

Feuille de TP n°5 :

Programmation événementielle avec wxWidgets

- Ajout de dynamique aux objets graphiques
- Modélisation de la dynamique avec UML

Objectif pédagogique

Cette séance de TP a pour but de vous familiariser avec le framework wxWidgets (hiérarchie de classes et méthodes) permettant de créer un programme avec interface graphique.

TRAVAUX A RENDRE

...à déposer **individuellement et uniquement(*)** sur l'espace Travaux à rendre de ce module, avant lundi 13/05/2019, 23h59, dans une archive .zip à votre nom et groupe de TP

(*) Ne jamais envoyer les travaux par mail aux enseignants, toujours les déposer sur les espaces prévus à cet effet.

1.- Réponses aux questions suivantes de la feuille de td n°6 :

- **Exercice A.-**
Questions 1 et 2
- **Exercice B.-**
Questions 2, 3, 4, 5, 6

2.- Sources des programmes deux et trois.

RESSOURCES A VOTRE DISPOSITION POUR REALISER LE TP

- ~ Cette feuille de TP n°5.
- ~ documentation wxWidgets : <http://docs.wxwidgets.org/3.0/>

DIRECTIVES PARTICULIERES A CETTE SEANCE

Dans votre répertoire de travail habituel, **et dans le dossier M2105-IHMs**,

- créez un répertoire nommé **tp5**

A.- deux.exe : Confirmation de fermeture

- (a) Dans le répertoire **tp5**, **créez** un projet deux, **copiez** les codes sources du programme zero fourni au TP n°4.
- (b) Modifier/Compléter les codes sources selon la progression proposée sur la feuille de td n°6.

B.- trois.exe : Additionneur simplifié (Deuxième étape)

- (c) Dans le répertoire **tp5**, **créez** un projet trois, **copiez** les codes sources du programme **un** élaboré lors de votre TP n°4.
- (d) Si nécessaire, normalisez les noms des variables de votre programme afin qu'ils soient cohérents avec leur rôle dans le programme (cf. Annexe 2 du TD n°6)
- (e) Modifier/Compléter les codes sources selon la progression proposée sur la feuille de td n°6.

Indications

1) Pour transformer un texte (type wxString) en nombre entier

Utiliser la méthode `ToLong()` de la classe `wxString`, de la manière suivante :

```
wxTextCtrl *text_X;
... en supposant que la zone de saisie text_X contient une valeur à tester...

wxString texte;
texte = text_X ->GetValue(); // cf méthode GetValue() de la classe wxTextCtrl

if (texte.ToLong(&x, 10) == true)
{ // actions à faire dans le cas où la variable x de type long a été correctement
  // initialisée à partir du contenu de texte
}
else
{ // actions à faire dans le cas où la variable x de type long n'a pas été correctement
  // initialisée à partir du contenu de texte
}

/*
la méthode ToLong() tente de convertir le contenu textuel de la variable texte
- en Nombre entier (long)
- codé en base 10
- qu'il range dans la variable x.
Si l'opération réussit, retourne TRUE, sinon, retourne FALSE
*/
```

Consultez la documentation en ligne qui vous a été fournie pour comparer l'exemple utilisé avec l'entête / la signature de la méthode.....

2) Pour affecter une variable de type wxString avec une valeur ou une expression

Si la valeur ou expression est de type `wxString`, utiliser l'opérateur d'affectation `=` habituel, ou bien l'opérateur `<<` de manière analogue à la redirection sur `cout`, comme indiqué ci-dessous :

```
wxString texte;
texte << wxT("Bonjour") ; // contient la chaîne wxString équivalente à "Bonjour"
texte = wxT("Bonjour") ; // idem
int rep = wxMessageBox(texte) ;
```

Si la valeur ou expression n'est pas de type `String`, utiliser l'opérateur `<<` de manière analogue à la redirection sur `cout`, comme indiqué ci-dessous :

```
int nombre = 5;
texte = wxT("") ; // réinitialisation de texte avec la chaîne vide
texte << (10 + nombre) ; // texte contient la chaîne wxString équivalente à "15"
int rep = wxMessageBox(texte) ;
```

Pour vous en persuader, utiliser `wxMessageBox` pour afficher les valeurs intermédiaires de la variable `texte`.

3) Pour concaténer 2 chaînes wxString

Utiliser l'opérateur `+`, comme indiqué ci-dessous :

```
wxString texte, partie1, partie2; // 3 chaînes de caractères
texte = wxT("");
partie1 = wxT("Bonjour ");
partie2 = wxT("monde");

texte = partie1 + partie2 ;
int rep = wxMessageBox(texte) ;
```