

TP Informatique 3 – 8h

Les chaînes

Le but du TP est également de continuer la prise en main de Visual studio.

Dans un premier temps, le but est de compléter le programme de résolution d'équation pour bien gérer le cas des solutions réelles ou complexes.

L'art de nommer des variables

Lorsqu'on déclare une variable, on lui choisit un nom. Ce nom doit permettre de facilement comprendre le rôle de cette variable.

Exemple :

Double MontantTTC ;

Double PrixUnitaire ;

Int quantiteArticle1 ;

String nomClientEnAttenteEnregistrement ;

Int Var

Vous remarquez que :

- on écrit en minuscule, sauf quand on combine plusieurs mots (majuscule au début de chaque mot)
- on utilise des caractères autorisés : lettres, chiffres et certains caractères spéciaux dont '_'.
- Pas de '+', '-', '*' etc...

Certains programmeurs utilisent le '_' comme séparateur ... nous éviterons

Int Valeur_du_Stock ; // A EVITER

Les noms tout en majuscule sont réservés pour désigner des variables globales .

double TOTAL_HT ; // A EVITER

Quand vous créez des composants graphiques, ils ont un nom par défaut : Button1, Button2,...

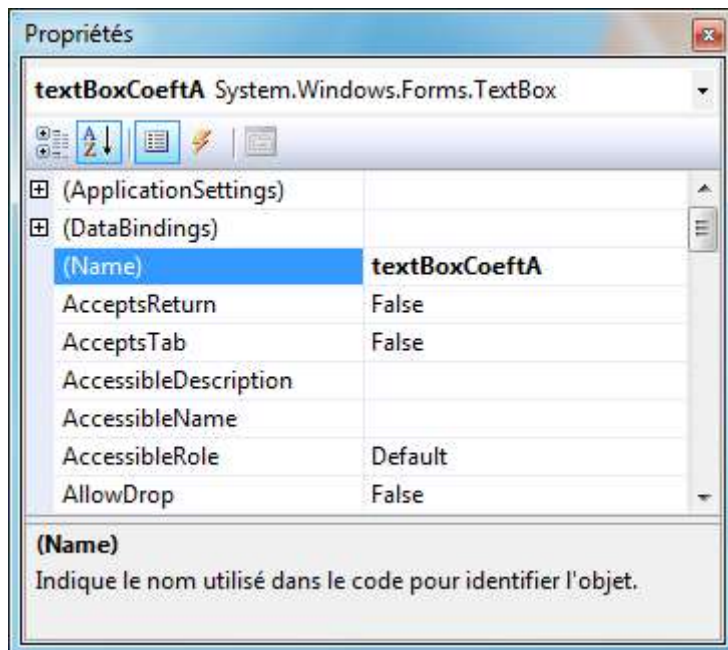
Vous pouvez renommer un objet graphique en allant dans ses propriétés.

Quand vous changez le nom, TOUT le programme est mis à jour automatiquement !!

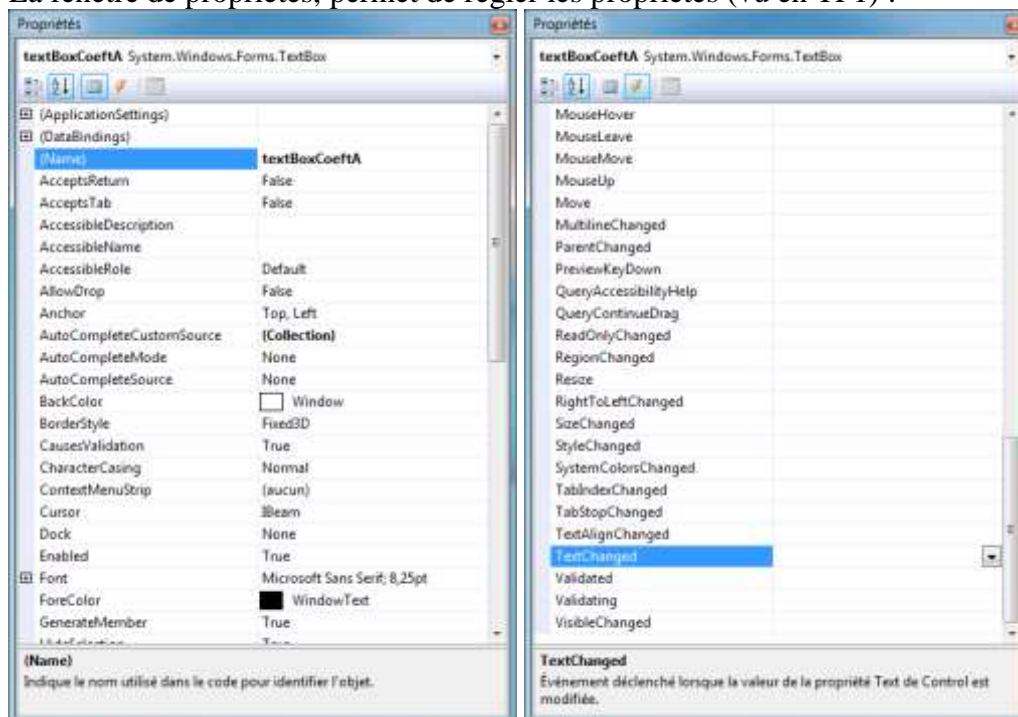
Exemple : ici, on a renommé la textBox1 en textBoxCoeftA. On sait de suite qu'il s'agit d'une textBox et on sait qu'il s'agit du CoeftA de notre équation du second degré.

On a renommé ainsi, les variables qu'on manipule :

```
private System.Windows.Forms.TextBox textBoxCoeftA;
private System.Windows.Forms.Button buttonResoudre;
private System.Windows.Forms.TextBox textBoxCoeftB;
private System.Windows.Forms.TextBox textBoxCoeftC;
private System.Windows.Forms.Label label1;
private System.Windows.Forms.Label label2;
private System.Windows.Forms.Label label3;
private System.Windows.Forms.Label label4;
private System.Windows.Forms.Label label5;
private System.Windows.Forms.Label label6;
private System.Windows.Forms.RadioButton radioButtonReel;
private System.Windows.Forms.RadioButton radioButtonComplexe;
private System.Windows.Forms.Label labelSolution1;
private System.Windows.Forms.Label labelSolution2;
```



La fenêtre de propriétés, permet de régler les propriétés (vu en TP1) :



Mais aussi les gestions des événements. Pour ajouter une fonction , il suffit de double-cliquer sur l'événement, une fonction sera créée.

Pour supprimée une fonction, on fait un click droit et on réinitialise.

RadioButton

Ce composant permet de faire un choix. A l'exécution, si on sélectionne un radiobutton, les autres seront automatiquement invalidés. Pour savoir si , un radiobutton est valide ou non, on test sa propriété 'checked' .

Exemple :

```
if (radioButtonComplexe.Checked == true) //si on a sélectionné en complexe
```

Les commentaires

Il paraît que c'est tout un art de bien commenter un programme.

Il ne faut pas trop en mettre.

Il est inutile de mettre des commentaires évidents :

```
Int val=2 ; // on crée une variable A entière et on met 2 dedans.
```

Il faut chercher à aider le programmeur qui devra reprendre votre programme pour le corriger ou le modifier.

qq techniques :

Ajout d'un commentaire en fin de ligne avec //

```
this.labelSolution2.Text = ""; // on efface les anciens résultats
```

Une ou des lignes de commentaires avec //

```
// -----  
//la condition calcul réel ou complexe ne concerne que le cas où delta==0  
//-----
```

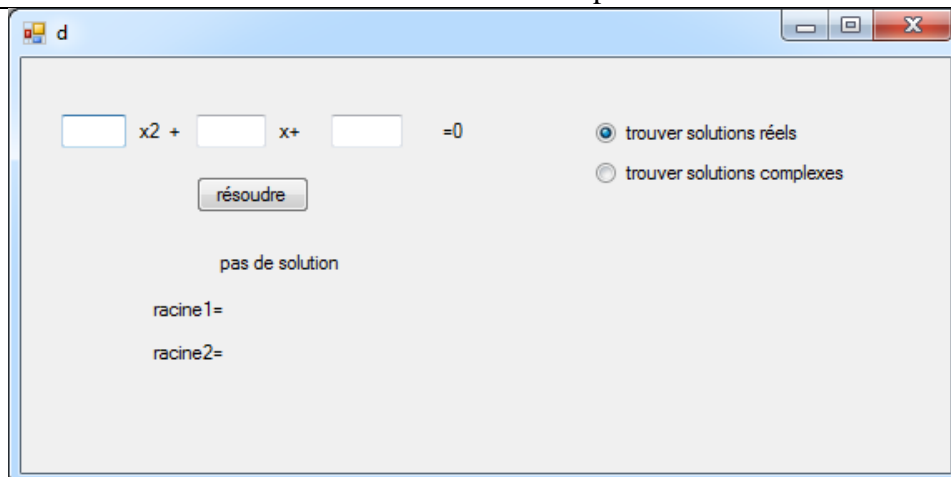
Créer un bloc de commentaires avec /* */

```
/* debut commentaire  
-----  
la condition calcul réel ou complexe ne concerne que le cas où delta==0  
-----  
Fin commentaire */
```

Equation (suite)

Manipulation :

1°)Reprendre le programme de résolution d'équation. Compléter le programme fourni .
On veut choisir une résolution en réel ou complexe.



```

private void button1_Click(object sender, EventArgs e)
{
    double a, b, c; // les coefts de l'équation
    double r1, r2; // les partie réelles des solutions
    double partieReel, partieImaginaire; // pour calcul du complexe
    double delta; // le discriminant
    try
    {
        // conversion en réel

        // calcul delta

        if (delta == 0)
        {
            this.labelMessage.Text = "UNE solution, nombre réel";
            // la racine unique
            // on efface les anciens résultats
        }
        if (delta > 0)
        {
            this.labelMessage.Text = "DEUX solutions , nombres réels ";
        }

        // -----
        // la condition calcul réel ou complexe ne concerne que le cas où delta==0
        //-----

        if (delta < 0)
        {
            if (radioButtonReel.Checked == true) // si on a sélectionné en réel
            {
                this.labelMessage.Text = "pas de solution";
                this.labelSolution1.Text = ""; // on efface les anciens résultats
                this.labelSolution2.Text = ""; // on efface les anciens résultats
            }
            if (radioButtonComplexe.Checked == true) // si on a sélectionné en
complexe
            {
                this.labelMessage.Text = "DEUX solutions complexes";
                // partie réel 1
                // imaginaire avec sqrt( -delta) car delta est négatif!!
                this.labelSolution1.Text = partieReel.ToString() + " +i " +
partieImaginaire.ToString(); // affiche en ..+ i.. (exmple : 2+i5,2)
                this.labelSolution2.Text = partieReel.ToString() + " -i " +
partieImaginaire.ToString();

            }
        }
    }
    catch (Exception err)
    {
        MessageBox.Show("erreur " + err.Message); // affiche message de l'exception
    }
}

```

2°) Modifier le programme : on veut être sûr que le coef A n'est pas nul.

Contrôle d'une adresse courriel.

Une adresse mail est une chaîne de caractères que a la structure suivante :

pseudo@domaine

Pour manipuler les chaînes en C# on utilise le string.

Ex :

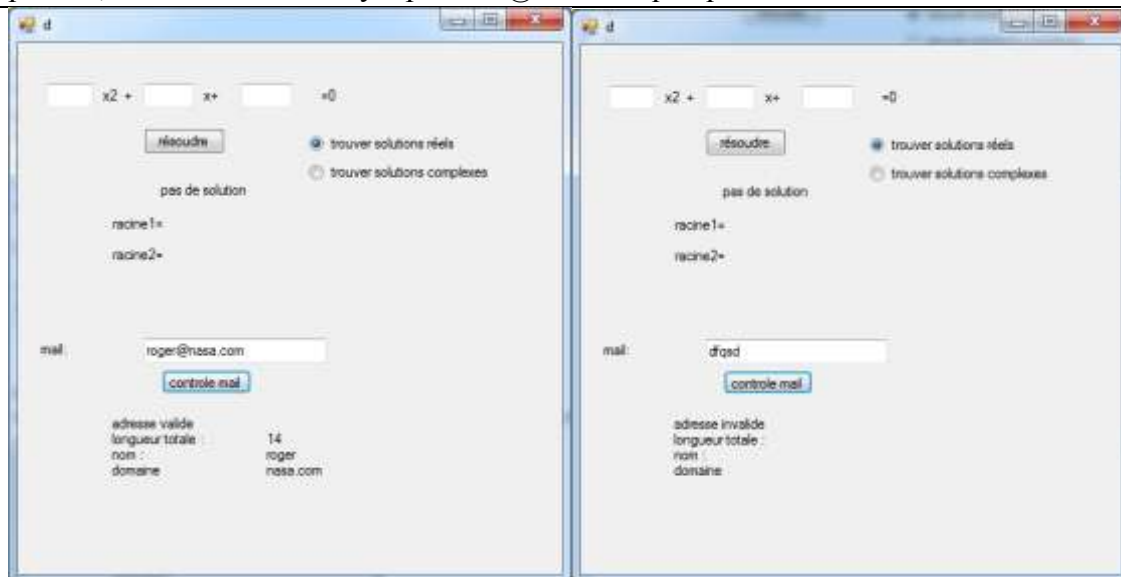
```
String s = this.textBox1.Text;  
String message = "erreur ";
```

Cette classe comporte des methodes :

```
int l = s.Length;  
int indexArobas = s.IndexOf('@');  
String nom = s.Substring(0, indexArobas);
```

Manipulation :

3°) On compléter notre appli pour pouvoir saisir une adresse mail et afficher : sa longueur, le pseudo, le domaine. Si il n'y a pas de '@' on indique que l'adresse est invalide



Remarque :

Notre contrôle est assez incomplet .Il existe une classe en C# qui prend cela en charge : Regex comme « Regular Expression ».

Nous n'étudierons pas en détail cette classe, mais voici un exemple d'utilisation pour contrôler une adresse mail.

```
Regex Email=new Regex("^([a-zA-Z0-9_.-]+@[1][a-zA-Z0-9_.-]{2,}\\.[a-zA-Z0-9_.-]{2,4}$");
```

- `^[a-zA-Z0-9_.-]+` signifie que la chaine email doit absolument commencer par au moins une lettre (majuscule ou minuscule) ou un chiffre ou les caractères "-", "_", "." (`paul@hebergeur.com`)

- `@{1}` signifie que la chaine email doit ensuite contenir une et une seule fois le caractère "@" (`paul@hebergeur.com`)

- `[a-zA-Z0-9_-]{2,}` Signifie que la chaîne doit ensuite posséder au moins deux caractères (paul@[hebergeur.com](#))

- `\\.` signifie que la chaîne doit contenir un point entre l'hebergeur et le domaine (paul@hebergeur.[com](#))

- `[a-zA-Z0-9_-]{2,}$` Signifie que le nom de domaine doit posséder entre 2 et 4 caractères (paul@hebergeur.[com](#))

Ensuite la méthode `IsMatch()` permet de contrôler une chaîne de caractère.
Voici un exemple :

```
private void buttonRegExClick(object sender, EventArgs e)
{
    Regex Email=new Regex(@"^[a-zA-Z0-9_-]+@{1}[a-zA-Z0-9_-]{2,}\\.[a-zA-Z0-9_-]{2,4}$");

    if(Email.IsMatch(textBoxAdresseMail.Text))
    {
        this.labelRegEx.Text="adresse OK";
    }
    else
    {
        this.labelRegEx.Text="adresse INVALIDE";
    }
}
```

Manipulation :
4°)Ajouter un bouton de contrôle pour tester avec Regex.

URL.

Qu'est-ce qu'une URL?

Une **URL** (*Uniform Resource Locator*) est un format de nommage universel pour désigner une ressource sur Internet. Il s'agit d'une chaîne de caractères [ASCII](#) imprimables qui se décompose en cinq parties :

- **Le nom du protocole** : c'est-à-dire en quelque sorte le langage utilisé pour communiquer sur le réseau. Le protocole le plus largement utilisé est le [protocole HTTP](#) (*HyperText Transfer Protocol*), le protocole permettant d'échanger des pages Web au format [HTML](#). De nombreux autres protocoles sont toutefois utilisables (FTP,News,Mailto,Gopher,...)
- **Identifiant et mot de passe** : permet de spécifier les paramètres d'accès à un serveur sécurisé. Cette option est déconseillée car le mot de passe est visible dans l'URL
- **Le nom du serveur** : Il s'agit d'un [nom de domaine](#) de l'ordinateur hébergeant la ressource demandée. Notez qu'il est possible d'utiliser l'[adresse IP](#) du serveur, ce qui rend par contre l'URL moins lisible.
- **Le numéro de port** : il s'agit d'un numéro associé à un service permettant au serveur de savoir quel type de ressource est demandée. Le port associé par défaut au protocole est le port numéro 80. Ainsi, lorsque le service Web du serveur est associé au numéro de port 80, le numéro de port est facultatif
- **Le chemin d'accès à la ressource** : Cette dernière partie permet au serveur de connaître l'emplacement auquel la ressource est située, c'est-à-dire de manière générale l'emplacement (répertoire) et le nom du fichier demandé

Une URL a donc la structure suivante :

Protocole	Mot de passe (facultatif)	Nom du serveur	Port(facultatif si 80)	Chemin
http://	user:password@	www.commentcamarche.net	:80	/glossair/glossair.php3

Le nom de fichier dans l'URL peut être suivi d'un point d'interrogation puis de données au format ASCII, il s'agit de données supplémentaires envoyées en paramètre d'une application sur le serveur (un script [CGI](#) par exemple). L'URL ressemblera alors à une chaîne de caractères comme celle-ci :

<http://www.commentcamarche.net/forum/index.php3?cat=1&page=2>

5°) Ecrire une application qui à partir d'une URL saisie au clavier :

Affiche dans différents labels : le protocole, le mot de passe, le serveur, le port et le chemin.

Pour les tests, vous avez intérêt à définir une textBox qui contienne par défaut une URL valide (pour éviter de saisir à chaque fois !).

Critères évaluation :

Respect consignes, sauvegarde (clé), autonomie : 3

Nommage : 2

Commentaires : 1

Fonctionnement résolution équation : 3

Cas coeft a nul : 1

Courriel : 4

Regex : 2

URL : 4