Approches Objet de la Programmation

- Numero: 1
- Prof: Didier Verna
- Date: 9 Octobre 2017

Introduction

Mail du prof: didier@lrde.epita.fr

Site du prof

Slides

Il y a aussi des tarballs avec des programmes complet a tester.

On va etudier les **Paradigmes**. * Approches classiques de la programmation Objet. (Ce cours) * Approches de la programmation fonctionnelle. * Approches autres de la programmation Objet (lisp & askell).

La plus part des langages Oriente Objet (C++, Java…) ne sont pas propre en realite d'un point de vue theorique.

Les notions de base de la programmation objet: * Reutilisation du code. * Polymorphisme d'inclusion. * Chaque objet est construit comme un moule (sa classe).

Sommaire

Contexte

Programmation Imperative

Imperatif signifie donner des ordres. La programmation imperative est une suite d'instruction. **Instruction**: Ordre destine a produire un effet de bord. Effet de bord ou **Side effect** signifie modifier l'environnement.

```
Instruction 1
Instruction 2
...
Instruction N
```

L'ordre des executions est important !

Programmation Procedural

Dans les annees 50. C'est une extension logique de la programmes imperative. Elle est nait d'une frustration de la prog imperative.

Elle permet de factoriser une suite d'instruction souvent repetees en procedure que l'on peut appeler autant de fsoi que l'on veut.

```
...
fais ceci
fais cela
...
fais ceci
fais cela
...
```

C'est de la merde car: * Manque de lisibilite. * Bug duplique. * Plus dure a maintenir.

Il faut donner des noms aux choses.

```
fais tout:
  fais ceci
  fais cela
...
fais tout
...
fais tout
...
```

Origine

Modele *Von Neumann*. L'idee c'est que les donnees et les instruction sont la meme chose et peuvent se stocker au meme endroit. C'est cette idee qui nous permet de parler de memoire / ordinateur etc…

Les langages machine font: * Manipulation de registres * Assurer les echanges avec la memoire

Deux familles de langages

Bottom-Up, langages construits en partant du bas niveau et en remontant. Il construit en voulant ajouter de l'abstraction.

Top-Down, prend le contre pied de l'approche historique. On part de l'abstraction et une fois que l'on est satisfait de la theorie alors on l'implemente. (Ex: HTML, Lisp)