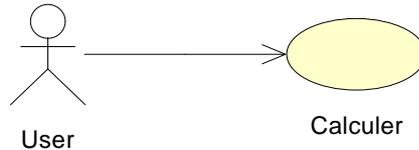


## DOSSIER CALCULETTE BASIC

DOSSIER CALCULETTE BASIC .....	1
LES BESOINS.....	1
<i>Le Cas d'Utilisation Calculer</i> .....	1
<i>Le Diagramme de séquence Calculer</i> .....	1
<i>L'interface avec l'Utilisateur l'IHM</i> .....	2
QUELQUES DÉTAILS DE LA CONCEPTION .....	2
<i>L'architecture de l'application</i> .....	2
<i>L'interface Utilisateur : La boîte de dialogue et la classe CIHMCalculetteDlg</i> .....	2
Boutons, Messages, Fonctions de traitement.....	2
Zone d'édition.....	3
<i>Saisir Opérande A</i> .....	3
<i>Détail d'implémentation de quelques fonctions</i> .....	3
<i>Gérer l'Etat SAISIE_A</i> .....	4
Détail de l'implémentation de la fonction gererEtat() pour l'état SAISIE_A .....	4
Détail de l'implémentation de la fonction traiterEtat() pour l'état SAISIE_A .....	4
ANNEXE.....	5
LES ÉTATS DE LA CALCULETTE.....	5
EXTRAIT DU DIAGRAMME DE CLASSES STATIQUE.....	6
EXTRAIT DU FICHIER CIHMCALCULETTEDLG.H.....	7
EXTRAIT DU FICHIER CIHMCALCULETTEDLG.CPP.....	8
LE FICHIER CALCULETTE.H .....	8

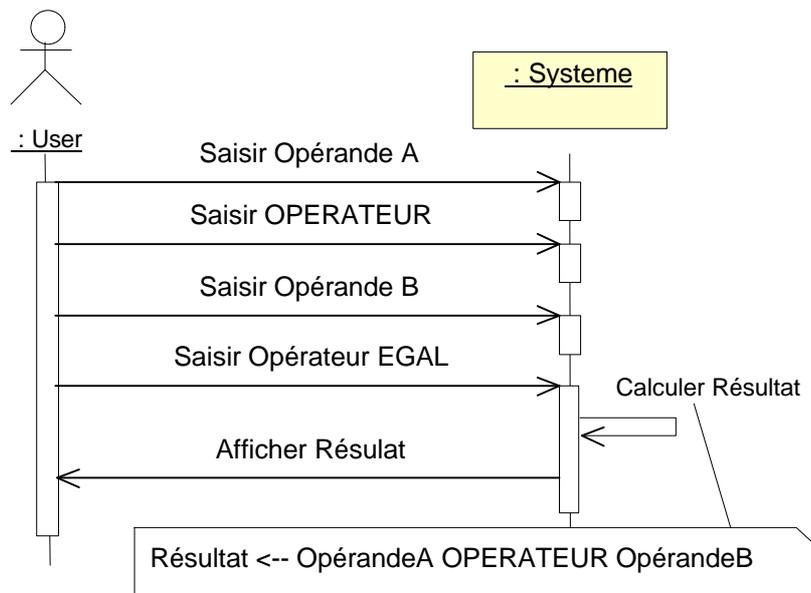
### LES BESOINS

#### Le Cas d'Utilisation Calculer



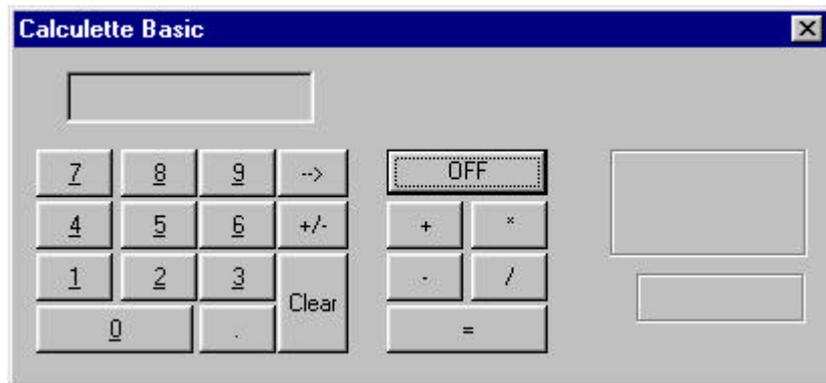
#### Le Diagramme de séquence Calculer

Décrit les interactions entre le système et l'utilisateur décrivant le principe de Calculer



Voir en Annexe le détail de Calculer dans le diagramme « Etats de la Calculette »

## L'interface avec l'Utilisateur l'IHM



Touches	Rôle
1 à 9 et le . (point)	Saisir les caractères pour constituer la valeur de l'opérande
+ - / *	Saisir l'OPERATEUR
=	Saisir l'opérateur EGAL
+/-	Saisir le SIGNE
→	Saisir la commande Effacer le dernier caractère saisi
Clear	Saisir la commande Initialiser la calculatrice
OFF	Saisir la commande Quitter la calculatrice

La zone d'affichage : Affiche la valeur de l'opérande saisi ou le Résultat de l'opération

### QUELQUES DETAILS DE LA CONCEPTION

#### L'architecture de l'application

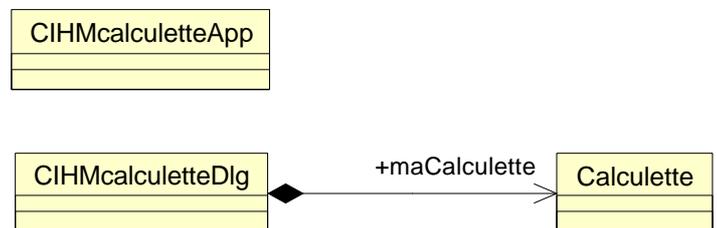
L'application est constituée de 3 classes  
 - Classe CIHMcalculatriceApp gère l'application Windows. Elle dérive de CWinApp

```
class CIHMcalculatriceApp : public CWinApp
{
}
```

- Classe CIHMcalculatriceDlg gère la boîte de dialogue assurant l'interface avec l'utilisateur. La classe dérive de CDialog

```
class CIHMcalculatriceDlg : public CDialog
{
}
```

- Classe Calculatrice assure le traitement.



La classe CIHMcalculatriceDlg est associée à la classe Calculatrice par une agrégation par valeur. Les fonctions de cette classe vont utiliser les fonctions de la classe Calculatrice pour réaliser le traitement.

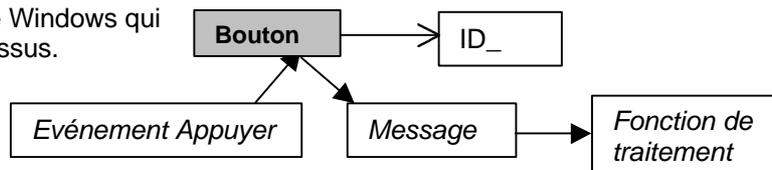
#### L'interface Utilisateur : La boîte de dialogue et la classe CIHMcalculatriceDlg

##### Boutons, Messages, Fonctions de traitement

Au Bouton est associé un identificateur ID\_. Cet identificateur est ensuite pour repérer l'objet bouton dans le programme.

- Au bouton est associé un message Windows qui sera produit lorsque que l'on clique dessus.
- Au message est associé une fonction de traitement

L'effet sur le bouton va se traduire dans le programme par l'appel à une fonction de traitement. *Programmation événementielle*



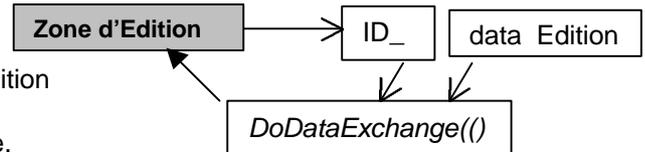
## Exemple le Bouton «1»

Définition de l'identifiant du Bouton «1»	<i>IDC_CHIFFRE_1</i>	resource.h
Associe au Bouton «1», identifié «IDC_CHIFFRE_1» le message «ON_BN_CLICKED» qui se produit lorsque l'on clique dessus. A ce message est associée la fonction de traitement «OnChiffre1»	Table d'interception des messages <i>BEGIN_MESSAGE_MAP(CIHMcalculetteDlg, CDialog)</i> <b>ON_BN_CLICKED (IDC_CHIFFRE_1, OnChiffre1)</b> ..... <i>END_MESSAGE_MAP()</i>	CIHMcalculetteDlg.cpp
La fonction de Traitement du message «ON_BN_CLICKED» du Bouton «1»	<i>afx_msg void OnChiffre1();</i>	CIHMcalculetteDlg.h CIHMcalculetteDlg.cpp

## Zone d'édition,

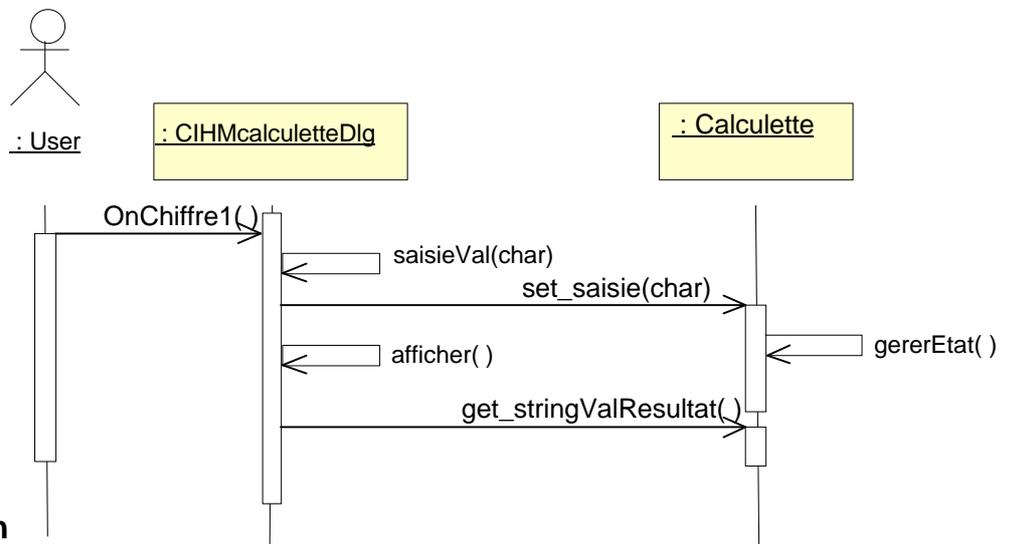
A la zone d'édition est associé un identificateur *ID\_*. Cet identificateur est permet de repérer l'objet zone d'édition dans le programme.

- A la zone d'édition est associé une donnée membre.
- Pour transférer la donnée vers la boite de dialogue on fait appel au service de la fonction membre (D.D.X.) de la boite de dialogue.



Définition de l'identifiant de la zone d'édition	<i>IDC_SAISIE</i>	resource.h
Définition d'une donnée membre pour l'échange des données	<i>CString m_saisieA;</i>	CIHMcalculetteDlg.h
La fonction de transfert des données <i>DoDataExchange()</i> Associe à la zone d'édition, identifié «IDC_SAISIE» à la donnée «m_saisie».	<i>void CIHMcalculetteDlg::DoDataExchange(CDataExchange* pDX) { DDX_Text(pDX, IDC_SAISIE, m_saisie); }</i>	CIHMcalculetteDlg.cpp

## Saisir Opérande A



## Détail d'implémentation de quelques fonctions

```

void Calculette::set_saisie(char value) {
    transition=EVENEMENT_VALEUR;
    carSaisie=value;
    gererEtat();
}
  
```

Voir en Annexe CIHMcalculetteDlg.cpp :

```

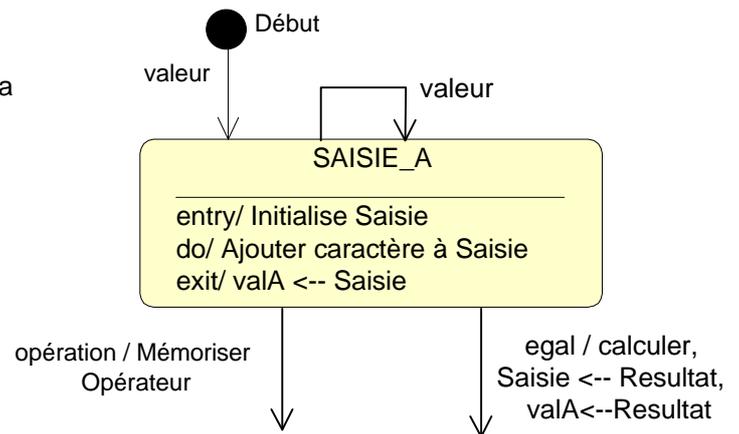
OnChiffre1() ; saisieVal(char value1 ; afficher()
  
```

## Gérer l'Etat SAISIE\_A

Une fonction *gererEtat()* fait évoluer les Etats de la machine à états. Elle appelle une fonction pour effectuer le traitement *traiterEtat()*.

Détail de l'implémentation de la fonction *gererEtat()* pour l'état SAISIE\_A

```
void Calculette::gererEtat() {
    switch(etat){
        case SAISIE_A : {
            switch(transition) {
                case EVENEMENT_VALEUR : traiterEtat(); break;
                case EVENEMENT_OPERATION : {
                    action=EXIT;
                    traiterEtat();
                    action=SORTIE_EVE_OPERATION;
                    traiterEtat();
                    etat=SAISIE_B;
                    action=ENTRY;
                } break;
                case EVENEMENT_EGAL : {
                    action=EXIT;
                    traiterEtat();
                    action=SORTIE_EVE_EGAL;
                    traiterEtat();
                    etat=EGAL;
                    action=DO;
                } break;
            }
        } break;
        .....;
    }
}
```

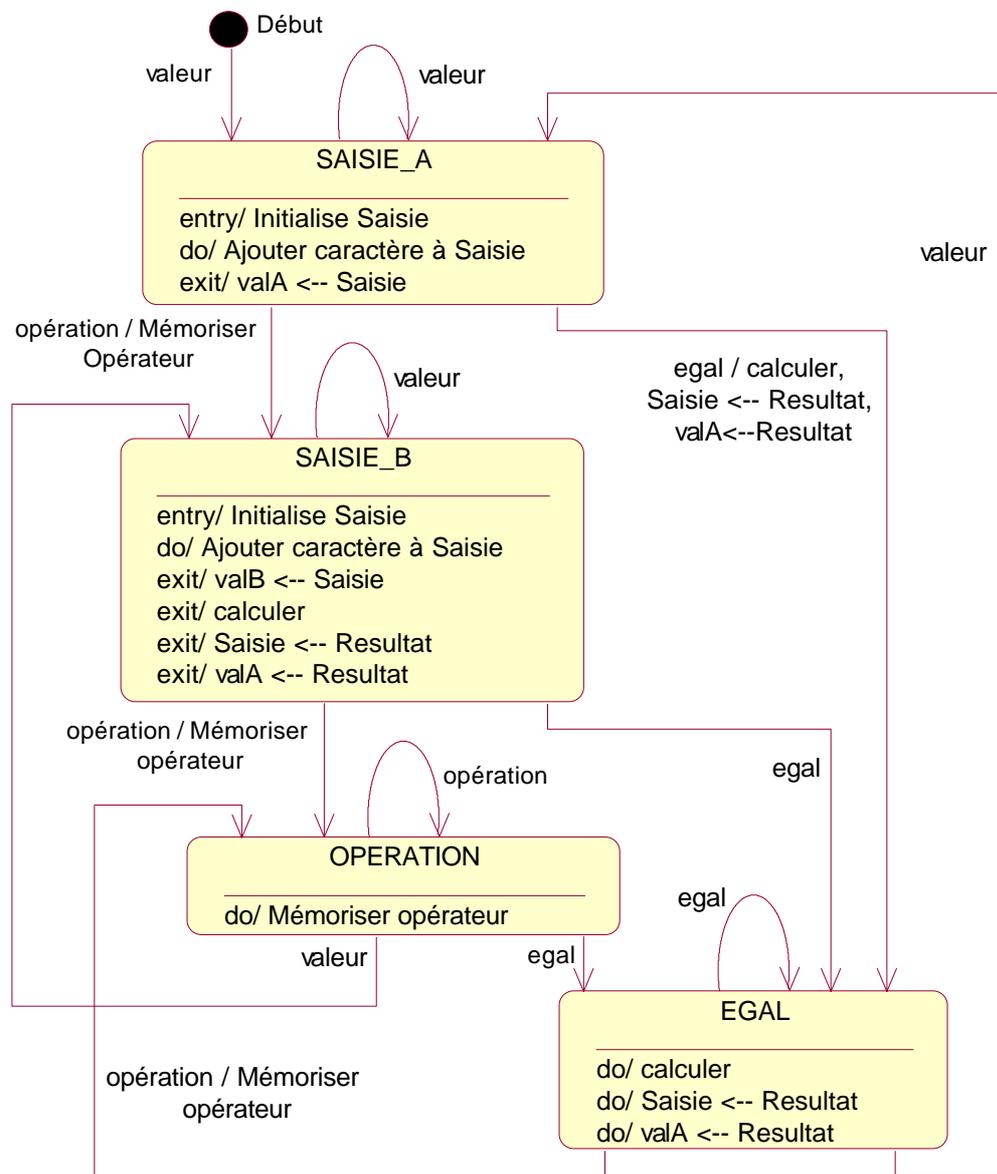


Détail de l'implémentation de la fonction *traiterEtat()* pour l'état SAISIE\_A

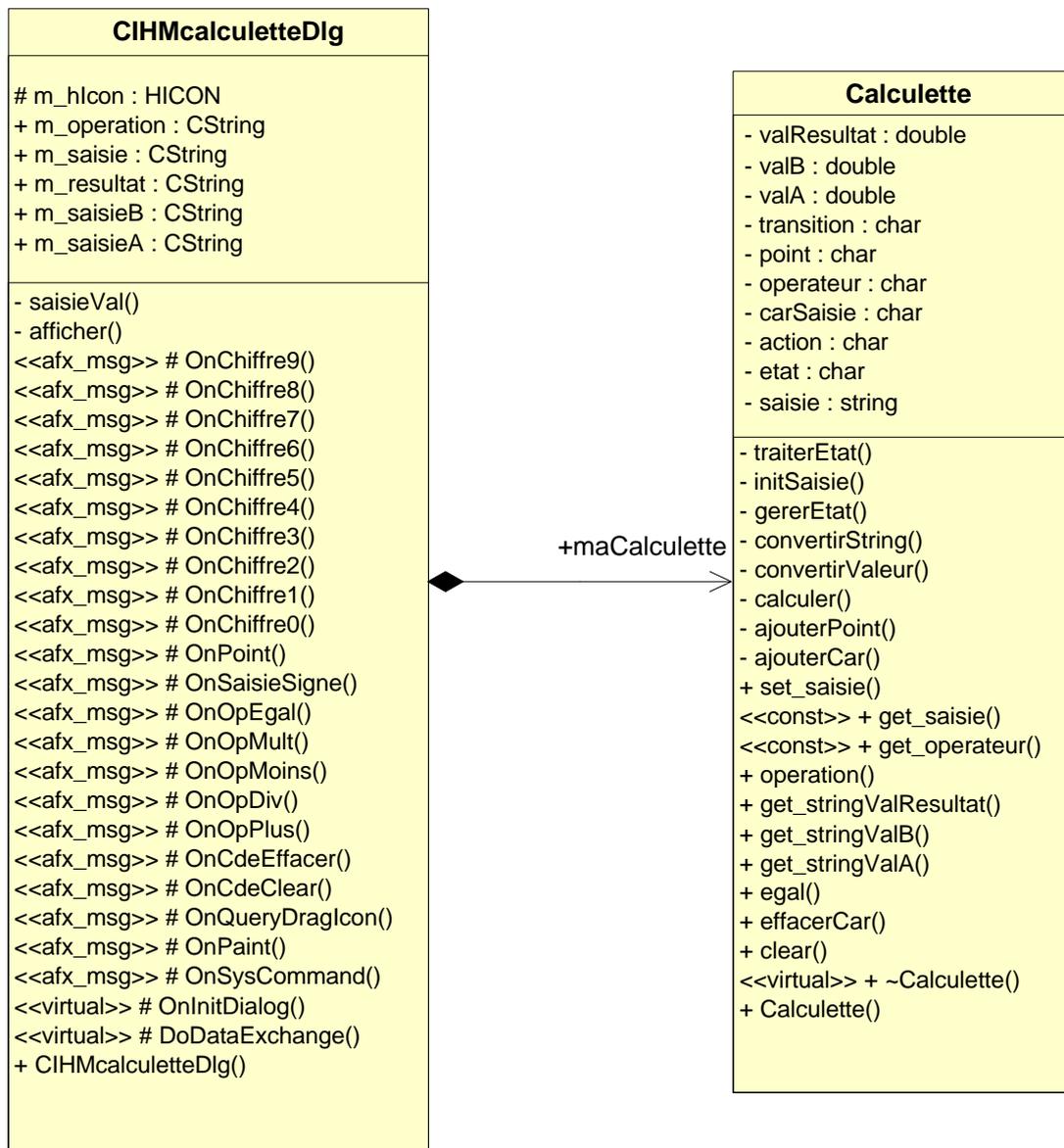
```
void Calculette::traiterEtat() {
    switch(etat) {
        case SAISIE_A : {
            switch(action) {
                case ENTRY : {
                    initSaisie(carSaisie);
                    ajouterCar(carSaisie);
                    action=DO;
                } break;
                case DO : ajouterCar(carSaisie); break;
                case EXIT : valA=convertirString(saisie.data()); break;
                case SORTIE_EVE_OPERATION : operateur=carSaisie; break;
                case SORTIE_EVE_EGAL : calculer(); break;
            }
        } break;
        ..... ;
    }
}
```

## ANNEXE

## LES ETATS DE LA CALCULETTE



## EXTRAIT DU DIAGRAMME DE CLASSES STATIQUE



**EXTRAIT DU FICHIER CIHMALCULETTE.DLG.H**

```
#include "Calcullette.h"

class Calcullette;

class CIHMcalculletteDlg : public Cdialog {
public:
    Calcullette maCalcullette;
    CIHMcalculletteDlg(CWnd* pParent = NULL); // standard constructor

    //{{AFX_DATA(CIHMcalculletteDlg)
    enum { IDD = IDD_CAL2_DIALOG };
    CString      m_saisieA;
    CString      m_saisieB;
    CString      m_resultat;
    CString      m_saisie;
    CString      m_operation;
    //}}AFX_DATA

    // ClassWizard generated virtual function overrides
    //{{AFX_VIRTUAL(CIHMcalculletteDlg)
protected:
    virtual void DoDataExchange(CDataExchange* pDX); //DDX/DDV support
    //}}AFX_VIRTUAL

protected:
    HICON m_hIcon;

    // Generated message map functions
    //{{AFX_MSG(CIHMcalculletteDlg)
    virtual BOOL OnInitDialog();
    afx_msg void OnSysCommand(UINT nID, LPARAM lParam);
    afx_msg void OnPaint();
    afx_msg HCURSOR OnQueryDragIcon();
    afx_msg void OnCdeClear();
    afx_msg void OnCdeEffacer();
    afx_msg void OnOpPlus();
    afx_msg void OnOpDiv();
    afx_msg void OnOpMoins();
    afx_msg void OnOpMult();
    afx_msg void OnOpEgal();
    afx_msg void OnSaisieSigne();
    afx_msg void OnPoint();
    afx_msg void OnChiffre0();
    afx_msg void OnChiffre1();
    afx_msg void OnChiffre2();
    afx_msg void OnChiffre3();
    afx_msg void OnChiffre4();
    afx_msg void OnChiffre5();
    afx_msg void OnChiffre6();
    afx_msg void OnChiffre7();
    afx_msg void OnChiffre8();
    afx_msg void OnChiffre9();
    //}}AFX_MSG
    DECLARE_MESSAGE_MAP()
private :
    void afficher();
    void saisieVal(char value);
};
```

**EXTRAIT DU FICHIER CIHMALCULETTEDLG.CPP**

```

CIHMcalculetteDlg::CIHMcalculetteDlg(CWnd* pParent/*=NULL*/ )
    : CDialog(CIHMcalculetteDlg::IDD, pParent){
    //{{AFX_DATA_INIT(CIHMcalculetteDlg)
    m_operation = _T("");
    ..... ;
    //}}AFX_DATA_INIT
}

void CIHMcalculetteDlg::DoDataExchange(CDataExchange* pDX){
    CDialog::DoDataExchange(pDX);
    //{{AFX_DATA_MAP(CIHMcalculetteDlg)
    DDX_Text(pDX, IDC_SAISIE, m_saisie);
    ..... ;
    //}}AFX_DATA_MAP
}

BEGIN_MESSAGE_MAP(CIHMcalculetteDlg, CDialog)
    //{{AFX_MSG_MAP(CIHMcalculetteDlg)
    ON_BN_CLICKED(IDC_CHIFFRE_1, OnChiffre1)
    ON_BN_CLICKED(IDC_OP_PLUS, OnOpPlus)
    ..... ;
    //}}AFX_MSG_MAP
END_MESSAGE_MAP()

void CIHMcalculetteDlg::OnOpPlus() {
    maCalculette.operation('+');
    afficher();
}

void CIHMcalculetteDlg::OnChiffre1() {
    saisieVal('1');
}

void CIHMcalculetteDlg::saisieVal(char value) {
    maCalculette.set_saisie(value);
    afficher();
}

void CIHMcalculetteDlg::afficher() {
    m_saisieB = maCalculette.get_stringValB().c_str();
    ..... ;
    // Mise à jour de l'affichage
    UpdateData(false);
}

```

**LE FICHIER CALCULETTE.H**

```

#ifndef MA_CALCULETTE_H
#define MA_CALCULETTE_H

#include <string>
using namespace std;

class Calculette {
public:
    // ----- INTERFACE Fonctions Publiques -----
    Calculette(); // Constructeur
    virtual ~Calculette(); // Destructeur

```

```
// Initialise la calculette
void clear();
// efface le dernier caractère de "saisie"
void effacerCar();
// Requête pour traiter le caractère '='
void egal();
// Retourne l'opérande valA sous la forme d'une chaîne de caractères
string get_stringValA();
// Retourne l'opérande valB sous la forme d'une chaîne de caractères
string get_stringValB();
// Retourne valResultat sous la forme d'une chaîne de caractères
string get_stringValResultat();
// Requette pour traiter le caractère opérateur : '+', '-', '*', '/'
// le caractère est passé en paramètre
void operation(char value);
// Retourne la valeur de l'opérateur
const char& get_operateur() const;
// Retourne resultat sous la forme d'une chaîne de "saisie"
const string& get_saisie() const;
// Requette pour traiter un caractère comme opérande
void set_saisie(char value);

private:
// ----- DONNES Privées -----
// Etat de la calculette
char etat;
// Action à réaliser dans les Etats de la calculette
char action;
// Valeur du caractère saisie
char carSaisie;
// Valeur de l'opérateur
char operateur;
// Etat du point
char point;
// Chaîne de caractère de la Saisie
string saisie;
// Transition pour faire évoluer les états
char transition;
// Valeur de l'opérande A
double valA;
// Valeur de l'opérande B
double valB;
// Valeur du Résultat de l'opération
double valResultat;

// ----- FONCTIONS Privées -----
// Ajoute un caractère à la chaîne "saisie"
void ajouterCar(char value);
// Ajoute le "." à la chaîne "saisie"
void ajouterPoint();
// Fait le calcul valResultat <- valA OPERATION valB
// Met à jour valResultat, valA, et saisie
void calculer();
// Converti un double en une string
string convertirValeur(double value);
// Converti une string en un double
double convertirString(const string &chaîne);
// Fait évoluer les états de la calculette
void gererEtat();
// Vide 'saisie' et ajoute 1 caractère
void initSaisie(char value);
// Fait les traitements en fonction des états de la calculette
void traiterEtat();
```

```
// ----- CONSTANTES - Enumérés Privées -----  
// Définition des Etats de la calculette  
enum ETAT {  
    SAISIE_A='A',  
    SAISIE_B='B',  
    OPERATION='O',  
    EGAL='E',  
};  
// Définition des transitions entre les Etats de la calculette  
enum TRANSITION {  
    EVENEMENT_OPERATION='p',  
    EVENEMENT_EGAL='e',  
    EVENEMENT_VALEUR='v',  
};  
// Définition des Actions pour un Etat  
enum ACTION {  
    ENTRY='Y',  
    DO='D',  
    ON='O',  
    EXIT='X',  
    SORTIE_EVE_OPERATION='P',  
    SORTIE_EVE_EGAL='E',  
};  
};  
#endif // MA_CALCULETTE_H
```