

TP 1: 1^{ère} APPLICATION AVEC VISUAL STUDIO (C#)

OBJECTIFS :

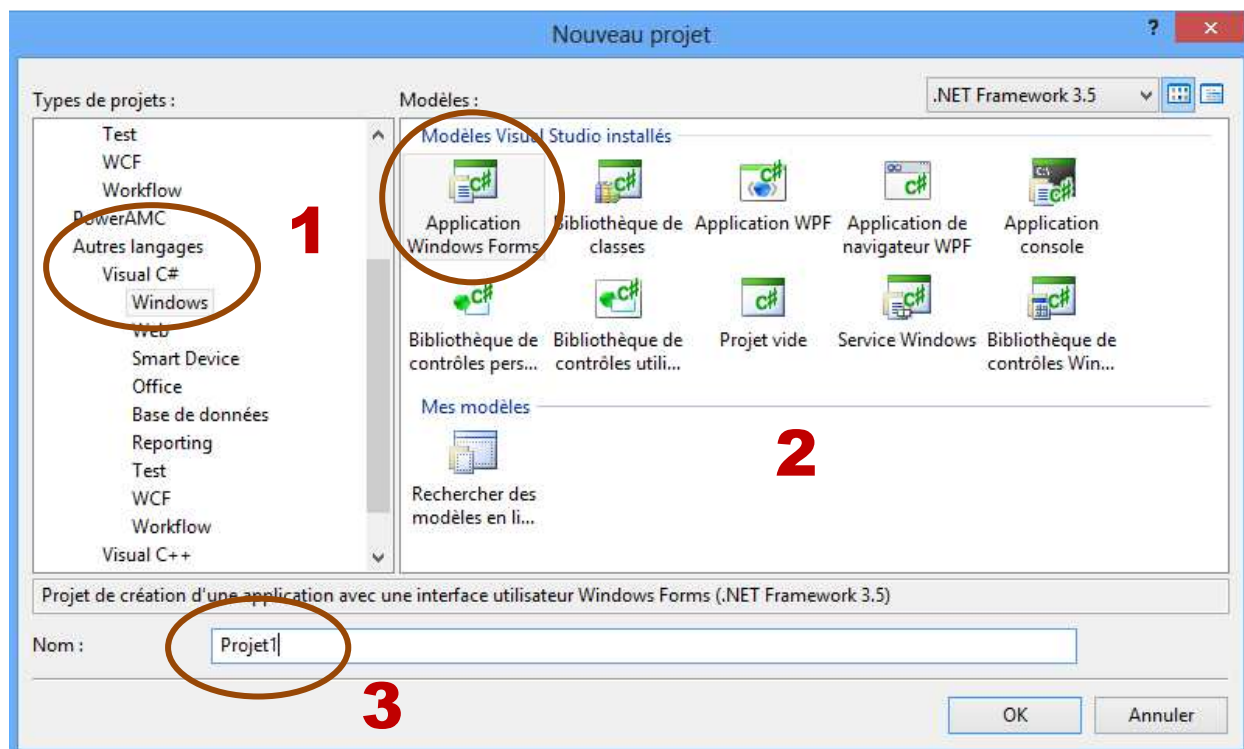
Prise en main de l'environnement de programmation Visual Studio 2008.

A. Création et lancement d'un projet:

La version .net de Visual Studio fonctionne suivant les notions de "**Projet**" et "**Solution**", avec le langage C# :

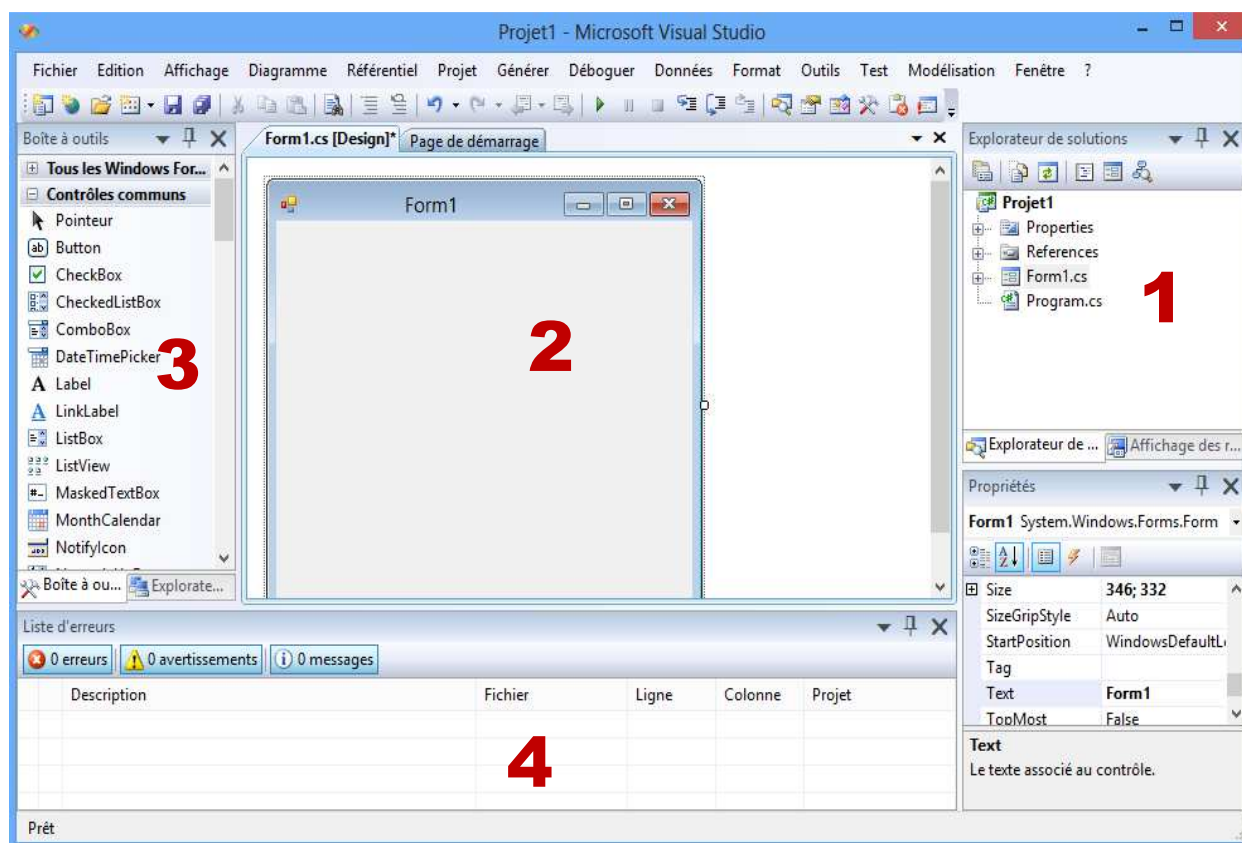
- **Projet**: regroupe des informations techniques relatives à la programmation. Il est représenté par le fichier "**Nomprojet.csproj**" dans le quel sera stockés le nom des fichiers présents, les composants et les bibliothèques utilisés, etc.
- **Solution** : c'est l'environnement de développement du programmeur, c'est un espace permettant de regrouper un ou plusieurs projets dont les détails sont sauvegardés dans le fichier "**NomSolution.sln**".

- Lancer:  Microsoft Visual Studio 2008
- Ensuite, lancer un nouveau projet : **Fichier -> Nouveau -> Projet**. Vous arrivez alors devant la boîte de dialogue suivante:

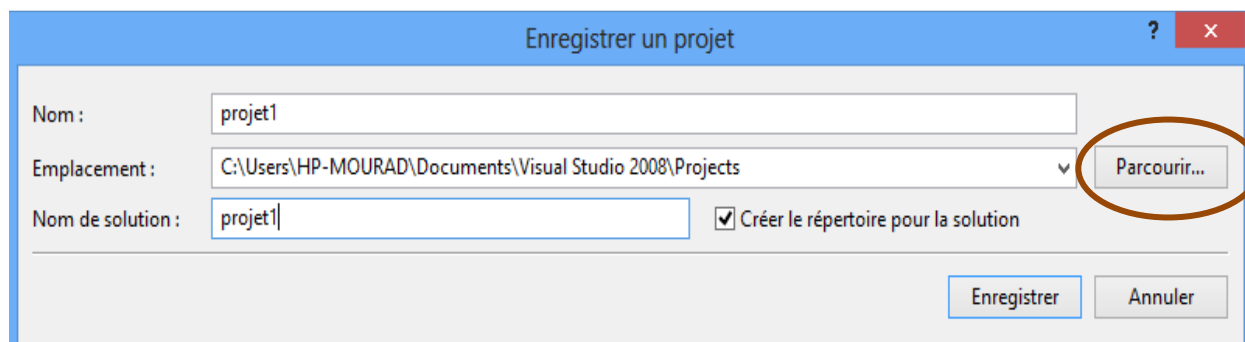


1. **Types de Projets** : Regroupe l'ensemble des projets (classés par langage de programmation) disponibles sous Visual Studio 2008. Choisissez **Visual C#**.
2. **Modèles** : ou bien groupes de projets pour un types de projets (pour le langage choisit). Choisissez **Application Windows Forms**
3. C'est le nom du projet, changez-le avec **Projet1**.

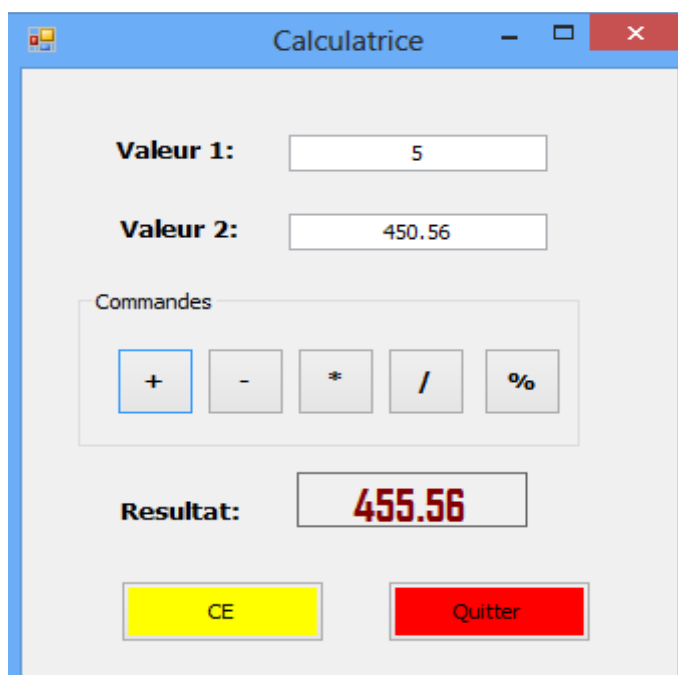
- Appuyez sur OK pour lancer le projet. L'interface générale se présente sous cette forme :



- L'interface de développement de Visual Studio en dehors de ses menus et barres d'outils, possède quatre zones :
 1. C'est l'explorateur de solution, Elle regroupe tout ce qui fait l'ensemble du programme c'est-à-dire le(s) projet(s), les Contrôles Utilisateur, Les Ressources, Les Classes, les Bases de Données, les Feuilles (Form), les Fichiers de Configuration, ...
 2. C'est la zone d'affichage des fichiers ouverts : sources, aide, etc ...
 3. C'est la boîte à outils, ou se trouve les différents composants graphiques à mettre dans les interfaces.
 4. C'est la sortie, fenêtre d'affichage des résultats de compilation, de débogage et de recherches.
- Enregistrer le projet, vérifier l'emplacement (par défaut la solution porte le même nom du projet et le répertoire sera créer suivant le nom de la solution)



B. Première Application:



Contrôles nécessaires :

<u>Type du contrôle</u>	<u>Nom de la propriété</u>	<u>Valeur de la propriété</u>
Form	Name	FrmCalculatrice
	Text	Calculatrice
TextBox	Name	TxtV1
	TextAlign	Centre
TextBox	Name	TxtV2
	TextAlign	Centre
Label	Name	LblV1
	Text	Valeur 1:
	Font	Gras , taille 10
Label	Name	LblV2
	Text	Valeur 2:
	Font	Gras , taille 10
Button	Name	BtnPlus
	Text	+
Button	Name	BtnMoins
	Text	-
Button	Name	BtnFois
	Text	*
Button	Name	BtnSur
	Text	/
Button	Name	BtnMod
	Text	%

Button	Name	BtnCE
	Text	CE
Button	Name	BtnQuitter
	Text	Quitter
Label	Name	LblResultat
	Text	Resultat:
	Font	Gras , taille 10
Label	Name	LblRes
	Text	
	Font	Gras , taille 14
	BorderStyle	FixedSingle
	Autosize	False
GroupBox	Name	GrBCmd
	Text	Commandes

Le code relatif aux différentes opérations:

```
private void BtnPlus_Click(object sender, EventArgs e)
{
    float s;
    s = float.Parse(TxtV1.Text) + float.Parse(TxtV2.Text);
    LblRes.Text = s.ToString();
}
private void BtnMoins_Click(object sender, EventArgs e)
{
    LblRes.Text=(float.Parse(TxtV1.Text)-float.Parse(TxtV2.Text)).ToString();
}
private void BtnFois_Click(object sender, EventArgs e)
{
    ....
}
.....
private void BtnCe_Click(object sender, EventArgs e)
{
    TxtV1.Text = "0";
    TxtV2.Text = "0";
    LblRes.Text = "0";
}

private void BtnQuitter_Click(object sender, EventArgs e)
{
    this.Close();
}
```

Amélioration avec MessageBox:

▪ Contrôle des champs avec TryParse:

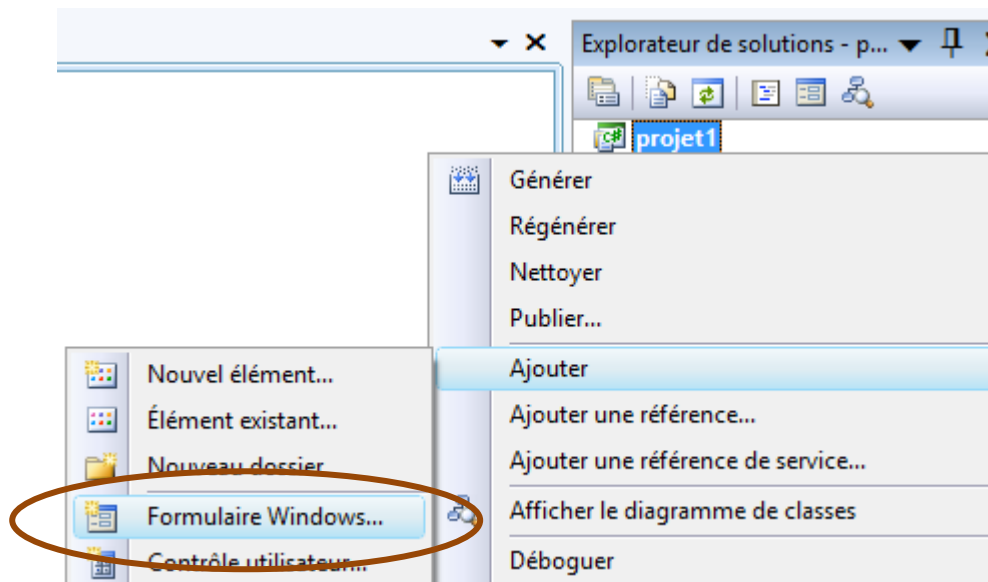
```
float s,x1,x2 ;
if (float.TryParse(TxtV1.Text, out x1) && float.TryParse(TxtV2.Text, out x2))
{
    s = x1 + x2;
    LblRes.Text = s.ToString();
}
else
    MessageBox.Show("Il faut donner deux réels !", "Erreur",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
```

- Message de confirmation:

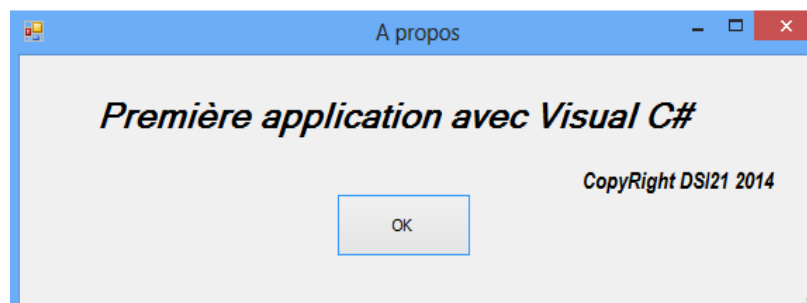
```
DialogResult x = MessageBox.Show("Voulez vous quitter?",  
"Confirmation", MessageBoxButtons.YesNo);  
  
if (x == DialogResult.Yes)  
    Application.Exit();
```

C. Autres interfaces:

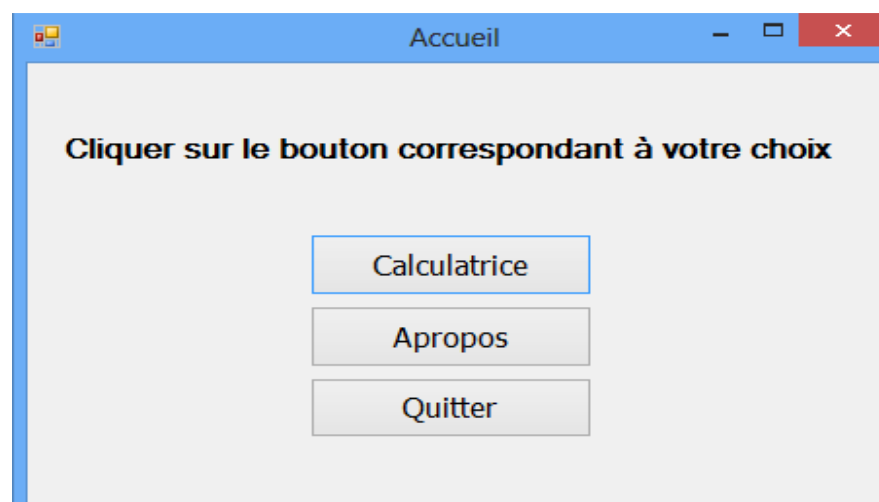
Avec un click droit sur le nom du projet,



- Ajouter la forme *frmApropos* suivante:



- Ajouter la forme *frmAccueil* suivante:



Forme frmAccueil :

```
private void BtnCalcul_Click(object sender, EventArgs e)
{
    frmCalculatrice cl = new frmCalculatrice();
    cl.Show();
    this.Hide();
}
```

```
private void BtnApropos_Click(object sender, EventArgs e)
{
    frmApropos ap = new frmApropos();
    ap.ShowDialog();
}
```

```
private void BtnQuitter_Click(object sender, EventArgs e)
{
    Application.Exit();
}
```

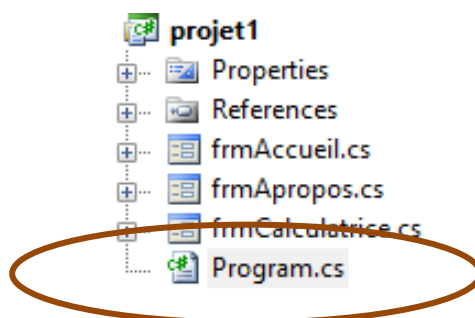
Forme frmCalculatrice:

Ajouter un bouton **BtnRetour** permettant de retourner à la forme d'accueil:

```
private void BTretour_Click(object sender, EventArgs e)
{
    frmAccueil acc = new frmAccueil ();
    acc.Show();
    this.Close();
}
```

Changement de la forme de démarrage:

- A partir de *l'explorateur de solutions*, ouvrir le fichier **Program.cs**



- Dans le code changer la forme de démarrage par celle **frmAccueil**:

```
static class Program
{
    /// <summary>
    /// Point d'entrée principal de l'application.
    /// </summary>
    [STAThread]
    static void Main()
    {
        Application.EnableVisualStyles();
        Application.SetCompatibleTextRenderingDefault(false);
        Application.Run(new frmAccueil());
    }
}
```