

```

1  #include "contrainte.h"
2
3  /* CONSTRUCTEURS */
4
5  // Constructeur simple
6  contrainte::contrainte(){
7      temps=0;
8      type=0;
9  }
10
11 // Constructeur évolu  , cr    la contrainte   partir des donn  es
12 contrainte::contrainte(int time, int type_contrainte, string nom_evt){
13     temps=time;
14     type=type_contrainte;
15
16     // recherche de l'event par nom
17     list<event>::iterator it;
18     for(it=MES_EVENTS.begin(); it!=MES_EVENTS.end(); it++) {
19         if ((*it).get_label() == nom_evt) {
20             evenement=(*it);
21         }
22     }
23 }
24
25 /* OPERATEURS */
26
27 contrainte contrainte::operator =(const contrainte &c){
28     evenement=c.evenement;
29     type=c.type;
30     return *this;
31 }
32
33 /* ASSESSEURS */
34
35 // Assesseur, ressort le type de la contrainte
36 int contrainte::get_type(){
37     return type;
38 }
39
40 // Assesseur, ressort l'event associ   la contrainte
41 event contrainte::get_event(){
42     return evenement;
43 }
44
45 // Assesseur, ressort le temps associ   la contrainte
46 int contrainte::get_time(){
47     return temps;
48 }
49
50 // Mutateur, change le nom de la contrainte
51 void contrainte::set_type(int type_contrainte){
52     type = type_contrainte;
53 }
54
55 // Mutateur, change l'event associ     la contrainte
56 void contrainte::set_type(event evt){
57     evenement = evt;
58 }
59
60 /* METHODES */
61
62 // Affiche les datas de la contrainte dans le terminal
63 void contrainte::afficher_contrainte(){
64     cout<<"contrainte de type : "<<type<<endl;
65     cout<<"evenement associe : ";
66     evenement.afficher();

```

```
67     cout<<endl;  
68 }
```