Documentation du TP Calculatrice

Version 3 - Alexis WIDMER

Généré par Doxygen 1.8.1.2

Mercredi Février 4 2015 14 :35 :21

Table des matières

Chapitre 1

Index des fichiers

1.1 Liste des fichiers

Liste de tous les fichiers avec une brève description :

Chapitre 2

Documentation des fichiers

2.1 Référence du fichier addition.cpp

```
#include <iostream>
#include <stdint.h>
```

Fonctions

- int32_t addition (int32_t oppA, int32_t oppB)
  Addition de nombres int32.
- int64_t addition (int64_t oppA, int64_t oppB)
  Addition de nombres int64.
- float addition (float oppA, float oppB)
  Addition de nombres float.
- double addition (double oppA, double oppB)
  Addition de nombres double.

2.1.1 Documentation des fonctions

2.1.1.1 int32_t addition (int32_t *oppA*, int32_t *oppB*)

Addition de nombres int32.

La fonction addition n'est pas la meme suivant les parametres (leur type) On appelle ca une fonction polymorphe

Paramètres

oppA	est un entier 32 bits
oppB	est un entier 32 bits

Renvoie

La somme des 2 entiers 32 bits donnes en parametres

Voir également

addition(int64_t, int64_t), addition(float, float), addition(double, double).

Définition à la ligne 16 du fichier addition.cpp.

```
{
        return (oppA+oppB);
}
```

2.1.1.2 int64_t addition (int64_t *oppA*, int64_t *oppB*)

Addition de nombres int64.

La fonction addition n'est pas la meme suivant les parametres (leur type) On appelle ca une fonction polymorphe

Paramètres

oppA	est un entier 64 bits
oppB	est un entier 64 bits

Renvoie

La somme des 2 entiers 64 bits donnes en parametres

Voir également

addition(int32_t, int32_t), addition(float, float), addition(double, double).

Définition à la ligne 33 du fichier addition.cpp.

```
{
        return (oppA+oppB);
}
```

2.1.1.3 float addition (float *oppA*, float *oppB*)

Addition de nombres float.

La fonction addition n'est pas la meme suivant les parametres (leur type) On appelle ca une fonction polymorphe

Paramètres

oppA	est un float
oppB	est un float

Renvoie

La somme des 2 float donnes en parametres

Voir également

addition(int32_t, int32_t), addition(int64_t, int64_t), addition(double, double).

Définition à la ligne 50 du fichier addition.cpp.

```
{
        return (oppA+oppB);
}
```

2.1.1.4 double addition (double *oppA*, double *oppB*)

Addition de nombres double.

La fonction addition n'est pas la meme suivant les parametres (leur type) On appelle ca une fonction polymorphe

Paramètres

oppA	est un entier double
oppB	est un entier double

Renvoie

La somme des 2 double donnes en parametres

Voir également

addition(int32_t, int32_t), addition(ibnt64_t, int64_t), addition(float, float).

Définition à la ligne 67 du fichier addition.cpp.

```
{
        return (oppA+oppB);
}
```

2.2 Référence du fichier division.cpp

```
#include <iostream>
#include <stdint.h>
```

Fonctions

- int32_t division (int32_t oppA, int32_t oppB)
  Division de nombres int32.
- int64_t division (int64_t oppA, int64_t oppB)
  Division de nombres int64.
- float division (float oppA, float oppB)
  Division de nombres float.
- double division (double oppA, double oppB)
  Division de nombres double.

2.2.1 Documentation des fonctions

2.2.1.1 int32_t division (int32_t *oppA*, int32_t *oppB*)

Division de nombres int32.

La fonction division n'est pas la meme suivant les parametres (leur type) On appelle ca une fonction polymorphe

Paramètres

oppA	est un entier 32 bits
oppB	est un entier 32 bits

Renvoie

La division des 2 entiers 32 bits donnes en parametres

Voir également

division(int64_t, int64_t), division(float, float), division(double, double).

Définition à la ligne 16 du fichier division.cpp.

```
{
        return (oppA/oppB);
}
```

2.2.1.2 int64_t division (int64_t *oppA*, int64_t *oppB*)

Division de nombres int64.

La fonction division n'est pas la meme suivant les parametres (leur type) On appelle ca une fonction polymorphe

Paramètres

oppA	est un entier 64 bits
oppB	est un entier 64 bits

Renvoie

La division des 2 entiers 64 bits donnes en parametres

Voir également

division(int32_t, int32_t), division(float, float), division(double, double).

Définition à la ligne 33 du fichier division.cpp.

```
{
        return (oppA/oppB);
}
```

2.2.1.3 float division (float *oppA*, float *oppB*)

Division de nombres float.

La fonction division n'est pas la meme suivant les parametres (leur type) On appelle ca une fonction polymorphe

Paramètres

oppA	est un float
oppB	est un float

Renvoie

La division des 2 float donnes en parametres

Voir également

division(int32_t, int32_t), division(int64_t, int64_t), division(double, double).

Définition à la ligne 50 du fichier division.cpp.

```
{
        return (oppA/oppB);
}
```

2.2.1.4 double division (double *oppA*, double *oppB*)

Division de nombres double.

La fonction division n'est pas la meme suivant les parametres (leur type) On appelle ca une fonction polymorphe

Paramètres

oppA	est un double
oppB	est un double

Renvoie

La division des 2 double donnes en parametres

Voir également

division(int32_t, int32_t), division(int64_t, int64_t), division(float, float).

Définition à la ligne 67 du fichier division.cpp.

```
{
        return (oppA/oppB);
}
```

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 ( )

Declaration et initialisation des nombres de type int32, int64, float et double

Affichage du contenu des nombres

Affichage des additions

Affichage des soustractions

Affichage des multiplications

Affichage des divisions

Affichage des modulos

Définition à la ligne 10 du fichier main.cpp.

```
{
int32_t a = 864;
int32_t b = 156;
int64_t c = 845;
int64_t d = 625;
float e = 456.25;
float f = 488.64;
double g = 78965.45;
double h = 424242.42;

cout << "Voici les nombres :\n-Entiers sur 32 bits : 864 et 156\n-Entiers sur
       64 bits : 845 et 625\n-Float : 456.25 et 488.64\n-Double : 78965.45 et 424242.42"
      ;

cout << "\n\nResultat des additions :\n-int32 : " << addition(a, b) <<
      "\n-int64 : " << addition(c, d) << "\n-float : " << addition(e,
      f) << "\n-double : " << addition(g, h);
```

```
cout << "\n\nResultat des soustractions :\n-int32 : " << soustraction
      (a, b) << "\n-int64 : " << soustraction(c, d) << "\n-float : " <<
      soustraction(e, f) << "\n-double : " << soustraction(g,
       h);

cout << "\n\nResultat des multiplications :\n-int32 : " << multiplication
      (a, b) << "\n-int64 : " << multiplication(c, d) << "\n-float : " <
      < multiplication(e, f) << "\n-double : " << multiplication
      (g, h);

cout << "\n\nResultat des divisions :\n-int32 : " << division(a, b) <<
      "\n-int64 : " << division(c, d) << "\n-float : " << division(e,
      f) << "\n-double : " << division(g, h);

cout << "\n\nResultat des modulos :\n-int32 : " << modulo(a, b) << "\n
      -int64 : " << modulo(c, d);

return 0;
}
```

2.4 Référence du fichier modulo.cpp

```
#include <iostream>
#include <stdint.h>
```

Fonctions

– int32_t modulo (int32_t oppA, int32_t oppB)
Modulo de nombres int32.
– int64_t modulo (int64_t oppA, int64_t oppB)
Modulo de nombres int64.

2.4.1 Documentation des fonctions

2.4.1.1 int32_t modulo (int32_t *oppA*, int32_t *oppB*)

Modulo de nombres int32.

La fonction modulo n'est pas la meme suivant les parametres (leur type) On appelle ca une fonction polymorphe

Paramètres

oppA	est un entier 32 bits
oppB	est un entier 32 bits

Renvoie

La multiplication des 2 entiers 32 bits donnes en parametres

Voir également

modulo(int64_t, int64_t)

Définition à la ligne 16 du fichier modulo.cpp.

```
{
        return (oppA%oppB);
}
```

2.4.1.2 int64_t modulo (int64_t *oppA*, int64_t *oppB*)

Modulo de nombres int64.

La fonction modulo n'est pas la meme suivant les parametres (leur type) On appelle ca une fonction polymorphe

Paramètres

oppA	est un entier 64 bits
oppB	est un entier 64 bits

Renvoie

La multiplication des 2 entiers 64 bits donnes en parametres

Voir également

modulo(int32_t, int32_t)

Définition à la ligne 33 du fichier modulo.cpp.

```
{
        return (oppA%oppB);
}
```

2.5 Référence du fichier multiplication.cpp

```
#include <iostream>
#include <stdint.h>
```

Fonctions

- int32_t multiplication (int32_t oppA, int32_t oppB)
  Multiplication de nombres int32.
- int64_t multiplication (int64_t oppA, int64_t oppB)
  Multiplication de nombres int64.
- float multiplication (float oppA, float oppB)
  Multiplication de nombres float.
- double multiplication (double oppA, double oppB)
  Multiplication de nombres double.

2.5.1 Documentation des fonctions

2.5.1.1 int32_t multiplication (int32_t *oppA*, int32_t *oppB*)

Multiplication de nombres int32.

La fonction multiplication n'est pas la meme suivant les parametres (leur type) On appelle ca une fonction polymorphe

Paramètres

oppA	est un entier 32 bits
oppB	est un entier 32 bits

Renvoie

La multiplication des 2 entiers 32 bits donnes en parametres

Voir également

multiplication(int64_t, int64_t), multiplication(float, float), multiplication(double, double).

Définition à la ligne 16 du fichier multiplication.cpp.

```
{
        return (oppA*oppB);
}
```

2.5.1.2 int64_t multiplication (int64_t *oppA*, int64_t *oppB*)

Multiplication de nombres int64.

La fonction multiplication n'est pas la meme suivant les parametres (leur type) On appelle ca une fonction polymorphe

Paramètres

oppA	est un entier 64 bits
oppB	est un entier 64 bits

Renvoie

La multiplication des 2 entiers 64 bits donnes en parametres

Voir également

multiplication(int32_t, int32_t), multiplication(float, float), multiplication(double, double).

Définition à la ligne 33 du fichier multiplication.cpp.

```
{
        return (oppA*oppB);
}
```

2.5.1.3 float multiplication (float *oppA*, float *oppB*)

Multiplication de nombres float.

La fonction multiplication n'est pas la meme suivant les parametres (leur type) On appelle ca une fonction polymorphe

Paramètres

oppA	est un float
oppB	est un float

Renvoie

La multiplication des 2 float donnes en parametres

Voir également

multiplication(int32_t, int32_t), multiplication(int64_t, int64_t), multiplication(double, double).

Définition à la ligne 50 du fichier multiplication.cpp.

```
{
        return (oppA*oppB);
}
```

2.5.1.4 double multiplication (double *oppA*, double *oppB*)

Multiplication de nombres double.

La fonction multiplication n'est pas la meme suivant les parametres (leur type) On appelle ca une fonction polymorphe

Paramètres

oppA	est un double
oppB	est un double

Renvoie

La multiplication des 2 double donnes en parametres

Voir également

multiplication(int32_t, int32_t), multiplication(int64_t, int64_t), multiplication(float, float).

Définition à la ligne 67 du fichier multiplication.cpp.

```
{
        return (oppA*oppB);
}
```

2.6 Référence du fichier soustraction.cpp

```
#include <iostream>
#include <stdint.h>
```

Fonctions

- int32_t soustraction (int32_t oppA, int32_t oppB)
  Soustraction de nombres int32.
- int64_t soustraction (int64_t oppA, int64_t oppB)
  Soustraction de nombres int64.
- float soustraction (float oppA, float oppB)
  Soustraction de nombres float.
- double soustraction (double oppA, double oppB)
  Soustraction de nombres double.

2.6.1 Documentation des fonctions

2.6.1.1 int32_t soustraction (int32_t *oppA*, int32_t *oppB*)

Soustraction de nombres int32.

La fonction soustraction n'est pas la meme suivant les parametres (leur type) On appelle ca une fonction polymorphe

Paramètres

oppA	est un entier 32 bits
oppB	est un entier 32 bits

Renvoie

La soustraction des 2 entiers 32 bits donnes en parametres

Voir également

soustraction(int64_t, int64_t), soustraction(float, float), soustraction(double, double).

Définition à la ligne 16 du fichier soustraction.cpp.

```
{
        return (oppA-oppB);
}
```

2.6.1.2 int64_t soustraction (int64_t *oppA*, int64_t *oppB*)

Soustraction de nombres int64.

La fonction soustraction n'est pas la meme suivant les parametres (leur type) On appelle ca une fonction polymorphe

Paramètres

oppA	est un entier 64 bits
oppB	est un entier 64 bits

Renvoie

La soustraction des 2 entiers 64 bits donnes en parametres

Voir également

soustraction(int32_t, int32_t), soustraction(float, float), soustraction(double, double).

Définition à la ligne 33 du fichier soustraction.cpp.

```
{
        return (oppA-oppB);
}
```

2.6.1.3 float soustraction (float *oppA*, float *oppB*)

Soustraction de nombres float.

La fonction soustraction n'est pas la meme suivant les parametres (leur type) On appelle ca une fonction polymorphe

Paramètres

oppA	est un float
oppB	est un float

Renvoie

La soustraction des 2 float donnes en parametres

Voir également

soustraction(int32_t, int32_t), soustraction(int64_t, int64_t), soustraction(double, double).

Définition à la ligne 50 du fichier soustraction.cpp.

```
{
        return (oppA-oppB);
}
```

2.6.1.4 double soustraction (double *oppA*, double *oppB*)

Soustraction de nombres double.

La fonction soustraction n'est pas la meme suivant les parametres (leur type) On appelle ca une fonction polymorphe

Paramètres

oppA	est un double
oppB	est un double

Renvoie

La soustraction des 2 double donnes en parametres

Voir également

soustraction(int32_t, int32_t), soustraction(int64_t, int64_t), soustraction(float, float).

Définition à la ligne 67 du fichier soustraction.cpp.

```
{
        return (oppA-oppB);
}
```

Index