

Projet POO, Dashboard

Romane Sallio Pierre Fontaine

Sommaire

o introduction

o Pourquoi QT

> Des classes JS Like

Spécifications techniques du code

Template
Forme
canonique
de coplien

Heritage

Classe Complexe

Classe Abstraite 1 introduction

2 Pourquoi QT

Des classes JS Like

3 Spécifications techniques du code

Template
Forme canonique de coplien
Heritage
Classe Complexe
Classe Abstraite



Introduction

• introduction

o Pourquoi QT

> Des classes JS Like

Spécifications techniques du code

Template

Forme canonique de coplien

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.



Pourquoi QT

Paradigme Objet

o introduction

o Pourquoi **QT**

Des classes

JS Like

o Spécifications techniques du code

Template

Forme canonique

de coplien

Heritage

Classe

Complexe

Classe

Abstraite

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



Pourquoi QT

JS LIKE

o introduction

o Pourquoi QT

Des classes

JS Like

Spécifications techniques du code

Template

Forme canonique de coplien

Heritage

Classe

Complexe

Classe

Abstraite

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 étudier le *JavaScript*, le *QT* qui partage la même philosophie de la gestion d'évènement (*Async/Sync*) serait plus digeste.



Template

Utilité?

o introduction

o Pourquoi QT

Des classes

JS Like

Spécifications
 techniques
 du code

Template

Forme canonique de coplien

Heritage Classe

Complexe

Classe

Abstraite

Utilisé dans List.h

Pourquoi?

- Oréer une liste de n'importe quoi
- 2 Container important



Template

Code

```
tion
o Pourquoi
QT
```

o introduc-

Des classes

JS Like

o Spécifications techniques du code

Template

Forme canonique de coplien

Heritage

Classe Complexe

Classe Abstraite template <class T> class List{...};



Forme canonique de coplien

Pourquoi

o introduction

o Pourquoi QT

> Des classes JS Like

 Spécifications techniques du code

Template

Forme canonique de coplien

Heritage Classe

Complexe

Classe Abstraite Il est interressant dans la conception de *container* comme des listes, des sets ... qu'ils puissent s'affecter entre eux, se construire à partir d'un modèle déjà existant.

C'est pourquoi nous intégrons ce modèle dans quelques classes comme :

- 1 list.h
- 2 abstractmesureunite.h



Forme canonique de coplien

code 1

```
o introduc-
tion
o Pourquoi
QT
  Des classes
  JS Like
o Spécifica-
tions
techniques
du code
  Template
  Forme
  canonique
  de coplien
  Heritage
  Classe
  Complexe
  Classe
  Abstraite
```

```
class AbstractMesureUnite{
protected:
    double _value;
public:
    AbstractMesureUnite();
    AbstractMesureUnite(double);
    AbstractMesureUnite(const AbstractMesureUnite&);
    ~AbstractMesureUnite();
    AbstractMesureUnite & operator=(const AbstractMesureUnite&);
    ...
};
```



Forme canonique de coplien

code 2

```
o introduc-
tion
                template <class T>
                class List{
o Pourquoi
               protected:
QT
                    struct cellule{
 Des classes
                         cellule *suivant:
 JS Like
                         T valeur:
o Spécifica-
                     };
tions
                    typedef cellule* liste;
techniques
                    liste 1:
du code
               public:
 Template
                    List();
 Forme
                    ~List();
 canonique
                    List(const List<T> &1);
 de coplien
                    List<T> operator=(const List<T>);
 Heritage
                     . . .
 Classe
                };
 Complexe
 Classe
```



Abstraite

Heritage

QT

o introduction

PourquoiQT

Des classes JS Like

 Spécifications techniques du code

Template
Forme
canonique
de coplien

Heritage

Classe Complexe

Classe Abstraite

Dérivation des composants :

- Ohaque composant du framework QT est une classe
- 2 Créer un composant spécifique se fait en dérivant un composant général

tips

Il est courant que les classes les plus généralistes ne soient pas instanciables, ce sont des classes *abstraites*.



Heritage

Hériter un composant QT

```
o introduc-
tion
                #include <QWidget>
o Pourquoi
                class Module : public QWidget{
QT
                    O OBJECT
 Des classes
               protected:
 JS Like
                  //méthodes protégées.
o Spécifica-
               public:
tions
                  explicit Module(QWidget *parent = 0, double h = 300,
techniques
                   \rightarrow double w = 400);
du code
                  //méthodes publiques
 Template
 Forme
                signals:
 canonique
                  //signals
 de coplien
               public slots:
 Heritage
                  //slots
 Classe
                };
 Complexe
```



Classe Abstraite

Heritage

Fn dehors de QT

tion o Pourquoi

o introduc-

Des classes

JS Like

o Spécifications techniques du code

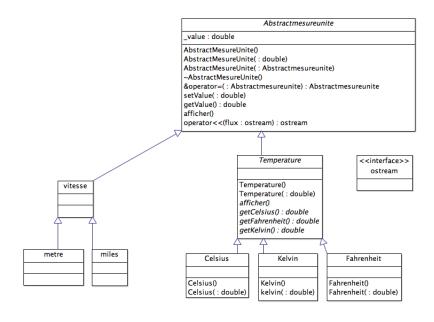
Template

Forme canonique de coplien

Heritage

Classe Complexe

Classe Abstraite





o introduction

o Pourquoi QT

Des classes

JS Like

Spécifications techniques du code

Template

Forme canonique de coplien

Heritage

Classe Complexe

Classe Abstraite classe complexe ...



o introduction

o Pourquoi QT

Des classes

JS Like

Spécifications techniques du code

Template

Forme canonique de coplien

Heritage

Classe Complexe

Classe Abstraite classe abstraite ...

