

Rapport de Projet en Programmation Orientée Objet C++

Sallio Romane and Fontaine Pierre

20 avril 2017

Table des matières

I	Introduction	1
II	Widgets	3
0.1	Menu	4
0.2	ToDoList	4
0.3	Météo	4
0.4	Actualité	4
0.5	Horloge	5
0.6	Convertisseur	5

Résumé

Ce projet à pour objectif de ré-introduire les notions vu en cours de programmation orientée objet et introduites lors des TPs. Pour cela nous avons donc créé un Dashboard composée de 5 widgets (météo,actualité,todolist,convertisseur et horloge).

Première partie

Introduction

Pour ce projet nous avons choisis d'implémenter un "Dashboard", cela se présente sous forme d'une fenêtre graphique qui est développée en C++ grâce au framework QT largement utilisé dans le monde professionnel de l'informatique.

Il implémente une todolist, la recherche de la météo grâce à une API, la recherche d'actualité grâce à un flux RSS, une horloge et un convertisseur de vitesse et de température.

Deuxième partie

Widgets

0.1 Menu

Le menu est composé d'un main.cpp, mainwindow.cpp/hpp, module.cpp/hpp. Le main.cpp permet l'affichage de la fenetre principal et la création du dossier dashboard.

La classe MainWindow permet la création des différents modules et de les afficher. La classe Module est le modèle pour les classes ToDoListModule, MeteoModule.. qui en dérivent.

0.2 ToDoList

***La todolist est composée de todolistajout.cpp/hpp, todolistdata.cpp/hpp, todolistdisplay.cpp/hpp, todolistmodule.hpp/hpp.

La classe ToDoListModule est une classe qui hérite de QWidget, elle permet d'afficher le calendrier et d'ouvrir la fenêtre ToDoListAjout lorsque l'on double cliques sur une date supérieure ou égale à la date actuelle.

ToDoListAjout permet à l'utilisateur de rentrer les informations titre, note et heure. La classeToDoListDisplay est utilisée pour afficher la liste des informations données par l'utilisateur.

ToDoListData permet la lecture et l'écriture dans un fichier des différentes informations.

0.3 Météo

***La recherche de la météo est composée de meteodata.cpp/hpp, meteojour.cpp/hpp, meteomodule.cpp/hpp et meteoparam.cpp/hpp.

0.4 Actualité

***La recherche d'actualité est composée de actudata.cpp/hpp, actumodule.cpp/hpp.

0.5 Horloge

L’horloge est composée de `horlogemodule.cpp` et `horlogemodule.hpp`.
Nous avons utilisé un timer ce qui permet l’actualisation de la date et de l’heure toutes les 0.1s.

0.6 Convertisseur

Le convertisseur est composé de `abstractmeasureunite.cpp/hpp`, `celsius.cpp/hpp`, `convertmodule.cpp/hpp`, `fahrenheit.cpp/hpp`, `kelvin.cpp/hpp`, `kilometre.cpp/hpp`, `metre.cpp/hpp`, `miles.cpp/hpp`, `temperature.cpp/hpp`, `vitesse.cpp/hpp` et `volume.cpp/hpp`.”

La classe `AbstractMesureUnite` est une classe abstraite qui sert de plateforme de base pour les classes `Vitesse` et `Température`.

Les classes `Vitesse` et `Temperature` servent de plateformes pour les classes `Celsius`, `Fahrenheit`, `Kelvin`, `Kilometre`, `Metre`, `Miles` et `Volume`.

Les sept classes cités précédemment permettent de récupérer un réel, de la convertir dans la mesure voulu et d’afficher le réel obtenu après conversion.