TD MOO : C++ / UML Séance 4/6

TD 4 – Utiliser des classes existantes, IHM

Préparation individuelle :

maitrise pointeur/héritage,

et Exercice 2

Exercice 1 : Gui with Qt Le but de cet exercice est de faire une interface graphique pour l'exercice 1 du TD0.

1-a Faire un projet QtCreator avec IHM. Voici les étapes à suivre :

Au lancement de QtCreator, faire un nouveau projet et choisir « Qt Gui Application » ou « Qt Widgets Application » pour QtCreator 4.0 et cliquer sur « Choisir » en bas de cette fenêtre :



Donner ensuite le nom de votre projet et le chemin, terminer par « Next »

Targets This wizard generates a Qt4 GUI application project. The application derives by default from QApplication and includes an empty widget. Details Summary	
largets Details Summary	
Details Summary	
Summary	
Name: TD1	
Create in: E-\tmn	
	owse
Name: TD 1	

QtCreator demande ensuite de choisir les configurations de construction de votre projet. Vous pouvez laisser les valeurs par défauts. Cliquer sur « Next ».

Location	Target Setup	targets for project TD1	
Targets Details Summary	V I Desktop		 Detai
	Create build configurations: Mi	E:\tmp\TD1-build-desktop-Qt_4_8_2Qt4_8_2Debug	Browse.
	Vt 4.8.2 (Qt4.8.2) release	E:\tmp\TD1-build-desktop-Qt_4_8_2Qt4_8_2Release	Browse.

Ensuite vous pouvez choisir le nom de votre fenêtre principale. Laisser les autres paramètres aux valeurs par défaut :

Location	Class Inform	mation	
Torgeta	Specify basic inf	formation about the classes for which you want to generate skeleton source code files.	
Targets Dotaile		, -	
Summary	Class name:	MainWindow	
Summary	Base class:	QMainWindow	
			_
	Header file:	mainwindow.h	
	Source file:	mainwindow.cpp	
	Generate form:		
	Form file:	mainwindow.ui	

Enfin, choisir un outil de gestion de version et de travail collaboratif. Dans le cadre de ces TD il n'y en a pas, vous pouvez cliquer sur « Finish »

🗖 Qt Gui Ap	pplication		Lating Transf	- 1
	Project Management			
Location				
Targets	Add as a subproject to project:	<none></none>		~
Details				
Summary	Add to version control:	<none></none>		 Manage
	Files to be added in			
	Files to be added in			
	D. (0mp (1211			
	main.cpp			
	mainwindow.cpp mainwindow.h			
	mainwindow.ui			
	TD1.pro			
				Finish
				Conce

QtCreator crée votre projet et les fichiers de base. On obtient un affichage sensiblement similaire à celui-ci :

🕵 m	ainwind	low.cpp	- TD1 - Q	t Creator						
File	Edit	Build	Debug	Analyze	Tools	Window	Hel	р		
		Projects				₽, ⊗ ₽+	×	+ +		mainwindow.cpp
Vel Vel De De Pro	dit sign jects		TD1 TD1.p Heade Source ma	ro es ain.cpp ainwindow.	срр			1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	4	<pre>#include "mainwindow.h" #include "ui_mainwindow.h" MainWindow::MainWindow(QWidget *parent) : QMainWindow(parent), ui(new Ui::MainWindow) { ui->setupUi(this); } MainWindow::~MainWindow() { delete ui; }</pre>

1-b Ajouter les objets à l'UI : *Label* et *PushButton*. Voici la démarche.

Cliquer sur « Form » puis double clic sur le fichier *mainwindow.ui*. Vous basculez sur le mode « Design » des interfaces graphiques dont voici un descriptif :



Il est possible de changer les noms des objets soit via la « liste du contenu de la fenêtre » soit en sélectionnant un objet et en modifiant son nom dans la zone « Détails du *widget* sélectionné ». Dans la zone « détails du widget sélectionné », il est aussi possible de paramétrer graphiquement l'objet (widget) sélectionné.

Faire un *drag and drop* d'un *Label* et d'un *PushButton* sur l'aperçu de votre fenêtre. Votre fenêtre doit ressembler à ceci (il est possible de faire plus joli...) :



1-c Vérifier que la compilation et le lancement de l'application se fassent correctement. Rien ne se produit lors de l'appui sur le bouton puisque qu'aucun code n'a été entré lorsque l'évènement « on clique sur le bouton » se produit. C'est le but de la question suivante.

1-d Affichage d'un « Hello world » lors d'un clic sur le bouton. Pour faire ceci il faut comprendre 2 choses :

- 1. à chaque fois que l'on pose un *widget* sur la fenêtre, QtCreator ajoute un objet du type du *widget* dans la variable *ui* de la fenêtre.
- 2. chaque objet dispose d'évènement (cliquer, survoler, relâcher, ...) sous forme de fonction que l'on va surcharger. Il s'agit des *slot*.

Un clic bouton droit sur l'objet PushButton, puis « Go to Slot » et choisir clicked().

elect signal		
clicked()	QAbstractButton	
clicked(bool)	QAbstractButton	
pressed()	QAbstractButton	=
released()	QAbstractButton	
toggled(bool)	QAbstractButton	
destroyed()	OObiect	-
	ОК	Cancel

Une fonction *on_pushButton_clicked()* est ajoutée à la classe *MainWindow*. Cette fonction sera exécutée lorsque le bouton sera cliqué. Ajouter les lignes suivantes à cette fonction et exécuter le programme :

```
QString txt("Hello les GE");
ui->label->setText(txt);
```

1-d Ajouter 2 *Double Spin Box* et une *ComboBox* à votre fenêtre et réaliser l'exercice 1b du TD0 (opérations +,-, /,*, %). Les calculs ne doivent pas être écrits dans la classe MainWindow mais dans un fichier cpp séparé. Votre application doit ressembler à ceci :

MainWindow		23
4		
12,00	30,00	▲ ▼
=	360	

Utiliser la touche « F1 » pour avoir de l'aide sur les classes des widgets et des classes Qt.

Ce programme n'est pas très complexe, mais la programmation n'est pas si évidente ! Quelles sont les difficultés rencontrées ? Comment pensez-vous pouvoir devenir plus efficace ?

Exercice 2 : vector

La classe *vector* est une classe permettant de gérer dynamiquement et efficacement des tableaux de tous types (double, char, Complexe, ...). On précise le type d'éléments lors de la création d'un objet *vector*. Des exemples d'utilisations sont donnés dans le cours. La documentation complète de la classe *vector* peut être consultée ici :

http://www.cplusplus.com/reference/vector/vector/

2-a Utiliser des objets de la classe *vector* de la std pour réaliser les mêmes additions que la question 2-f du TD0. Vous pourrez compléter le code TD_Ex2_main_TODO.cpp fourni.

2-b Questions

- Pourquoi utiliser des passages de paramètre par référence pour des objets vector ?
- A quoi sert le mot clé const lors du passage de paramètre par référence ?

2-c Utiliser la fonction *sort* pour trier votre tableau en ordre croissant.

2-d Utiliser la fonction *sort* pour trier votre tableau en ordre décroissant.

2-e Question hors préparation. La classe *vector* n'implémente pas les opérations arithmétiques sur des objets *vector*. La classe *valarray* est dédiée à la représentation de vecteurs et à leur manipulation mathématique. Refaire les questions 2-a, 2-c et 2-d avec des objets de la classe *valarray*.