

Projet POO, Dashboard

Sallio Romane, Fontaine Pierre

UPPA

26 avril 2017

- 1 introduction
- 2 Pourquoi QT
 - Des classes
 - JS Like
- 3 Spécifications techniques du code
 - Template
 - Heritage
 - Classe Complexe
 - Classe Abstraite

Dashboard est le projet que nous avons conçu en C++ avec le paradigme Objet.

Nous avons utilisé le Framework QT pour développer l'interface utilisateur.

Son objectif est simple : permettre à l'utilisateur d'avoir les éléments essentiels à porté de click.

QT est écrit en C++ et est implémenté selon le paradigme objet. Chacun des composants réfère à une classe particulière qui peut ou non dériver d'une autre classe mère.

Lors de l'utilisation d'un nouveau *Framework*, une partie crucial du temps est concacré à l'étude du fonctionnement de celui ci. Il semblait évident qu'après avoir étudiier le *JavaScript*, le *QT* qui partage la même philosophie de la gestion d'évènement (*Async/Sync*) serait plus digeste.

Utilisé dans *List.h*

Pourquoi ?

- 1 Créer une liste de n'importe quoi
- 2 Container important

```
1  template <class T>
2  class List{
3  protected:
4      struct cellule{
5          cellule *suivant;
6          T valeur;
7      };
8      typedef cellule* liste;
9      liste _l;
10 public:
11     List();
12     ~List();
13     List(const List<T> &l);
14     List<T> operator=(const List<T>);
15
16
17     void insertElemAtPos(int pos,T elt);
18     void deleteElemAtPos(int pos);
19     T getElemAtPos(int pos)const;
20     int getLength()const;
21     bool empty()const;
22
23     };
```

heritage ...

classe complexe ...

classe abstraite ...