

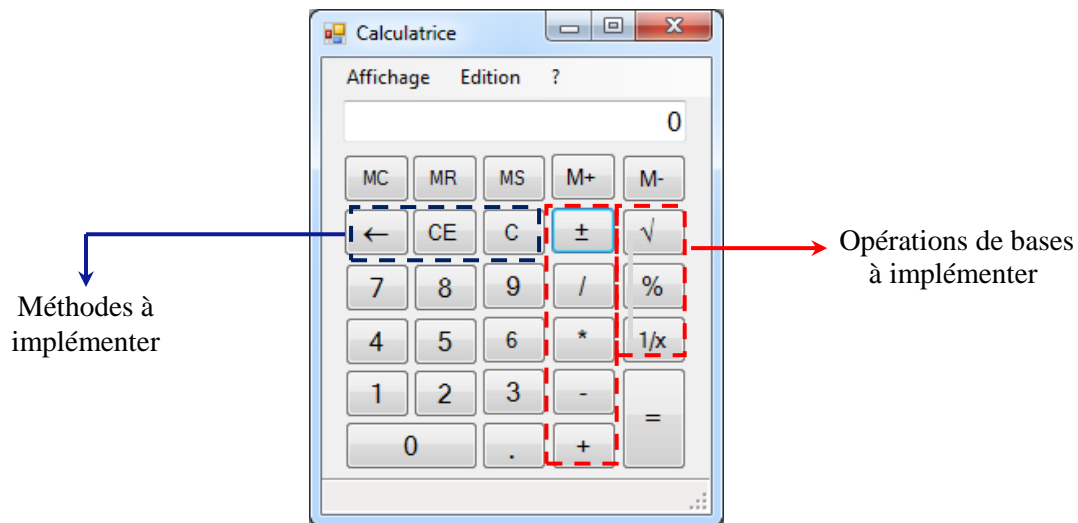
## Programmation Objets avec C# – TP1 –

### I/ Objectif du TP

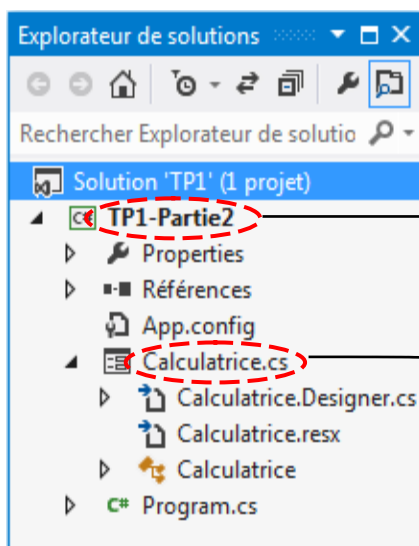
Le but de ce TP est de faire une révision des concepts de base la programmation par objets avec C# à travers l'implémentation d'une **application Console** pour sa première partie et d'une **application Windows Forms** pour sa deuxième partie.

### Partie II/

Réaliser une application Windows Forms pour simuler le fonctionnement d'une calculatrice standard.



### 1- Exemple : Réalisation d'une calculatrice avec Windows Forms



→ 1. **Projet TP1-Partie2** : relatif à la création d'une application Windows Forms.

→ 2. **Classe Calculatrice.cs** : contenant l'interface, le fichier designer et le code source de la calculatrice.

## Programmation Objets avec C# – TP1 –

### 2. Projet TP1 Partie2 :

#### 2.1. Classe Calculatrice :

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace TP1_Partie2
{
    public partial class Calculatrice : Form
    {
        string Operande, Operateur;

        // _____//
        public Calculatrice()
        {
            InitializeComponent();
        }
        // _____//
        private void Button_Click(object sender, EventArgs e)
        {
            if (Resultat.Text == "0") { Resultat.Text = ((Button)sender).Text; }
            else { Resultat.Text = Resultat.Text + ((Button)sender).Text; }
        }
        // _____//
        private void C_Click(object sender, EventArgs e)
        {
            Resultat.Text = "0";
        }
        // _____//
        private void Operation_Click(object sender, EventArgs e)
        {
            Operande = Resultat.Text; Resultat.Text = "";
            Operateur = ((Button)sender).Text;
        }
        // _____//
        private void Egal_Click(object sender, EventArgs e)
        {

```

## Programmation Objets avec C# – TP1 –

```
switch (Operateur)
{
    case "+": { Resultat.Text = (double.Parse(Operande) +
        double.Parse(Resultat.Text)).ToString(); break; }
    case "-": { Resultat.Text = (double.Parse(Operande) -
        double.Parse(Resultat.Text)).ToString(); break; }
    case "*": { Resultat.Text = (double.Parse(Operande) *
        double.Parse(Resultat.Text)).ToString(); break; }
    case "/": {
        if (Resultat.Text != "0") { Resultat.Text = (double.Parse(Operande)
            / double.Parse(Resultat.Text)).ToString(); break; }
        else { MessageBox.Show("Division impossible !"); break; } }
    case "±": { Resultat.Text = ((-1) * double.Parse(Operande)).ToString();
        break; }
    case "√": {
        if (double.Parse(Operande) >= 0) { Resultat.Text =
            (Math.Sqrt(double.Parse(Operande))).ToString(); break; }
        else { MessageBox.Show("Impossible de calculer la racine d'un nombre
            négatif !"); break; } }
    case "%": { Resultat.Text = (((0.001) * double.Parse(Operande)).ToString());
        break; }
    case "1/x": {
        if (Resultat.Text != "0") { Resultat.Text =
            ((1/(double.Parse(Operande))).ToString()); break; }
        else { MessageBox.Show("Impossible de fractionner sur zéro !"); break; }
    }
}
// _____ //
```

**Remarque :** Tous les boutons chiffre (0 à 9) doivent avoir la méthode `Button_Click` pour l'évènement `Click` et tous les boutons opération (+, -, \*, /, ..) doivent avoir la méthode `Operation_Click` pour l'évènement `Click`. Modifiez pour cela les évènements des boutons dans la barre d'outils Propriétés.

### 2.2. Travail à faire :

Complétez la programmation de la calculatrice initiale en corrigeant les petites imperfections qui peuvent surgir et en y intégrant les fonctions restantes. Modifiez la zone `TextBox` pour qu'elle devienne une zone multi-ligne. Inspirez-vous pour cela de la calculatrice scientifique de Windows.