

```

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 de définition
des chroniques
32
33
34 public:
35
36     /* CONSTRUCTEURS */
37
38     // Constructeur simple
39     chronique();
40
41     // Constructeur évolué, créé la chronique à partir des données
42     chronique(string label, int nb_contraintes, vector<contrainte> vect_contraintes);
43
44     /* ASSESSEURS */
45
46     // Assesseur, ressort le nom de la chronique
47     string get_nom();
48
49     // Assesseur, ressort l'état de la chronique (pour les chroniques de chroniques)
50     bool get_event();
51
52     // Assesseur, ressort le nombre de validation
53     int get_nb_valid();
54
55     // Assesseur, ressort le nombre de contraintes restantes
56     int get_contraintes_restantes();
57
58
59     // Assesseur, ressort le nombre de contraintes totales
60     int get_contraintes_total();
61
62
63     // Assesseur, ressort le tableau des heures de validation
64     std::vector<int> get_h_valid();
65

```

```

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

```