.h"
9  #include "event.h"
10 using namespace std;
11
12 /* VARIABLES GLOBALES APPELEES */
13
14 extern int HEURE_COURANTE; // Variable du temps, mise à jour tout au long de la lecture de la séquence
15
16 /* CLASSES */
17
18 // Chronique unitaire
19 class chronique {
20
21 private:
22
23     /* ATTRIBUTS */
24
25     string nom; //initialisé par le fichier de définition des chroniques
26     bool event_verify; // initialisé à false, utile pour les chroniques de chroniques
27     int nb_validation; // initialisé à 0, indique le nombre de validation de la chronique
28     std::vector<int> heure_validation; // tableau des heures de validation
29     int nb_contraintes_restantes; // initialisé à nb_contraintes_total
30     int nb_contraintes_total; // initialisé par le fichier de définition des chroniques
31     std::vector<contrainte> contraintes; // tableau de contraintes, initialisé par le fichier
32                                     // de définition des chroniques
33
34
35 public:
36
37     /* CONSTRUCTEURS */
38
39     // Constructeur simple
40     chronique();
41
42     // Constructeur évolué, créé la chronique à partir des données
43     chronique(string label, int nb_contraintes, vector<contrainte> vect_contraintes);
44
45     /* ASSESSEURS */
46
47     // Assesseur, ressort le nom de la chronique
48     string get_nom();
49
50     // Assesseur, ressort l'état de la chronique (pour les chroniques de chroniques)
51     bool get_event();
52
53     // Assesseur, ressort le nombre de validation
54     int get_nb_valid();
55
56     // Assesseur, ressort le nombre de contraintes restantes
57     int get_contraintes_restantes();
58
59
60     // Assesseur, ressort le nombre de contraintes totales
61     int get_contraintes_total();
62
63
64     // Assesseur, ressort le tableau des heures de validation
65     std::vector<int> get_h_valid();
66

```

```

67 // Assesseur, ressort l'heure de la dernière validation
68 int get_last_h_valid();
69
70 // Assesseur, ressort le tableau des contraintes
71 std::vector<contrainte> get_contraintes();
72
73 // Mutateur, change la valeur du nombre de contraintes restantes de la chronique
74 void set_contraintes_restantes(int nb);
75
76 // Mutateur, change le label de l'élément
77 void set_nom(string name);
78
79 // Mutateur, change la valeur d'évènement de la chronique
80 void set_event(bool event);
81
82 // Mutateur, change la valeur du nombre de validation de la chronique
83 void set_valid(int nb);
84
85 // Mutateur, ajoute une valeur dans les heures de validation de la chronique
86 void set_h_valid(int heure);
87
88 // Reinitialisateur, remet à 0 la valeur du nombre de validation
89 void reset_valid();
90
91 // Reinitialisateur, remet à 0 le tableau des heures de validation
92 void reset_h_valid();
93
94 // Reinitialisateur, remet au maximum la valeur des contraintes restantes à valider
95 void reset_nb_contraintes();
96
97 // Reinitialisateur, réinitialise tous les attributs
98 void reinit_chronique_all();
99
100 /* METHODES */
101
102 // Affiche certains éléments dans la console
103 void afficher();
104
105 // Affiche les heures de validation de la chronique dans le terminal
106 void afficher_heures_validation();
107
108 // Affiche le nombre de fois où la chronique a été terminée
109 void afficher_nb_validation();
110
111 // Valide la chronique (affiche un message dans le terminal pour annoncer la
112 // validation de la chronique, incrémente nb_validation, met à jour heure_validation et
113 // passe nb_contraintes_restantes à nb_contraintes_total)
114 void validation_chronique();
115
116 };
117
118
119 #endif

```