

```

1  /*****
2      TabTrajet.h - Interface de TabTrajet.cpp
3      Un tableau Dynamique de Trajet*
4      -----
5      début          : 15/11/2017
6      auteurs        : B3405 - Etienne et Grégoire
7  *****/
8
9  //----- Interface du module <TabTrajet> (fichier TabTrajet.h) -----
10 #if ! defined ( TabTrajet_H )
11 #define TabTrajet_H
12
13 //-----
14 // Rôle du module <TabTrajet>
15 // Stocke un tableau dynamique de Trajet *
16 //     Se réalloue automatiquement, gère l'ajout de TrajetSimple
17 //     Dispose de Geters pour sa taille et ses trajets par indice
18 //-----
19
20 //////////////////////////////////////// INCLUDE
21 #include "Trajet.h"
22
23 class TabTrajet {
24 public:
25 //----- Fonctions publiques
26     void Add ( const Trajet * t );
27     // Contrat :
28     //     Ajoute à la fin du tableau le pointeur de trajet passé en paramètre
29     // Mode d'emploi:
30     //     t un Trajet alloué, non deleted avant la fin de vie du TabTrajet
31
32     const Trajet * Get ( unsigned int i ) const;
33     // Contrat :
34     //     Renvoie le pointeur de trajet positionné à la i-ème case du tableau,
35     //     nullptr si i >= utilise
36     // Mode d'emploi:
37     //     i est un int positif inférieur à la taille de TabTrajet
38
39     unsigned int GetUtilise() const;
40     // Contrat :
41     //     Renvoie le nombre de trajet contenu de le tableau
42
43 //----- Constructeur et Destructeur
44     TabTrajet();
45     virtual ~TabTrajet();
46     // Contrat :
47     //     Fais appelle au destructeur adéquat sur chaque
48     //     Trajet qui le compose, puis détruit le tableau
49
50 protected:
51     unsigned int max;

```

```

52     unsigned int utilise;
53     const Trajet** listTrajet;
54     void realloc ();
55     // Contrat :
56     //         Double la taille du tableau listTrajet contenant les Trajet *
57 };
58 #endif // TabTrajet_H
59
60 /*****
61                                     TabTrajet.cpp
62                                     -----
63     début                          : 15/11/2017
64     auteurs                        : B3405 - Etienne et Grégoire
65 *****/
66 //----- Réalisation de la classe <TabTrajet> (fichier TabTrajet.cpp)
67
68 //----- INCLUDE
69 //----- Include système
70 using namespace std;
71 #include <iostream>
72 //----- Include personnel
73 #include "TabTrajet.h"
74
75 //----- PUBLIC
76 //----- Méthodes publiques
77 void TabTrajet::Add(const Trajet * t)
78 {
79     if (this->utilise >= this->max)
80     {
81         this->realloc();
82     }
83     for (unsigned int i = 0; i < utilise; i++)
84     {
85         if (listTrajet[i] == t)
86         {
87             cout << "Trajet déjà présent dans le catalogue..." << endl;
88             return;
89         }
90     }
91     this->listTrajet[this->utilise] = t;
92     this->utilise++;
93     return;
94 }
95
96 const Trajet * TabTrajet::Get(unsigned int i) const
97 {
98     if (i >= this->utilise)
99     {
100         return nullptr;
101     }
102     return this->listTrajet[i];
103 }
104

```

```
105 unsigned int TabTrajet::GetUtilise() const
106 {
107     return this->utilise;
108 }
109
110 //----- Constructeurs - destructeurs
111 TabTrajet::TabTrajet()
112 {
113     #ifdef MAP
114         cout << "Appel au constructeur de <TabTrajet>" << endl;
115     #endif
116     max = 5;
117     utilise = 0;
118     listTrajet = new const Trajet*[max];
119     for (unsigned int i = 0; i < max; i++)
120     {
121         listTrajet[i] = nullptr;
122     }
123 } //----- Fin de TabTrajet
124
125 TabTrajet::~TabTrajet()
126 {
127     #ifdef MAP
128         cout << "Appel au destructeur de <TabTrajet>" << endl;
129     #endif
130     for (unsigned int i = 0; i < max; i++)
131     {
132         if (listTrajet[i] != nullptr)
133         {
134             delete listTrajet[i];
135         }
136     }
137     delete[] listTrajet;
138 } //----- Fin de ~TabTrajet
139
140 //----- PRIVE
141 //----- Méthode protégée
142 void TabTrajet::realloc()
143 {
144     const Trajet** newList = new const Trajet*[max * 2];
145     for (unsigned int i = 0; i < utilise; i++)
146     {
147         newList[i] = listTrajet[i];
148     }
149     for (unsigned int i = utilise; i < 2 * max; i++)
150     {
151         newList[i] = nullptr;
152     }
153     const Trajet ** old = listTrajet;
154     listTrajet = newList;
155     delete[] old;
156     max *= 2;
157 }
```