

tp\_calculatrice\_Cpp

1

Généré par Doxygen 1.8.6

Mercredi 4 Février 2015 14 :51 :27



# **Table des matières**



# Chapitre 1

## Index des fichiers

### 1.1 Liste des fichiers

Liste de tous les fichiers avec une brève description :

<a href="#">addition.cpp</a>	..	??
<a href="#">division.cpp</a>	..	??
<a href="#">main.cpp</a>	..	??
<a href="#">modulo.cpp</a>	..	??
<a href="#">multiplication.cpp</a>	..	??
<a href="#">soustraction.cpp</a>	..	??



## Chapitre 2

# Documentation des fichiers

### 2.1 Référence du fichier 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`.*

#### 2.1.1 Documentation des fonctions

##### 2.1.1.1 `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 }
```

##### 2.1.1.2 `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 }
```

**2.1.1.3 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 }
```

**2.1.1.4 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 }
```



## 2.2 Référence du fichier 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.*

### 2.2.1 Documentation des fonctions

#### 2.2.1.1 `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 }
```

#### 2.2.1.2 `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 }
```

#### 2.2.1.3 `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 }
```

**2.2.1.4 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 }
```

**2.3 Référence du fichier main.cpp**

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

**Fonctions**

— int [main](#) ()

**2.3.1 Documentation des fonctions****2.3.1.1 int main ( )**

```
12 {
```

```

13 // Affichage titre Calculatrice C++ :
14 cout << "----\nCalculatrice C++\n----\n\n";
15 int16_t a=321, b=23;
16
17 int32_t c=321, d=23;
18
19 float e=2324, f=25;
20
21 double g=67863, h=432;
22
23 cout << "Valeur pour calcul : " << endl;
24 cout << "Valeur de a : " << a << endl;
25 cout << "Valeur de b : " << b << endl;
26 cout << "Valeur de c : " << c << endl;
27 cout << "Valeur de d : " << d << endl;
28 cout << "Valeur de e : " << e << endl;
29 cout << "Valeur de f : " << f << endl;
30 cout << "Valeur de g : " << g << endl;
31 cout << "Valeur de h : " << h << endl << endl;
32
33
34 cout << "Addition : " << ajouter(a,b) << endl;
35 cout << "Soustraction : " << soustraction(a,b) << endl;
36 cout << "Multiplication : " << multiplication(a,b) << endl;
37 cout << "Division : " << division(a,b) << endl;
38 cout << "Modulo : " << modulo(a,b) << endl;
39
40 return 0;
41 }

```

## 2.4 Référence du fichier 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.*

### 2.4.1 Documentation des fonctions

#### 2.4.1.1 `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 }

```

#### 2.4.1.2 `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 }
```

## 2.5 Référence du fichier 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.*

### 2.5.1 Documentation des fonctions

#### 2.5.1.1 `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 }
```

#### 2.5.1.2 `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 }
```

2.5.1.3 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 }
```

2.5.1.4 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 }
```

## 2.6 Référence du fichier 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.*

### 2.6.1 Documentation des fonctions

#### 2.6.1.1 `int32_t soustraction ( int32_t operandeA, int32_t operandeB )`

Fonction de soustraction 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 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 }
```

#### 2.6.1.2 `int64_t soustraction ( int64_t operandeA, int64_t operandeB )`

Fonction de soustraction 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 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 }
```

#### 2.6.1.3 `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 }
```

2.6.1.4 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 }
```