Calculatrice C++

3

Généré par Doxygen 1.8.6

Dimanche 8 Février 2015 22 :19 :12

Table des matières

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 :

Chapitre 3

Documentation des fichiers

3.1 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.

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

operandeA	est un entier sur 32 bits
operandeB	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

operandeA	est un entier sur 64 bits
operandeB	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

operandeA	est un Float
operandeB	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

operandeA	est un Double
operandeB	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 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

operandeA	est un entier sur 32 bits
operandeB	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

operandeA	est un entier sur 64 bits
operandeB	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

operandeA	est un Float
operandeB	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

operandeA	est un Double
operandeB	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 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 ()

3.3.1 Documentation des fonctions

3.3.1.1 int main ( )

```
26 {
```

```
27     // Affichage titre Calculatrice C++ :
28     cout << "---\nCalculatrice C++\n---\n\n";
29     int16_t a=321, b=23;
30
31     int32_t c=321, d=23;
32
33     float e=2324, f=25;
34
35     double g=67863, h=432;
36
37     cout << "Valeur pour calcul : " << endl;
38     cout << "Valeur de  a : " << a << endl;
39     cout << "Valeur de  b : " << b << endl;
40     cout << "Valeur de  c : " << c << endl;
41     cout << "Valeur de  d : " << d << endl;
42     cout << "Valeur de  e : " << e << endl;
43     cout << "Valeur de  f : " << f << endl;
44     cout << "Valeur de  g : " << g << endl;
45     cout << "Valeur de  h : " << h << endl << endl;
46
47
48     cout << "Addition : " << ajouter(a,b) << endl;
49     cout << "Soustraction : " << soustraction(a,b) << endl;
50     cout << "Multiplication : " << multiplication(a,b) << endl;
51     cout << "Division : " <<division(a,b) << endl;
52     cout << "Modulo : " << modulo(a,b) << endl;
53
54     return 0;
55 }
```

3.4 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.

3.4.1 Documentation des fonctions

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

Fonction de modulo de deux nombre de type int32_t.

Paramètres

operandeA	est un entier sur 32 bits
operandeB	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 }
```

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

Fonction de modulo de deux nombre de type int64_t.

Paramètres

operandeA	est un entier sur 64 bits
operandeB	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 }
```

3.5 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.

3.5.1 Documentation des fonctions

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

Fonction de soustraction de deux nombre de type int32_t.

Paramètres

operandeA	est un entier sur 32 bits
operandeB	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 }
```

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

Fonction de multiplication de deux nombre de type int64_t.

Paramètres

operandeA	est un entier sur 64 bits
operandeB	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 }
```

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

Fonction de multiplication de deux nombre de type Float.

Paramètres

operandeA	est un Float
operandeB	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 }
```

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

Fonction de multiplication de deux nombre de type Double.

Paramètres

operandeA	est un Double
operandeB	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 }
```

3.6 Référence du fichier save/division_conflict-20150202-160133.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.

3.6.1 Documentation des fonctions

3.6.1.1 int32_t division (int32_t *operandeA*, int32_t *operandeB*)

Fonction de division de deux nombre de type int32_t.

Paramètres

operandeA	est un entier sur 32 bits
operandeB	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 }
```

3.6.1.2 int64_t division (int64_t *operandeA*, int64_t *operandeB*)

Fonction de division de deux nombre de type int64_t.

Paramètres

operandeA	est un entier sur 64 bits
operandeB	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 }
```

3.6.1.3 float division (float *operandeA*, float *operandeB*)

Fonction de division de deux nombre de type Float.

Paramètres

operandeA	est un Float
operandeB	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 }
```

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

Fonction de division de deux nombre de type Double.

Paramètres

operandeA	est un Double
operandeB	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 }
```

3.7 Référence du fichier save/mainInteractionOperateur.cpp

```
#include <iostream>
```

Fonctions

— int main ()

3.7.1 Documentation des fonctions

3.7.1.1 int main ( )

```
5 {
6     // Declaration des variables entiers
7     int addition, soustraction, multiplication,
      division, modulo, entier1, entier2, moduloCalcul;
8     char operateur;
9
10     // Demande a l utilisateur, affichage
11     do{
12         cout << "---\nCalculatrice C++ par Pierrick Bobet\n---\nChoisir votre opérateur (+, -, x, /, %) : "
      ;
```

```
13          cin >> operateur;
14          cout << operateur;
15      } while (operateur != '+' && operateur != '-' && operateur != 'x' && operateur != '/' && operateur !=
    '%');
16
17      cout << "---\nSaisir le premier entier de votre calcul : ";
18      cin >> entier1;
19
20      cout << "\nSaisir le deuxieme entier de votre calcul : ";
21      cin >> entier2;
22
23      // Calcul avec l operateur
24          // ADDITION
25          if (operateur == '+'){
26          entier1 = entier1 + entier2;
27          }
28
29          // SOUSTRACTION
30          else if (operateur == '-'){
31              if (entier1 > entier2){
32                  entier1 = entier1 - entier2;
33              }
34              else {
35                  entier1 = entier2 - entier1;
36              }
37          }
38          // MULTIPLICATION
39          else if (operateur == 'x'){
40          entier1 = entier1 * entier2;
41          }
42
43          // DIVISION
44          else if (operateur == '/'){
45              while (entier1 == 0){
46                      cout << "---\n!!! Division impossible par 0 !!!" << endl;
47                  cout << "---\nSaisir le premier entier de votre calcul : ";
48                      cin >> entier1;
49                  }
50
51              while (entier2 == 0){
52                  cout << "---\n!!! Division impossible par 0 !!!" << endl;
53                  cout << "---\nSaisir le deuxieme entier de votre calcul : ";
54                  cin >> entier2;
55              }
56
57                  if (entier1 != 0){
58                          entier1 = entier1 / entier2;
59                  }
60          }
61
62          // MODULO
63          else if (operateur == '%'){
64              entier1 = entier1%entier2;
65              //entier1 = moduloCalcul;
66          }
67      // Affichage du résultat
68          cout << "\n\n---\n***\nLe résultat de l'opération est : " << entier1 << "\n***" << endl;
69
70  return 0;
71
72
73
74  }
```

3.8 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.

3.8.1 Documentation des fonctions

3.8.1.1 int32_t soustraction (int32_t *operandeA*, int32_t *operandeB*)

Fonction de soustraction de deux nombre de type int32_t.

Paramètres

operandeA	est un entier sur 32 bits
operandeB	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 }
```

3.8.1.2 int64_t soustraction (int64_t *operandeA*, int64_t *operandeB*)

Fonction de soustraction de deux nombre de type int64_t.

Paramètres

operandeA	est un entier sur 64 bits
operandeB	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 }
```

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

Fonction de soustraction de deux nombre de type Float.

Paramètres

operandeA	est un Float
operandeB	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 }
```

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

Fonction de soustraction de deux nombre de type Double.

Paramètres

operandeA	est un Double
operandeB	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 }
```