

Calculatrice C++

3

Généré par Doxygen 1.8.6

Mardi 17 Février 2015 13 :32 :52

Table des matières

1	Documentation Calculatrice Cpp	1
1.1	Introduction	1
2	Index des fichiers	3
2.1	Liste des fichiers	3
3	Documentation des fichiers	5
3.1	Référence du fichier addition.cpp	5
3.1.1	Documentation des fonctions	5
3.1.1.1	ajouter	5
3.1.1.2	ajouter	5
3.1.1.3	ajouter	6
3.1.1.4	ajouter	6
3.2	Référence du fichier division.cpp	7
3.2.1	Documentation des fonctions	7
3.2.1.1	division	7
3.2.1.2	division	7
3.2.1.3	division	7
3.2.1.4	division	8
3.3	Référence du fichier main.cpp	8
3.3.1	Documentation des fonctions	8
3.3.1.1	main	8
3.4	Référence du fichier modulo.cpp	9
3.4.1	Documentation des fonctions	9
3.4.1.1	modulo	9
3.4.1.2	modulo	9
3.5	Référence du fichier multiplication.cpp	10
3.5.1	Documentation des fonctions	10
3.5.1.1	multiplication	10
3.5.1.2	multiplication	10
3.5.1.3	multiplication	11
3.5.1.4	multiplication	11

3.6	Référence du fichier save/division_conflict-20150202-160133.cpp	12
3.6.1	Documentation des fonctions	12
3.6.1.1	division	12
3.6.1.2	division	12
3.6.1.3	division	12
3.6.1.4	division	13
3.7	Référence du fichier save/mainInteractionOperateur.cpp	13
3.7.1	Documentation des fonctions	13
3.7.1.1	main	13
3.8	Référence du fichier soustraction.cpp	14
3.8.1	Documentation des fonctions	15
3.8.1.1	soustraction	15
3.8.1.2	soustraction	15
3.8.1.3	soustraction	15
3.8.1.4	soustraction	16

Chapitre 1

Documentation Calculatrice Cpp

1.1 Introduction

Documentation d'un programme écrit en C++ qui doit permettre de réaliser les calculs de type +, -, x, ÷, modulo. Utilisant les types de variables int32, int64, float et double.

Chapitre 2

Index des fichiers

2.1 Liste des fichiers

Liste de tous les fichiers avec une brève description :

/home/bebo/Documents/BTS/SI6/TP/tp_calculatrice_Cpp/ main.cpp	
Calculatrice affichant le résultat des valeurs définies au préalable dans le programme	8
/home/bebo/Documents/BTS/SI6/TP/tp_calculatrice_Cpp/fichiersDecoupees/ addition.cpp	5
/home/bebo/Documents/BTS/SI6/TP/tp_calculatrice_Cpp/fichiersDecoupees/ division.cpp	7
/home/bebo/Documents/BTS/SI6/TP/tp_calculatrice_Cpp/fichiersDecoupees/ modulo.cpp	9
/home/bebo/Documents/BTS/SI6/TP/tp_calculatrice_Cpp/fichiersDecoupees/ multiplication.cpp	10
/home/bebo/Documents/BTS/SI6/TP/tp_calculatrice_Cpp/fichiersDecoupees/ soustraction.cpp	14

Chapitre 3

Documentation des fichiers

3.1 Référence du fichier /home/bebo/Documents/BTS/Sl6/TP/tp_calculatrice_Cpp/fichiers-Decoupes/addition.cpp

Fonctions

- `int32_t ajouter (int32_t operandeA, int32_t operandeB)`
Fonction d'addition de deux nombre de type int32_t.
- `int64_t ajouter (int64_t operandeA, int64_t operandeB)`
Fonction d'addition de deux nombre de type int64_t.
- `float ajouter (float operandeA, float operandeB)`
Fonction d'addition de deux nombre de type Float.
- `double ajouter (double operandeA, double operandeB)`
Fonction d'addition de deux nombre de type Double.

3.1.1 Documentation des fonctions

3.1.1.1 `int32_t ajouter (int32_t operandeA, int32_t operandeB)`

Fonction d'addition de deux nombre de type `int32_t`.

Paramètres

<code>operandeA</code>	est un entier sur 32 bits
<code>operandeB</code>	est un entier sur 32 bits

Renvoie

La somme des deux entiers 32 bits saisie par l'utilisateur.

Voir également

`ajouter(int64_t, int64_t), ajouter(float, float), ajouter(double, double)`

```
11 {  
12     return operandeA + operandeB;  
13 }
```

3.1.1.2 `int64_t ajouter (int64_t operandeA, int64_t operandeB)`

Fonction d'addition de deux nombre de type `int64_t`.

Paramètres

<i>operandeA</i>	est un entier sur 64 bits
<i>operandeB</i>	est un entier sur 64 bits

Renvoie

La somme des deux entiers 64 bits saisie par l'utilisateur.

Voir également

[ajouter\(float, float\)](#), [ajouter\(double, double\)](#), [ajouter\(int32_t, int32_t\)](#)

```
24 {  
25     return operandeA + operandeB;  
26 }
```

3.1.1.3 float ajouter (float *operandeA*, float *operandeB*)

Fonction d'addition de deux nombre de type Float.

Paramètres

<i>operandeA</i>	est un Float
<i>operandeB</i>	est un Float

Renvoie

La somme des deux Float saisie par l'utilisateur.

Voir également

[ajouter\(double, double\)](#), [ajouter\(int32_t, int32_t\)](#), [ajouter\(int64_t, int64_t\)](#)

```
37 {  
38     return operandeA + operandeB;  
39 }
```

3.1.1.4 double ajouter (double *operandeA*, double *operandeB*)

Fonction d'addition de deux nombre de type Double.

Paramètres

<i>operandeA</i>	est un Double
<i>operandeB</i>	est un Double

Renvoie

La somme des deux Double saisie par l'utilisateur.

Voir également

[ajouter\(int32_t, int32_t\)](#), [ajouter\(int64_t, int64_t\)](#), [ajouter\(float, float\)](#)

```
50 {  
51     return operandeA + operandeB;  
52 }
```

3.2 Référence du fichier /home/bebo/Documents/BTS/SI6/TP/tp_calculatrice_Cpp/fichiers-Decoupes/division.cpp

Fonctions

- `int32_t division (int32_t operandeA, int32_t operandeB)`
Fonction de division de deux nombre de type int32_t.
- `int64_t division (int64_t operandeA, int64_t operandeB)`
Fonction de division de deux nombre de type int64_t.
- `float division (float operandeA, float operandeB)`
Fonction de division de deux nombre de type Float.
- `double division (double operandeA, double operandeB)`
Fonction de division de deux nombre de type Double.

3.2.1 Documentation des fonctions

3.2.1.1 `int32_t division (int32_t operandeA, int32_t operandeB)`

Fonction de division de deux nombre de type `int32_t`.

Paramètres

<code>operandeA</code>	est un entier sur 32 bits
<code>operandeB</code>	est un entier sur 32 bits

Renvoie

La division des deux entiers 32 bits saisie par l'utilisateur.

Voir également

`division(int64_t, int64_t)`, `division(float, float)`, `division(double, double)`

```
11 {
12     return operandeA / operandeB;
13 }
```

3.2.1.2 `int64_t division (int64_t operandeA, int64_t operandeB)`

Fonction de division de deux nombre de type `int64_t`.

Paramètres

<code>operandeA</code>	est un entier sur 64 bits
<code>operandeB</code>	est un entier sur 64 bits

Renvoie

La division des deux entiers 64 bits saisie par l'utilisateur.

Voir également

`division(float, float)`, `division(double, double)`, `division(int32_t, int32_t)`

```
24 {
25     return operandeA / operandeB;
26 }
```

3.2.1.3 `float division (float operandeA, float operandeB)`

Fonction de division de deux nombre de type `Float`.

Paramètres

<i>operandeA</i>	est un Float
<i>operandeB</i>	est un Float

Renvoie

La division des deux Float saisie par l'utilisateur.

Voir également

[division\(double, double\)](#), [division\(int32_t, int32_t\)](#), [division\(int64_t, int64_t\)](#)

```
37 {
38     return operandeA / operandeB;
39 }
```

3.2.1.4 double division (double *operandeA*, double *operandeB*)

Fonction de division de deux nombre de type Double.

Paramètres

<i>operandeA</i>	est un Double
<i>operandeB</i>	est un Double

Renvoie

La division des deux Double saisie par l'utilisateur.

Voir également

[division\(int32_t, int32_t\)](#), [division\(int64_t, int64_t\)](#), [division\(float, float\)](#)

```
50 {
51     return operandeA * operandeB;
52 }
```

3.3 Référence du fichier /home/bebo/Documents/BTS/SI6/TP/tp_calculatrice_Cpp/fichiers-Decoupes/modulo.cpp**Fonctions**

- `int32_t modulo (int32_t operandeA, int32_t operandeB)`
Fonction de modulo de deux nombre de type int32_t.
- `int64_t modulo (int64_t operandeA, int64_t operandeB)`
Fonction de modulo de deux nombre de type int64_t.

3.3.1 Documentation des fonctions**3.3.1.1 int32_t modulo (int32_t *operandeA*, int32_t *operandeB*)**

Fonction de modulo de deux nombre de type int32_t.

Paramètres

<i>operandeA</i>	est un entier sur 32 bits
<i>operandeB</i>	est un entier sur 32 bits

Renvoie

Le modulo des deux entiers 32 bits saisie par l'utilisateur.

Voir également

[modulo\(int64_t, int64_t\)](#), [modulo\(float, float\)](#), [modulo\(double, double\)](#)

```
11 {
12     return operandeA % operandeB;
13 }
```

3.3.1.2 int64_t modulo (int64_t *operandeA*, int64_t *operandeB*)

Fonction de modulo de deux nombre de type int64_t.

Paramètres

<i>operandeA</i>	est un entier sur 64 bits
<i>operandeB</i>	est un entier sur 64 bits

Renvoie

Le modulo des deux entiers 64 bits saisie par l'utilisateur.

Voir également

[modulo\(float, float\)](#), [modulo\(double, double\)](#), [modulo\(int32_t, int32_t\)](#)

```
24 {
25     return operandeA % operandeB;
26 }
```

3.4 Référence du fichier /home/bebo/Documents/BTS/SI6/TP/tp_calculatrice_Cpp/fichiers-Decoupes/multiplication.cpp

Fonctions

- [int32_t multiplication](#) (int32_t *operandeA*, int32_t *operandeB*)
Fonction de soustraction de deux nombre de type int32_t.
- [int64_t multiplication](#) (int64_t *operandeA*, int64_t *operandeB*)
Fonction de multiplication de deux nombre de type int64_t.
- [float multiplication](#) (float *operandeA*, float *operandeB*)
Fonction de multiplication de deux nombre de type Float.
- [double multiplication](#) (double *operandeA*, double *operandeB*)
Fonction de multiplication de deux nombre de type Double.

3.4.1 Documentation des fonctions

3.4.1.1 int32_t multiplication (int32_t *operandeA*, int32_t *operandeB*)

Fonction de soustraction de deux nombre de type int32_t.

Paramètres

<i>operandeA</i>	est un entier sur 32 bits
<i>operandeB</i>	est un entier sur 32 bits

Renvoie

La multiplication des deux entiers 32 bits saisie par l'utilisateur.

Voir également

[multiplication\(int64_t, int64_t\)](#), [multiplication\(float, float\)](#), [multiplication\(double, double\)](#)

```
11 {  
12     return operandeA * operandeB;  
13 }
```

3.4.1.2 int64_t multiplication (int64_t *operandeA*, int64_t *operandeB*)

Fonction de multiplication de deux nombre de type int64_t.

Paramètres

<i>operandeA</i>	est un entier sur 64 bits
<i>operandeB</i>	est un entier sur 64 bits

Renvoie

La multiplication des deux entiers 64 bits saisie par l'utilisateur.

Voir également

[multiplication\(float, float\)](#), [multiplication\(double, double\)](#), [multiplication\(int32_t, int32_t\)](#)

```
24 {  
25     return operandeA * operandeB;  
26 }
```

3.4.1.3 float multiplication (float *operandeA*, float *operandeB*)

Fonction de multiplication de deux nombre de type Float.

Paramètres

<i>operandeA</i>	est un Float
<i>operandeB</i>	est un Float

Renvoie

La multiplication des deux Float saisie par l'utilisateur.

Voir également

[multiplication\(double, double\)](#), [multiplication\(int32_t, int32_t\)](#), [multiplication\(int64_t, int64_t\)](#)

```
37 {  
38     return operandeA * operandeB;  
39 }
```

3.4.1.4 double multiplication (double *operandeA*, double *operandeB*)

Fonction de multiplication de deux nombre de type Double.

Paramètres

<i>operandeA</i>	est un Double
<i>operandeB</i>	est un Double

Renvoie

La multiplication des deux Double saisie par l'utilisateur.

Voir également

[multiplication\(int32_t, int32_t\)](#), [multiplication\(int64_t, int64_t\)](#), [multiplication\(float, float\)](#)

```
50 {
51     return operandeA * operandeB;
52 }
```

3.5 Référence du fichier /home/bebo/Documents/BTS/SI6/TP/tp_calculatrice_Cpp/fichiers-Decoupes/soustraction.cpp

Fonctions

- [int32_t soustraction](#) (int32_t operandeA, int32_t operandeB)
Fonction de soustraction de deux nombre de type int32_t.
- [int64_t soustraction](#) (int64_t operandeA, int64_t operandeB)
Fonction de soustraction de deux nombre de type int64_t.
- [float soustraction](#) (float operandeA, float operandeB)
Fonction de soustraction de deux nombre de type Float.
- [double soustraction](#) (double operandeA, double operandeB)
Fonction de soustraction de deux nombre de type Double.

3.5.1 Documentation des fonctions

3.5.1.1 int32_t soustraction (int32_t operandeA, int32_t operandeB)

Fonction de soustraction de deux nombre de type int32_t.

Paramètres

<i>operandeA</i>	est un entier sur 32 bits
<i>operandeB</i>	est un entier sur 32 bits

Renvoie

La soustraction des deux entiers 32 bits saisie par l'utilisateur.

Voir également

[soustraction\(int64_t, int64_t\)](#), [soustraction\(float, float\)](#), [soustraction\(double, double\)](#)

```
11 {
12     return operandeA - operandeB;
13 }
```

3.5.1.2 int64_t soustraction (int64_t operandeA, int64_t operandeB)

Fonction de soustraction de deux nombre de type int64_t.

Paramètres

<i>operandeA</i>	est un entier sur 64 bits
<i>operandeB</i>	est un entier sur 64 bits

Renvoie

La soustraction des deux entiers 64 bits saisie par l'utilisateur.

Voir également

[soustraction\(float, float\)](#), [soustraction\(double, double\)](#), [soustraction\(int32_t, int32_t\)](#)

```
24 {  
25     return operandeA - operandeB;  
26 }
```

3.5.1.3 float soustraction (float *operandeA*, float *operandeB*)

Fonction de soustraction de deux nombre de type Float.

Paramètres

<i>operandeA</i>	est un Float
<i>operandeB</i>	est un Float

Renvoie

La soustraction des deux Float saisie par l'utilisateur.

Voir également

[soustraction\(double, double\)](#), [soustraction\(int32_t, int32_t\)](#), [soustraction\(int64_t, int64_t\)](#)

```
37 {  
38     return operandeA - operandeB;  
39 }
```

3.5.1.4 double soustraction (double *operandeA*, double *operandeB*)

Fonction de soustraction de deux nombre de type Double.

Paramètres

<i>operandeA</i>	est un Double
<i>operandeB</i>	est un Double

Renvoie

La soustraction des deux Double saisie par l'utilisateur.

Voir également

[soustraction\(int32_t, int32_t\)](#), [soustraction\(int64_t, int64_t\)](#), [soustraction\(float, float\)](#)

```
50 {  
51     return operandeA - operandeB;  
52 }
```


3.6 Référence du fichier /home/bebo/Documents/BTS/SI6/TP/tp_calculatrice_Cpp/main.cpp

Calculatrice affichant le résultat des valeurs définit au préalable dans le programme.

```
#include <iostream>
#include <stdint.h>
#include "../fichiersDecoupes/addition.cpp"
#include "../fichiersDecoupes/soustraction.cpp"
#include "../fichiersDecoupes/multiplication.cpp"
#include "../fichiersDecoupes/division.cpp"
#include "../fichiersDecoupes/modulo.cpp"
```

Fonctions

— int `main` ()

3.6.1 Description détaillée

Calculatrice affichant le résultat des valeurs définit au préalable dans le programme. Calculatrice affichant le résultat des valeurs définit au préalable\$.

Auteur

Pierrick.B

Version

3.0

Date

février 2015

Ceci est le coeur du programme. Le programme est découpé en 5 autres fichiers qui seront appelés ici afin de l'optimiser. Il y a un fichier par opérande +, -, x, ÷ et modulo.

Auteur

Pierrick.B

Version

3.0

Date

février 2015

Ceci est le coeur du programme. Le programme est découpé en 5 au\$

Définition dans le fichier main.cpp.

3.6.2 Documentation des fonctions

3.6.2.1 int `main` ()

```
35 {
36     // Affichage du titre Calculatrice C++ :
37     cout << "----\nCalculatrice C++\n---\n\n";
38
39     // Définition des variables qui seront calculés
```

```
40         int16_t a=321, b=23;
41
42         int32_t c=321, d=23;
43
44         float e=2324, f=25;
45
46         double g=67863, h=432;
47
48         // Affichage des variables définit ci-dessus pour faciliter la compréhension à l'utilisateur
49         cout << "Valeur pour calcul : " << endl;
50         cout << "Valeur de a : " << a << endl;
51         cout << "Valeur de b : " << b << endl;
52         cout << "Valeur de c : " << c << endl;
53         cout << "Valeur de d : " << d << endl;
54         cout << "Valeur de e : " << e << endl;
55         cout << "Valeur de f : " << f << endl;
56         cout << "Valeur de g : " << g << endl;
57         cout << "Valeur de h : " << h << endl << endl;
58
59         // Affichage du résultat sous tous les types de variables (int32, int64, float, double)
60         // Avec également les 5 opérandes
61         cout << "Addition : " << ajouter(a,b) << endl;
62         cout << "Soustraction : " << soustraction(a,b) << endl;
63         cout << "Multiplication : " << multiplication(a,b) << endl;
64         cout << "Division : " << division(a,b) << endl;
65         cout << "Modulo : " << modulo(a,b) << endl;
66
67         return 0;
68 }
```