

Initiation à l'algorithmique et la programmation : la machine caractère

Généré par Doxygen 1.8.12

Table des matières

1	Documentation de la machine "caractères"	2
1.1	Introduction	2
1.2	Installation	2
1.3	Utilisation de la machine caractères	2
1.3.1	Description de la machine caractères	2
1.3.2	Action sur la machine caractères	2
1.3.3	État de la machine caractères	2
1.4	Exercices utilisant la machine caractères	3
2	Index des classes	3
2.1	Liste des classes	3
3	Index des fichiers	3
3.1	Liste des fichiers	3
4	Documentation des classes	3
4.1	Référence de la classe CMachineCaractere	3
4.1.1	Description détaillée	4
4.1.2	Documentation des constructeurs et destructeur	4
4.1.3	Documentation des fonctions membres	4
5	Documentation des fichiers	6
5.1	Référence du fichier machineCaractere.cc	6
5.2	machineCaractere.cc	6
5.3	Référence du fichier machineCaractere.h	6
5.4	machineCaractere.h	7
5.5	Référence du fichier main.cc	8
5.5.1	Documentation des fonctions	8
5.6	main.cc	9
	Index	11

1 Documentation de la machine "caractères"

1.1 Introduction

La machine caractères a été développée dans le cadre de l'apprentissage du C++ au club informatique de Lorient. Elle est inspirée du cours d'algorithmique de Pierre-Claude SCHOLL (université de Grenoble).

1.2 Installation

Pour utiliser la machine caractères, il suffit de joindre à votre projet C++ les fichiers `machineCaractere.h` et `machineCaractere.cc`

Vous disposez ainsi de la classe C++ `CMachineCaractere` permettant d'utiliser une machine caractères.

1.3 Utilisation de la machine caractères

1.3.1 Description de la machine caractères

Une machine caractères comporte :

- un ruban.
 - Le ruban est une suite de caractères terminée par une marque.
 - Un ruban est initialisé par le constructeur `CMachineCaractere : :CMachineCaractere (const string& kpr←Ruban)`.
- Une fenêtre montrant le caractère courant.
 - Après initialisation, la fenêtre est positionnée sur le premier caractère du ruban.

1.3.2 Action sur la machine caractères

L'action possible sur la machine caractères :

- Vous pouvez déplacer la fenêtre de la machine caractères vers le caractère suivant via la méthode void `CMachineCaractere : :avancer ()`.
Vous ne devez pas avoir atteint la fin du ruban.

1.3.3 État de la machine caractères

Vous pouvez consulter l'état de la machine caractères :

- Vous pouvez consulter le caractère courant du ruban de la machine caractères via la méthode char `C←MachineCaractere : :caractereCourant ()`.
Vous ne devez pas avoir atteint la fin du ruban.
- Vous pouvez demander si vous avez atteint la marque de fin du ruban de la machine caractères via la méthode bool `CMachineCaractere : :marque ()`

1.4 Exercices utilisant la machine caractères

- Compter les 'e' du texte.
Tester avec "Elle n'est pas veuve"
Tester avec "Quelle belle epee!"
Tester avec ""
- Compter les mots du texte. Un mot est une suite de caractères majuscules, minuscules non accentuées ou chiffres séparés par tout autre caractère.
Tester avec "Elle n'est pas veuve"
Tester avec "Il a bu 25 litres de bière"
Tester avec ""
- Compter le nombre de 'le' du texte.
Tester avec "le garçon la trouve très belle"
Tester avec ""

2 Index des classes

2.1 Liste des classes

Liste des classes, structures, unions et interfaces avec une brève description :

[CMachineCaractere](#) 3

3 Index des fichiers

3.1 Liste des fichiers

Liste de tous les fichiers avec une brève description :

[machineCaractere.cc](#) 6

[machineCaractere.h](#) 6

[main.cc](#) 8

4 Documentation des classes

4.1 Référence de la classe CMachineCaractere

```
#include <machineCaractere.h>
```

Fonctions membres publiques

- [CMachineCaractere](#) (const string &kprRuban)
- virtual [~CMachineCaractere](#) ()
- bool [marque](#) ()
- void [avancer](#) ()
- char [caractereCourant](#) ()

4.1.1 Description détaillée

Définition à la ligne 60 du fichier [machineCaractere.h](#).

4.1.2 Documentation des constructeurs et destructeur

4.1.2.1 CMachineCaractere()

```
CMachineCaractere::CMachineCaractere (
    const string & kprRuban )
```

Constructeur [CMachineCaractere : :CMachineCaractere](#) (const string& kprRuban) Initialise une machine caractères avec la chaîne kprRuban.

Paramètres

<i>kprRuban</i>	: Chaîne permettant l'initialisation du ruban
-----------------	-----------------------------------------------

Définition à la ligne 4 du fichier [machineCaractere.cc](#).

4.1.2.2 ~CMachineCaractere()

```
CMachineCaractere::~~CMachineCaractere ( ) [virtual]
```

Destructeur [CMachineCaractere : :~CMachineCaractere](#) ().
Termine la machine caractères

Définition à la ligne 9 du fichier [machineCaractere.cc](#).

4.1.3 Documentation des fonctions membres

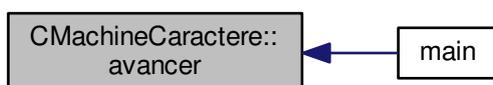
4.1.3.1 avancer()

```
void CMachineCaractere::avancer ( )
```

Méthode bool [CMachineCaractere : :avancer](#) ().
Avance le ruban sur le caractère suivant.
La fin du ruban ne doit pas avoir été atteinte.

Définition à la ligne 18 du fichier [machineCaractere.cc](#).

Voici le graphe des appelants de cette fonction :



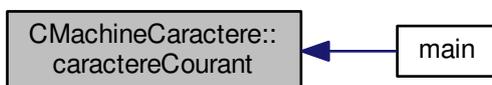
4.1.3.2 caractereCourant()

```
char CMachineCaractere::caractereCourant ( )
```

Méthode char [CMachineCaractere::caractereCourant\(\)](#).
renvoi le caractère courant sous la fenêtre du ruban.
La fin du ruban ne doit pas avoir été atteinte

Définition à la ligne [24](#) du fichier [machineCaractere.cc](#).

Voici le graphe des appelants de cette fonction :



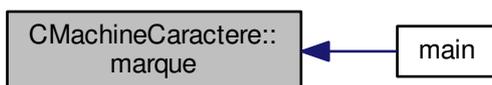
4.1.3.3 marque()

```
bool CMachineCaractere::marque ( )
```

Méthode bool [CMachineCaractere::marque\(\)](#).
renvoi Vrai (true) si la fin du ruban est atteinte

Définition à la ligne [12](#) du fichier [machineCaractere.cc](#).

Voici le graphe des appelants de cette fonction :



La documentation de cette classe a été générée à partir des fichiers suivants :

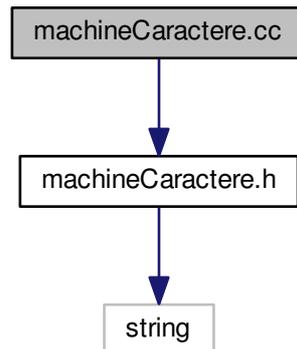
- [machineCaractere.h](#)
- [machineCaractere.cc](#)

5 Documentation des fichiers

5.1 Référence du fichier machineCaractere.cc

```
#include "machineCaractere.h"
```

Graphe des dépendances par inclusion de machineCaractere.cc :



5.2 machineCaractere.cc

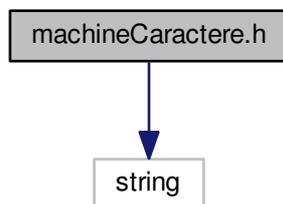
```

00001 #include "machineCaractere.h"
00002
00003 // =====
00004 CMachineCaractere::CMachineCaractere (const string& kprRuban) :
00005     kmvRuban (kprRuban),
00006     mvPosition (0) {}
00007
00008 // =====
00009 CMachineCaractere::~CMachineCaractere () {}
00010
00011 // =====
00012 bool CMachineCaractere::marque () {
00013
00014     return (kmvRuban.length () == mvPosition);
00015 };
00016
00017 // =====
00018 void CMachineCaractere::avancer () {
00019
00020     if (kmvRuban.length () > mvPosition) mvPosition++;
00021 };
00022
00023 // =====
00024 char CMachineCaractere::caractereCourant () {
00025
00026     return kmvRuban[mvPosition];
00027 };
  
```

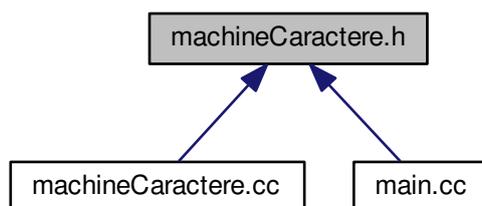
5.3 Référence du fichier machineCaractere.h

```
#include <string>
```

Graphe des dépendances par inclusion de machineCaractere.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



Classes

— class [CMachineCaractere](#)

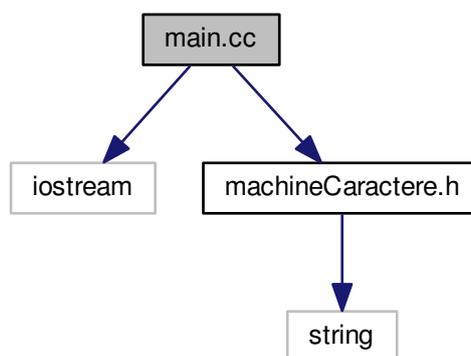
5.4 machineCaractere.h

```
00001 #ifndef MACHINECARACTERE_HH
00002 #define MACHINECARACTERE_HH
00003
00004 using namespace std;
00005
00006 #include <string>
00007
00008 /* Documentation tag for Doxygen
00009 */
00010
00059 // =====
00060 class CMachineCaractere {
00061
00062     public:
00066         CMachineCaractere (const string& kprRuban);
00069         virtual ~CMachineCaractere ();
00070
00071         // Méthodes publiques
00074         bool marque ();
00078         void avancer ();
00082         char caractereCourant ();
```

```
00083
00084     protected:
00085
00086     private:
00087         // Données privées
00088         const string kmvRuban;
00089         int mvPosition;
00090
00091         // Méthodes privées
00092     };
00093
00094 #endif // MACHINECARACTERE_HH
```

5.5 Référence du fichier main.cc

```
#include <iostream>
#include "machineCaractere.h"
Graphe des dépendances par inclusion de main.cc :
```



Fonctions

— int [main](#) ()

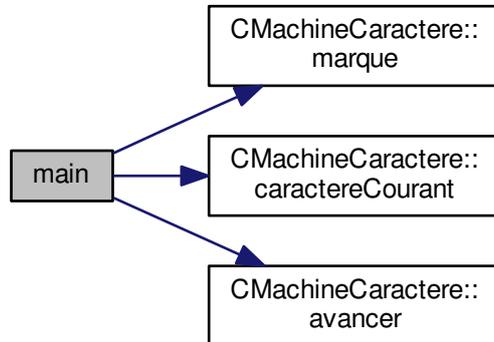
5.5.1 Documentation des fonctions

5.5.1.1 main()

```
int main ( )
```

Définition à la ligne [8](#) du fichier [main.cc](#).

Voici le graphe d'appel pour cette fonction :



5.6 main.cc

```
00001 #include <iostream>
00002
00003 using namespace std;
00004
00005 #include "machineCaractere.h"
00006
00007 // =====
00008 int main () {
00009
00010     int lvCompteur;
00011     CMachineCaractere lvMC ("Ceci est un texte d'essai");
00012
00013     cout << "Compter les 'e'" << endl;
00014     lvCompteur = 0;
00015     while (!lvMC.marque ()) {
00016         if (lvMC.caractereCourant () == 'e') lvCompteur++;
00017         lvMC.avancer ();
00018     }
00019     cout << "Il y a " << lvCompteur << " 'e'" << endl;
00020     return 0;
00021 }
```


Index

- ~CMachineCaractere
 - CMachineCaractere, [4](#)
- avancer
 - CMachineCaractere, [4](#)
- CMachineCaractere, [3](#)
 - ~CMachineCaractere, [4](#)
 - avancer, [4](#)
 - CMachineCaractere, [4](#)
 - caractereCourant, [4](#)
 - marque, [5](#)
- caractereCourant
 - CMachineCaractere, [4](#)
- machineCaractere.cc, [6](#)
- machineCaractere.h, [6](#)
- main
 - main.cc, [8](#)
- main.cc, [8](#)
 - main, [8](#)
- marque
 - CMachineCaractere, [5](#)