

# Introduction au module POOGL

Yves Caniou

2 février 2005

# Programmation Orientée Objet et Génie Logiciel ?

- ▶ Là dedans, vous connaissez **Programmation**
- ▶ Vous avez peut-être des notions d'**Orientée Objet**
- ▶ Le mot **Génie** vous rappelle l'armée, ou alors Mozart

# Programmation Orientée Objet et Génie Logiciel ?

- ▶ Là dedans, vous connaissez **Programmation**
- ▶ Vous avez peut-être des notions d'**Orientée Objet**
- ▶ Le mot **Génie** vous rappelle l'armée, ou alors Mozart

# Programmation Orientée Objet et Génie Logiciel ?

- ▶ Là dedans, vous connaissez **Programmation**
- ▶ Vous avez peut-être des notions d'**Orientée Objet**
- ▶ Le mot **Génie** vous rappelle l'armée, ou alors Mozart

# Programmation Orientée Objet et Génie Logiciel ?

- ▶ Là dedans, vous connaissez **Programmation**
- ▶ Vous avez peut-être des notions d'**Orientée Objet**
- ▶ Le mot **Génie** vous rappelle l'armée, ou alors Mozart (Aladdin ?)

# Oui mais on sait déjà programmer OO

Vous aurez quand même quelques cours magistraux sur le sujet...

▶ Un programme classique

```
#include <stdio.h>
int main(int argc, char** argv)
{
    printf ("Hello World!\n");
}
```

▶ Un programme orienté objet

```
#include <iostream>
int main(int argc, char** argv)
{
    std::cout << "Hello, world!\n";
}
```

# Oui mais on sait déjà programmer OO

Vous aurez quand même quelques cours magistraux sur le sujet...

- ▶ Un programme classique

```
#include <stdio.h>
int main(int argc, char** argv)
{
    printf ("Hello World!\n");
}
```

- ▶ Un programme orienté objet

```
#include <iostream>
int main(int argc, char** argv)
{
    std::cout << "Hello, world!\n";
}
```

# Oui mais on sait déjà programmer OO

Vous aurez quand même quelques cours magistraux sur le sujet...

- ▶ Un programme classique

```
#include <stdio.h>
int main(int argc, char** argv)
{
    printf ("Hello World!\n");
}
```

- ▶ Un programme orienté objet

```
#include <iostream>
int main(int argc, char** argv)
{
    std::cout << "Hello, world!\n";
}
```



# Oui mais on sait déjà programmer OO

Vous aurez quand même quelques cours magistraux sur le sujet...

► Un programme classique

```
#include <stdio.h>
int main(int argc, char** argv)
{
    printf ("Hello World!\n");
}
```

► Un programme orienté objet

```
#include <iostream>
int main(int argc, char** argv)
{
    std::cout << "Hello, world!\n";
}
```

# Validation du module ?

- ▶ Un projet
- ▶ Par groupes de trois, avec dérogations...
- ▶ Faire un logiciel
  - ▶ qui marche
  - ▶ robuste
  - ▶ documenté
  - ▶ extensible

## Et faire tout un tas de paperasses “inutiles”

- ▶ Avant :
  - ▶ faire un cahier des charges
  - ▶ faire un échéancier
  - ▶ définir l'architecture du logiciel
  - ▶ se répartir le boulot
- ▶ Pendant
  - ▶ travailler à plusieurs
  - ▶ respecter l'échéancier
  - ▶ documenter au fur et à mesure
- ▶ Après
  - ▶ faire la doc utilisateur
  - ▶ faire une démo

# Tout cela ne paraît pas très marrant

... c'est pourquoi ce module est quasi obligatoire 😊

Par ailleurs on apprend tout de même quelques outils au passage.

- ▶ UML, un langage graphique standard pour décrire des objets, des classes, etc ;
- ▶ CVS, un outil pour travailler à plusieurs sur un projet en minimisant le risque de se bousiller le boulot les uns des autres.
- ▶ *autoconf/automake* si c'est vraiment indispensable.
- ▶ Plus un ou plusieurs langages de programmation pour certains.

# Tout cela ne paraît pas très marrant

... c'est pourquoi ce module est quasi obligatoire 😊

Par ailleurs on apprend tout de même quelques outils au passage.

- ▶ UML, un langage graphique standard pour décrire des objets, des classes, etc ;
- ▶ CVS, un outil pour travailler à plusieurs sur un projet en minimisant le risque de se bousiller le boulot les uns des autres.
- ▶ *autoconf/automake* si c'est vraiment indispensable.
- ▶ Plus un ou plusieurs langages de programmation pour certains.

# Tout cela ne paraît pas très marrant

... c'est pourquoi ce module est quasi obligatoire 😊

Par ailleurs on apprend tout de même quelques outils au passage.

- ▶ UML, un langage graphique standard pour décrire des objets, des classes, etc ;
- ▶ CVS, un outil pour travailler à plusieurs sur un projet en minimisant le risque de se bousiller le boulot les uns des autres.
- ▶ *autoconf/automake* si c'est vraiment indispensable.
- ▶ Plus un ou plusieurs langages de programmation pour certains.

# Tout cela ne paraît pas très marrant

... c'est pourquoi ce module est quasi obligatoire 😊

Par ailleurs on apprend tout de même quelques outils au passage.

- ▶ UML, un langage graphique standard pour décrire des objets, des classes, etc ;
- ▶ CVS, un outil pour travailler à plusieurs sur un projet en minimisant le risque de se bousiller le boulot les uns des autres.
- ▶ *autoconf/automake* si c'est vraiment indispensable.
- ▶ Plus un ou plusieurs langages de programmation pour certains.

# Tout cela ne paraît pas très marrant

... c'est pourquoi ce module est quasi obligatoire 😊

Par ailleurs on apprend tout de même quelques outils au passage.

- ▶ UML, un langage graphique standard pour décrire des objets, des classes, etc ;
- ▶ CVS, un outil pour travailler à plusieurs sur un projet en minimisant le risque de se bousiller le boulot les uns des autres.
- ▶ *autoconf/automake* si c'est vraiment indispensable.
- ▶ Plus un ou plusieurs langages de programmation pour certains.



## Tout cela ne paraît pas très marrant

... c'est pourquoi ce module est quasi obligatoire 😊

Par ailleurs on apprend tout de même quelques outils au passage.

- ▶ UML, un langage graphique standard pour décrire des objets, des classes, etc ;
- ▶ CVS, un outil pour travailler à plusieurs sur un projet en minimisant le risque de se bousiller le boulot les uns des autres.
- ▶ *autoconf/automake* si c'est vraiment indispensable.
- ▶ Plus un ou plusieurs langages de programmation pour certains.

# Sujets de projets ?

- ▶ Sujets autogènes : vous avez une idée et vous voulez la développer.
  - ▶ on encourage
  - ▶ le chef de projet sera moi, pas vous
- ▶ Sujets exogènes : des chercheurs ont du boulot pour vous
  - ▶ venant d'autres labos ?
  - ▶ le chef de projet sera un chercheur

# Sujets de projets ?

- ▶ Sujets autogènes : vous avez une idée et vous voulez la développer.
  - ▶ on encourage
  - ▶ le chef de projet sera moi, pas vous
- ▶ Sujets exogènes : des chercheurs ont du boulot pour vous
  - ▶ venant d'autres labos ?
  - ▶ le chef de projet sera un chercheur

# Sujets de projets ?

- ▶ Sujets autogènes : vous avez une idée et vous voulez la développer.
  - ▶ on encourage
  - ▶ le chef de projet sera moi, pas vous
- ▶ Sujets exogènes : des chercheurs ont du boulot pour vous
  - ▶ venant d'autres labos ?
  - ▶ le chef de projet sera un chercheur

# Langages utilisés ?

- ▶ Java *par défaut* voire plus
- ▶ C++, Ruby, Python si bonneS raisonS
- ▶ Jeu : les raisons suivantes sont-elles bonnes?
  - C++ : pas de template pas Java

• Si il y a une bibliothèque qui existe en yyy qui n'existerait pas la moitié du boulot.

- ▶ Ocaml est le fleuron du logiciel d'excellence français, c'est mon devoir de patriote de l'utiliser.
- ▶ Java est pas GPL, il est pas libre, je refuse de polluer ma Debian avec.
- ▶ J'ai déjà eu du mal avec Caml, je veux pas apprendre un autre langage.

# Langages utilisés ?

- ▶ Java *par défaut*, voire plus
- ▶ C++, Ruby, Python si bonneS raisonS
- ▶ Jeu : les raisons suivantes sont-elles bonnes ?

- ▶ il y a une bibliothèque qui existe en yyy qui m'économiserait la moitié du boulot.
- ▶ Ocaml est le fleuron du logiciel d'excellence français, c'est mon devoir de patriote de l'utiliser.
- ▶ Java est pas GPL, il est pas libre, je refuse de polluer ma Debian avec.
- ▶ J'ai déjà eu du mal avec Caml, je veux pas apprendre un autre langage.

# Langages utilisés ?

- ▶ Java *par défaut*, voire plus
- ▶ C++, Ruby, Python si bonneS raisonS
- ▶ Jeu : les raisons suivantes sont-elles bonnes ?

## ▶ C++ est complexe pas libre

- ▶ Il y a une bibliothèque qui existe en yyy qui m'économiserait la moitié du boulot.
- ▶ Ocaml est le fleuron du logiciel d'excellence français, c'est mon devoir de patriote de l'utiliser.
- ▶ Java est pas GPL, il est pas libre, je refuse de polluer ma Debian avec.
- ▶ J'ai déjà eu du mal avec Caml, je veux pas apprendre un autre langage.

# Langages utilisés ?

- ▶ Java *par défaut*, voire plus
- ▶ C++, Ruby, Python si bonneS raisonS
- ▶ Jeu : les raisons suivantes sont-elles bonnes ?
  - ▶ C++ a des templates, pas Java.
  - ▶ On n'est plus au moyen-âge, on ne peut plus le régresser, nous-même, C++ est exclus.
  - ▶ J'ai déjà eu du mal avec Java.
  - ▶ Il y a une bibliothèque qui existe en yyy qui m'économiserait la moitié du boulot.
  - ▶ Ocaml est le fleuron du logiciel d'excellence français, c'est mon devoir de patriote de l'utiliser.
  - ▶ Java est pas GPL, il est pas libre, je refuse de polluer ma Debian avec.
  - ▶ J'ai déjà eu du mal avec Caml, je veux pas apprendre un autre langage.



# Langages utilisés ?

- ▶ Java *par défaut*, voire plus
- ▶ C++, Ruby, Python si bonneS raisonS
- ▶ Jeu : les raisons suivantes sont-elles bonnes ?
  - ▶ C++ a des *templates*, pas Java.
  - ▶ On n'est plus au moyen-âge, on va pas gérer la mémoire nous-même, C++ est exclus.
  - ▶ Je dois interfacer mon code avec du xxx.
  - ▶ Il y a une bibliothèque qui existe en yyy qui m'économiserait la moitié du boulot.
  - ▶ Ocaml est le fleuron du logiciel d'excellence français, c'est mon devoir de patriote de l'utiliser.
  - ▶ Java est pas GPL, il est pas libre, je refuse de polluer ma Debian avec.
  - ▶ J'ai déjà eu du mal avec Caml, je veux pas apprendre un autre langage.

# Langages utilisés ?

- ▶ Java *par défaut*, voire plus
- ▶ C++, Ruby, Python si bonneS raisonS
- ▶ Jeu : les raisons suivantes sont-elles bonnes ?
  - ▶ C++ a des *templates*, pas Java.
  - ▶ On n'est plus au moyen-âge, on va pas gérer la mémoire nous-même, C++ est exclus.
  - ▶ Je dois interfacer mon code avec du xxx.
  - ▶ Il y a une bibliothèque qui existe en yyy qui m'économiserait la moitié du boulot.
  - ▶ Ocaml est le fleuron du logiciel d'excellence français, c'est mon devoir de patriote de l'utiliser.
  - ▶ Java est pas GPL, il est pas libre, je refuse de polluer ma Debian avec.
  - ▶ J'ai déjà eu du mal avec Caml, je veux pas apprendre un autre langage.

# Langages utilisés ?

- ▶ Java *par défaut*, voire plus
- ▶ C++, Ruby, Python si bonneS raisonS
- ▶ Jeu : les raisons suivantes sont-elles bonnes ?
  - ▶ C++ a des *templates*, pas Java.
  - ▶ On n'est plus au moyen-âge, on va pas gérer la mémoire nous-même, C++ est exclus.
  - ▶ Je dois interfacier mon code avec du xxx.
  - ▶ Il y a une bibliothèque qui existe en yyy qui m'économiserait la moitié du boulot.
  - ▶ Ocaml est le fleuron du logiciel d'excellence français, c'est mon devoir de patriote de l'utiliser.
  - ▶ Java est pas GPL, il est pas libre, je refuse de polluer ma Debian avec.
  - ▶ J'ai déjà eu du mal avec Caml, je veux pas apprendre un autre langage.

# Langages utilisés ?

- ▶ Java *par défaut*, voire plus
- ▶ C++, Ruby, Python si bonneS raisonS
- ▶ Jeu : les raisons suivantes sont-elles bonnes ?
  - ▶ C++ a des *templates*, pas Java.
  - ▶ On n'est plus au moyen-âge, on va pas gérer la mémoire nous-même, C++ est exclus.
  - ▶ Je dois interfacer mon code avec du xxx.
  - ▶ Il y a une bibliothèque qui existe en yyy qui m'économiserait la moitié du boulot.
  - ▶ Ocaml est le fleuron du logiciel d'excellence français, c'est mon devoir de patriote de l'utiliser.
  - ▶ Java est pas GPL, il est pas libre, je refuse de polluer ma Debian avec.
  - ▶ J'ai déjà eu du mal avec Caml, je veux pas apprendre un autre langage.

# Langages utilisés ?

- ▶ Java *par défaut*, voire plus
- ▶ C++, Ruby, Python si bonneS raisonS
- ▶ Jeu : les raisons suivantes sont-elles bonnes ?
  - ▶ C++ a des *templates*, pas Java.
  - ▶ On n'est plus au moyen-âge, on va pas gérer la mémoire nous-même, C++ est exclus.
  - ▶ Je dois interfacer mon code avec du xxx.
  - ▶ Il y a une bibliothèque qui existe en yyy qui m'économiserait la moitié du boulot.
  - ▶ Ocaml est le fleuron du logiciel d'excellence français, c'est mon devoir de patriote de l'utiliser.
  - ▶ Java est pas GPL, il est pas libre, je refuse de polluer ma Debian avec.
  - ▶ J'ai déjà eu du mal avec Caml, je veux pas apprendre un autre langage.

# Langages utilisés ?

- ▶ Java *par défaut*, voire plus
- ▶ C++, Ruby, Python si bonneS raisonS
- ▶ Jeu : les raisons suivantes sont-elles bonnes ?
  - ▶ C++ a des *templates*, pas Java.
  - ▶ On n'est plus au moyen-âge, on va pas gérer la mémoire nous-même, C++ est exclus.
  - ▶ Je dois interfacer mon code avec du xxx.
  - ▶ Il y a une bibliothèque qui existe en yyy qui m'économiserait la moitié du boulot.
  - ▶ Ocaml est le fleuron du logiciel d'excellence français, c'est mon devoir de patriote de l'utiliser.
  - ▶ Java est pas GPL, il est pas libre, je refuse de polluer ma Debian avec.
  - ▶ J'ai déjà eu du mal avec Caml, je veux pas apprendre un autre langage.

# Langages utilisés ?

- ▶ Java *par défaut*, voire plus
- ▶ C++, Ruby, Python si bonneS raisonS
- ▶ Jeu : les raisons suivantes sont-elles bonnes ?
  - ▶ C++ a des *templates*, pas Java.
  - ▶ On n'est plus au moyen-âge, on va pas gérer la mémoire nous-même, C++ est exclus.
  - ▶ Je dois interfacer mon code avec du xxx.
  - ▶ Il y a une bibliothèque qui existe en yyy qui m'économiserait la moitié du boulot.
  - ▶ Ocaml est le fleuron du logiciel d'excellence français, c'est mon devoir de patriote de l'utiliser.
  - ▶ Java est pas GPL, il est pas libre, je refuse de polluer ma Debian avec.
  - ▶ J'ai déjà eu du mal avec Caml, je veux pas apprendre un autre langage.

# Langages utilisés ?

- ▶ Java *par défaut*, voire plus
- ▶ C++, Ruby, Python si bonneS raisonS
- ▶ Jeu : les raisons suivantes sont-elles bonnes ?
  - ▶ C++ a des *templates*, pas Java.
  - ▶ On n'est plus au moyen-âge, on va pas gérer la mémoire nous-même, C++ est exclus.
  - ▶ Je dois interfacer mon code avec du xxx.
  - ▶ Il y a une bibliothèque qui existe en yyy qui m'économiserait la moitié du boulot.
  - ▶ Ocaml est le fleuron du logiciel d'excellence français, c'est mon devoir de patriote de l'utiliser.
  - ▶ Java est pas GPL, il est pas libre, je refuse de polluer ma Debian avec.
  - ▶ J'ai déjà eu du mal avec Caml, je veux pas apprendre un autre langage.



# Échéancier approximatif

- ▶ 2-9 février : les sujets sont à-peu-près fixés et approuvés par moi.
- ▶ Février : je fais quelques cours, et vous modélisez.
- ▶ Les séances du mois de mars sont consacrées à des exposés/discussions de votre modélisation.
- ▶ En avril vous programmez, et je fais du support Java.
- ▶ Le code est rendu le jeudi 5 mai à 23h59 (sauf contre-ordre).
- ▶ Une journée de démos, publique, par exemple le mercredi 11 mai.

Des questions ?