

# Approches Objet de la Programmation

## ～ Introduction ～

Didier Verna  
EPITA / LRDE

[didier@lrde.epita.fr](mailto:didier@lrde.epita.fr)



[lrde/~didier](#)



[@didierverna](#)



[didier.verna](#)



[google+](#)



[in/didierverna](#)

# Plan

## Contexte

- Programmation Impérative
- Programmation Procédurale
- Évolution

## Paradigmes de Programmation

- Notion de Paradigme
- Limitations de l'Impératif / Procédural

## L'Approche Orientée-Objet

- Génèse
- Historique



# Plan

## Contexte

- Programmation Impérative
- Programmation Procédurale
- Évolution

## Paradigmes de Programmation

## L'Approche Orientée-Objet



# Programmation Impérative

## Programme

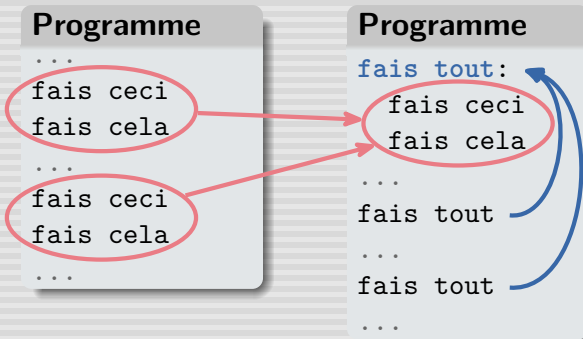
```
instruction 1  
instruction 2  
...  
instruction n
```

- ▶ **Instruction** : ordre destiné à produire un effet de bord
- ▶ **Effet de bord** : modification du contexte environnant
- ▶ Ordre d'exécution important !



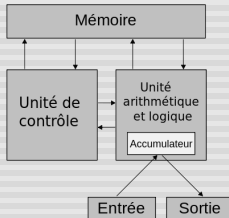
# Programmation Procédurale

- ▶ Extension logique de la programmation impérative
- ▶ Factorisation d'une séquence d'instructions souvent répétées en *procédure* que l'on peut appeler autant de fois que l'on veut



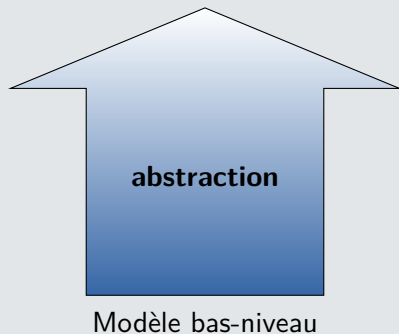
# Origine

- ▶ Paradigmes proches du matériel sous-jacent  
*modèle de Von Neumann*
- ▶ Langage machine
  1. manipulation de registres
  2. échanges avec la mémoire

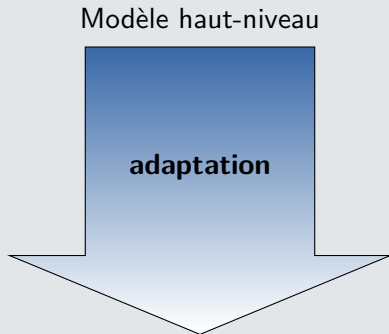


# Deux Familles de Langages

## Bottom-Up



## Top-Down



- La programmation impérative / procédurale est bottom-up

# Plan

Contexte

Paradigmes de Programmation

Notion de Paradigme

Limitations de l'Impératif / Procédural

L'Approche Orientée-Objet





# Paradigmes de Programmation

- ▶ **Notion de paradigme :**
  - ▶ Affecte l'expressivité
  - ▶ Affecte la manière de penser
- ▶ **Des tas :** impératif, procédural, orienté-objet, fonctionnel, logique, méta, etc.
- ▶ **Turing-Complétude :** nos langages de programmation ont tous la même puissance d'expression
- ▶ **La question véritable :**
  - « ~~Peut-on exprimer X ?~~ »
  - « Peut-on exprimer X *facilement* ? »

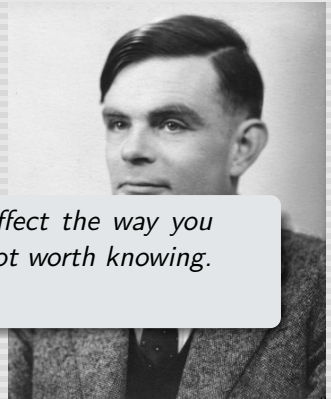


# Paradigmes de Programmation

- ▶ **Notion de paradigme :**
  - ▶ Affecte l'expressivité
  - ▶ Affecte la manière de penser
- ▶ **Des tas :** impératif, procédural, orienté-objet, fonctionnel, logique, méta
- ▶ **Turris** de puissance d'expression
- ▶ **La question véritable :**
  - « ~~Peut-on exprimer X ?~~ »
  - « Peut-on exprimer X *facilement* ? »

*A language that doesn't affect the way you think about programming, is not worth knowing.*

— Alan Perlis



# Spécificités d'Autres Paradigmes

## Fonctionnel

```
map (+4) [1, 2, 3, 4, 5] -- [5, 6, 7, 8, 9]
```

## Logique

```
pere(didier,coline).  
homme(X) :- pere(X,_).  
? - homme(didier).      % yes
```

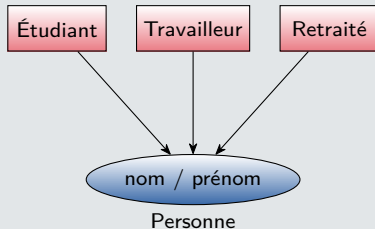
## Méta

```
(defvar program (list '+ 1 2))  
(eval program)           ;; 3
```

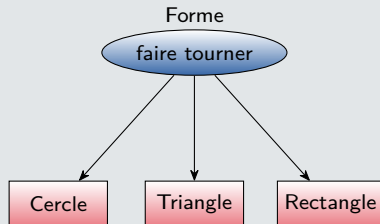


# Deux Limitations en Procédural

## Convergence d'état



## Divergence de comportement



► La POO nous permet de nous affranchir de ces deux limitations

# Plan

Contexte

*Paradigmes de Programmation*

L'Approche Orientée-Objet

Génèse

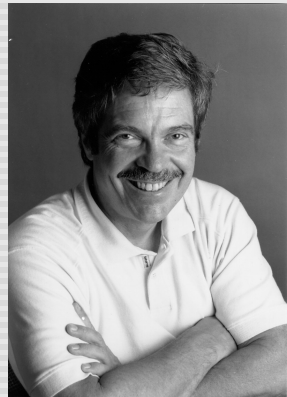
Historique



# Terminologie

*It was probably in 1967 when someone asked me what I was doing, and I said : « It's object-oriented programming. » [...] I thought of objects being like biological cells and/or individual computers on a network, only able to communicate with messages.*

— Alan Kay [Kay, 2003]



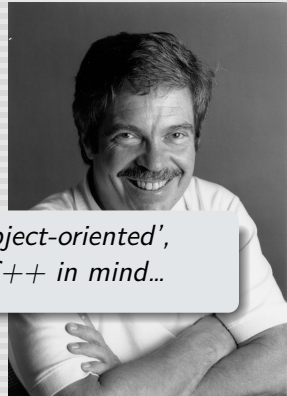
## ► Grosse erreur !

« Message-Oriented Programming » aurait été plus approprié (dixit Alan Kay lui-même)

# Terminologie

*It was probably in 1967 when someone asked me what I was doing, and I said : « It's object-oriented programming. » [...] I thought of objects being like biological cells and/or individual computers on a network. only able to comm*

*— Actually I made up the term 'object-oriented', and I can tell you I did not have C++ in mind...*



## ► Grosse erreur !

« Message-Oriented Programming » aurait été plus approprié (dixit Alan Kay lui-même)

# Les Objets du Monde Réel

- ▶ Un *état* (statique)
- ▶ Un *comportement* (dynamique)

## Personne

nom / prénom / âge

—

mange / travaille / se déplace

## Avion

marque / couleur

—

prend des passagers / se déplace

- ▶ Toute personne a un nom, mais pas le même. Tout salarié a un employeur, mais toute personne n'est pas forcément salariée. Une personne et un avion peuvent se déplacer, mais différemment. *etc.*
- ▶ La POO nous aide à exprimer ce type de relations





# Historique

## Smalltalk (1971, publique en 1980)

- ▶ Alan Kay et Dan Ingalls
- ▶ « Tout objet » et envoi de message
- ▶ Premier langage dit « orienté-objet »
- ▶ Contexte : DynaBook [Kay, 1972]



## Simula (1962, mais surtout 1967)

- ▶ Ole-Johan Dahl et Kristen Nygaard
- ▶ Contient déjà tous les concepts
- ▶ Considérablement inspiré Smalltalk
- ▶ La paternité des concepts est attribuée à Dahl / Nygaard
- ▶ Leur but était pourtant différent de celui de Kay



# Paysage Orienté-Objet Actuel

## ▶ Aucune réelle définition

- ▶ Nombreux langages, visions très différentes
- ▶ Ensemble flou de concepts, ni exclusif, ni exhaustif
- ▶ Concepts mal définis / articulés [Nierstrasz, 2010]

## ▶ Effort de standardisation

- ▶ OMG (Object Management Group)  
*association à but non lucratif américaine créée en 1989*
- ▶ UML (Unified Modelling Language)  
*langage standardisé de l'OMG pour la modélisation graphique de systèmes orientés-objet (composants logiciels, interactions, déroulement séquentiel, déploiement matériel etc.)*



# Paysage Orienté-Objet Actuel

## ▶ Aucune réelle définition

- ▶ Nombreux langages, visions très différentes
- ▶ Ensemble flou de concepts, ni exclusif, ni exhaustif
- ▶ Concepts mal définis / articulés [Nierstrasz, 2010]

## ▶ Effort de standardisation

*Object-oriented programming is an exceptionally bad idea which could only have originated in California*

— Edsger Dijkstra




*systèmes orientés-objet (composants logiciels, interactions, déroulement séquentiel, déploiement matériel etc.)*



# Plan

Bibliographie

## Bibliographie

-  Alan Kay.  
Dr. Alan Kay on the Meaning of “Object-Oriented Programming”.  
*Email à Stefan Ram, 23 Juillet 2003.*
-  Alan Kay.  
A Personal Computer for Children of All Ages.  
ACM Conference, 1972.
-  Oscar Nierstrasz.  
10 Things I Hate About Object Oriented Programming.  
ECOOP Conference Dinner Speech, 2010.