

Compte Rendu TP2 CPOO

Maud LERAY, Pierre-Marie AIRIAU

INSA de Rennes
4INFO, groupe 1.2

17 septembre 2014

Listing 1 – Fichier source de la classe Chaîne : Chaîne.cpp

```
1 #include "Chaîne.h"
2 #include <iostream>
3 #include <cstdlib>
4 #include <iomanip>
5
6 Chaîne::Chaîne(){
7     char* str = new char[1];
8     str[0] = '\0';
9     this->tab = str;
10    std::cout << "Creation d'une chaîne vide \n";
11 }
12
13 Chaîne::Chaîne(const char * c){
14     int i = 0;
15     int j = 0;
16     while (c[j] != '\0') {
17         j++;
18     }
19     char* res = new char[j+1];
20     while (c[i] != '\0') {
21         res[i] = c[i];
22         i++;
23     };
24     res[j] = '\0';
25     this->tab = res;
26     std::cout << "Creation (par pointeur) d'une chaîne de longueur " << j << "\n";
27 }
28
29 Chaîne::Chaîne(const Chaîne & c){
30     int i = 0;
```

```

31 int j = 0;
32 while (c.tab[j] != '\0') {
33     j++;
34 }
35 char* res = new char[j + 1];
36 while (c.tab[i] != '\0') {
37     res[i] = c.tab[i];
38     i++;
39 };
40 res[j] = '\0';
41 this->tab = res;
42 std::cout << "Creation (par reference) d'une chaine de longueur " << j << "\n";
43 }
44
45 Chaine::~Chaine() {
46     delete [] this->tab;
47     std::cout << "Suppression d'une chaine \n";
48 }
49
50 void Chaine::affiche(){
51
52     if (this->tab[0] == '\0') { std::cout << "Chaine vide\n"; }
53     else {
54         std::cout << this->tab;
55         std::cout << "\n";
56     }
57 }
58
59 int Chaine::longueur(Chaine *ch){
60
61     int i = 0;
62     while (ch->tab[i] != '\0') {
63         i++;
64     }
65     return i;
66 }
67
68 bool Chaine::comparer(Chaine* ch1, Chaine* ch2){
69
70     int l1 = longueur(ch1);
71     int l2 = longueur(ch2);
72
73     if (l1 != l2) { std::cout << "Les deux chaines ne sont pas egales\n"; return
false; }
74     else {
75         int i;
76         for (i = 0; i < l1 + 1; i++){
77             if (ch1->tab[i] != ch2->tab[i]) {
78                 std::cout << "Les deux chaines ne sont pas egales\n";

```

```

79         return false;
80     }
81 }
82     std::cout << "Les deux chaines sont egales\n";
83     return true;
84 }
85 }
86
87 Chaine *Chaine::concat(Chaine *ch1, Chaine *ch2) {
88
89     int l1 = longueur(ch1);
90     int l2 = longueur(ch2);
91
92     int l = l1 + l2;
93     char * res = new char[l];
94
95     int i, j;
96     for (i = 0; i < l1; i++){
97         res[i] = ch1->tab[i];
98     }
99     for (j = 0; j < l2; j++){
100         res[i+j] = ch2->tab[j];
101     }
102     res[l] = '\0';
103     Chaine* r = new Chaine(res);
104     std::cout << "Concatenation reussie\n";
105     return r;
106 }
107
108 bool Chaine::caractere(int pos){
109     if (pos > longueur(this) - 1 || pos < 0) {
110         std::cout << "La position demandee n'existe pas\n";
111         return false;
112     } else {
113         std::cout << "Le caractere demande est : " << this->tab[pos] << "\n";
114         return true;
115     }
116 }
117
118 bool Chaine::sous_chaine(char deb, char fin){
119
120     int i, j;
121     int ind1 = -1;
122     int ind2 = -1;
123
124     for (i = 0; i < longueur(this); i++) {
125         if (this->tab[i] == deb) {
126             ind1 = i;
127

```

```

128 }
129 if (ind1 == -1) {
130     std::cout << "Premier caractere introuvable\n";
131     return false;
132 }
133 else {
134     for (j = ind1+1; j < longueur(this); j++) {
135         if (this->tab[j] == fin) {
136             ind2 = j;
137         }
138     }
139     if (ind2 == -1) {
140         std::cout << "Dernier caractere introuvable\n";
141         return false;
142     }
143 }
144 this->sous_chaine(ind1, ind2);
145 }
146 bool Chaine::sous_chaine(int ind1, int ind2) {
147
148     int i;
149     char* res = new char[ind2 - ind1 + 1];
150
151     if (ind1 > longueur(this) - 1 || ind1 < 0 || ind2 > longueur(this) || ind2 < 0)
152     {
153         std::cout << "La position demandee n'existe pas\n";
154         return false;
155     }
156     else {
157         for (i = ind1; i < ind2 + 1; i++) {
158             res[i - ind1] = this->tab[i];
159         }
160         res[i - ind1] = '\0';
161     }
162     std::cout << "La sous-chaine demandee est : " << res << "\n";
163     return true;
164 }

```

Listing 2 – Fichier header de la classe Chaine : Chaine.h

```

1 class Chaine {
2
3 private:
4     char* tab;
5
6 public:
7     Chaine();
8     Chaine(const char * c);
9     Chaine(const Chaine & c);
10    ~Chaine();

```

```

11
12 void affiche();
13 static int longueur(Chaine *ch);
14 static bool comparer(Chaine *ch1, Chaine *ch2);
15 static Chaine* concat(Chaine *ch1, Chaine *ch2);
16 bool caractere(int pos);
17 bool sous_chaine(char deb, char fin);
18 bool sous_chaine(int ind1, int ind2);
19 };

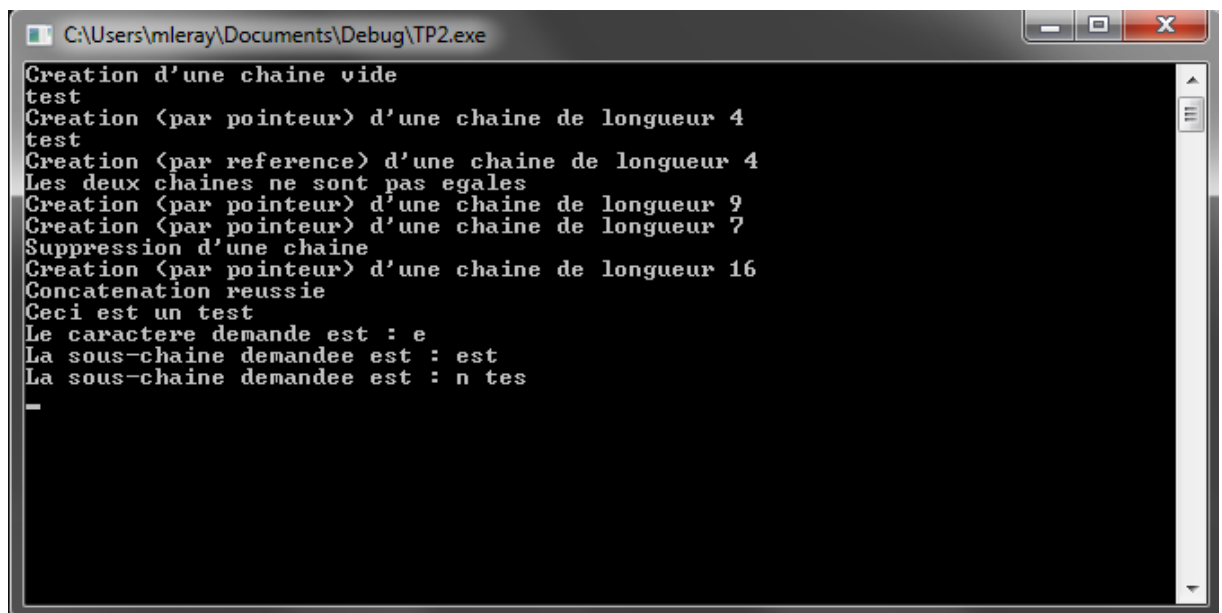
```

Listing 3 – Programme principal : main.cpp

```

1 #include "Chaine.h"
2 #include <iostream>
3 #include <cstdlib>
4 #include <iomanip>
5
6 int main(){
7
8     Chaine* c1 = new Chaine();
9     char input[81];
10    std::cin >> input;
11    Chaine ptr = Chaine(input);
12    ptr.affiche();
13    Chaine* c2 = new Chaine(ptr);
14    Chaine::comparer(c1, c2);
15    Chaine* c3 = new Chaine("Ceci est ");
16    Chaine* c4 = new Chaine("un test");
17    delete c2;
18    Chaine* c5 = Chaine::concat(c3, c4);
19    c5->affiche();
20    c3->caractere(5);
21    c3->sous_chaine(5, 7);
22    c4->sous_chaine('n', 's');
23    return 0;
24 }

```



```
C:\Users\mleray\Documents\Debug\TP2.exe
Creation d'une chaine vide
test
Creation (par pointeur) d'une chaine de longueur 4
test
Creation (par reference) d'une chaine de longueur 4
Les deux chaines ne sont pas egales
Creation (par pointeur) d'une chaine de longueur 9
Creation (par pointeur) d'une chaine de longueur 7
Suppression d'une chaine
Creation (par pointeur) d'une chaine de longueur 16
Concatenation reussie
Ceci est un test
Le caractere demande est : e
La sous-chaine demandee est : est
La sous-chaine demandee est : n tes
-
```