

ISIMA 3^{ème} année - MODL/C++

TP 3 : Introduction à Qt

Exercice 1 : la communication avec signaux et slots

Principe de base :

Le moyen privilégié pour la communication entre différents objets en Qt est l'envoi de messages : l'objet émetteur active un de ses *signaux*, ce signal est connecté à un *slot* d'un objet récepteur via la couche système Qt, puis l'objet récepteur réalise le code correspondant au slot. Les signaux et les slots sont des méthodes particulières comme vu en cours.

- Reprendre l'exemple vu en cours pour faire communiquer un **Compteur** c1 avec un **Compteur** c2. Chaque compteur stocke une valeur entière. Le signal sera la méthode `valeurChangee(int)` et le slot la méthode `setValeur(int)`. Toute modification du compteur c1 devra entraîner la modification du compteur c2.
- Même chose, mais on souhaite lier également c1 à c2 : toute modification de c2 doit entraîner une modification de c1 (une modification simple est nécessaire pour éviter une boucle infinie de messages).
- Ajouter maintenant une classe **GroupeCompteur** contenant des compteurs synchronisés entre eux. Chaque compteur d'une même instance de cette classe doit donc communiquer avec tous les autres. Le groupe est dynamique, il évolue au cours du temps par l'ajout et la suppression de compteur. La définition figée des communications proposée précédemment est invalide, il faut donc proposer un mécanisme simple pour adapter les connexions à l'entrée et à la sortie d'un élément du groupe. Remarque : la méthode `disconnect` de **QObject** permet de déconnecter un signal d'un récepteur.
- Proposer ensuite une version dans laquelle c'est le groupe qui gère l'envoi des messages à tous les autres.

Exercice 2 : une première interface graphique

On souhaite réaliser une application simple avec Qt permettant d'ouvrir un fichier de données, de calculer des statistiques et d'afficher les résultats. Le fichier contient n nombres (des réels compris entre 0 et 20). On souhaite afficher la moyenne et l'écart-type.

- Créer une fenêtre simple (**QMainWindow**). La barre de menu est un objet **QMenuBar**, et comporte des menus (**QMenu**), eux-mêmes composés d'actions (**QAction**). Ajouter dans un premier temps un menu « *Fichier* » avec l'action « *Quitter* » qui devra fermer la fenêtre.
- Ajouter des composants pour afficher les données : **QTextEdit** pour la liste des valeurs, et des **QLabel** pour le nombre de valeurs, la moyenne et l'écart-type. Utiliser un ou plusieurs *layouts* (**QHBoxLayout**, **QVBoxLayout**, **QGridLayout**...) pour disposer les composants à sa guise.
- Réaliser la lecture d'un fichier et l'affichage des valeurs dans le composant correspondant. Respecter l'architecture MVC : les données doivent être représentées en dehors de l'interface graphique (i.e. le modèle), et l'interface vient interroger le modèle.

- d) Ajouter une boîte de dialogue pour sélectionner le fichier à charger (**QFileDialog**).
- e) Ajouter un ascenseur horizontal (**QSlider**) à côté de la moyenne permettant de la modifier par translation de l'échantillon.
- f) Autoriser la modification directe de la valeur de la moyenne en remplaçant le **QLabel** contenant la valeur par un objet **QDoubleSpinBox**.
- g) Réaliser la sauvegarde de l'échantillon modifié.
- h) Ajouter la capacité d'annulation de la dernière opération réalisée.