

Typologie des langages

Étienne RENAULT

20 mars 2018

Table des matières

1	Historique	2
1.1	Introduction	2
1.2	Chronologie	2
2	Une histoire des langages de programmation	4
2.1	Première partie	4
2.1.1	FORTRAN	4
2.1.2	ALGOL	4
2.1.3	COBOL	4
2.2	Seconde vague	4
2.2.1	APL	4
2.2.2	PL/I	5
2.2.3	BASIC	5
2.2.4	Pascal & Heirs	5
2.3	The final wave	5
2.3.1	K. N. King	5
3	Histoire de l'orienté objet	5
3.1	Early languages	5
3.1.1	Simula	5
3.1.2	Smalltalk	6
3.1.3	C++	6
3.2	Propriété de l'OO	6
4	Paramètres de fonctions	7
4.1	(N)RVO Return value optimization	7
4.2	Call by name	7
4.3	Default parameters	7
4.4	Subprograms as arguments	7
4.5	Conclusion	7
5	Technologie objet Eiffel	7
5.1	Introduction	7
5.2	Des concepts originaux	7
5.3	Heritage	8
6	Garbage Collector	8

1 Historique

1.1 Introduction

Most of the course can be found here : <https://www.lrde.epita.fr/~tiger/lecture-notes/slides/tyla/>
On a envie d'apprendre les noms des créateurs des langages pour le partiel.

1.2 Chronologie

Vérifier les dates !

- Première machine pour calculer : abaque. (addition, soustraction)
- 1643 : Pascaline, première machine à calculer mécanique. (addition, soustraction)
- 1673 : Première machine avec multiplications et divisions.
- 1725 : Basille Bouchon, mécanisation d'un métier à tisser. Transposition d'un programme sur un support. La machine n'a pas duré faute de fragilité.
- 1780 : Découverte de l'électricité par Benjamin Franklin. (Avec cerf- volant).
- 1805 : Jacquard fait un métier à tisser complètement automatisé à carte perforée et alimenté à l'électricité.
- 1833 : Babbage : machine analytique. Sortie non-tissu. Sortie sur des roues crantées. Ada Lovelace, comtesse, première programmeuse. Elle faisait de la science, rare à l'époque. Elle fait de la traduction de sciences pour la machine de Babbage.
- 1876 : Téléphone, par Alexander Graham Bell et Elisha Gray (stagiaire). Le stagiaire fait tomber un truc (?)
- 1911 : CTRC (ancêtre d'IBM). Doit toquer chez tous les Américains pour faire un recensement. IBM a construit des machines pour faire le recensement.
- 1924 : IBM prend son nom.
- 1927 : Première démonstration publique d'une télévision.
- 1936 : Premier calculator, le Z1. Konrad Zuse. Fait un ordinateur chez ses parents. machine détruite par les Allemands pendant la guerre. Reconstituée après par le même.
- 1939 : Radio Shack (catalogue).
- 1939 : Conception de l'ABC (Atanasoff-Berry Computer). Ruled the first automatic digital computer en 1973. Not programmable, not turing-complete.
- 1940 : Première TV couleur.
- 1941 : Première machine Turing-complète : Zuse's Z3. Basée sur des tubes à vide.
- 1944 : March 1. Machine pour de la défense militaire. 12 000 tubes à vide. Une chance sur 2 pour qu'un tube à vide avant la fin de l'exécution du programme.
- 1945 : Grace Hopper : trouve le premier bug sur le Harvard Mark II. Invente aussi le COBOL plus tard.
- 1946 : ENIAC, 18 000 tubes.
- 1946 ; Univac
- 1948 : Transistor. (Hyper artisanal).
- 1949 : EDVAC : Première machine pour tester des disques magnétiques.
- 1949 : Binac : Premier ordinateur pour diriger des missiles **en temps réel**.
- 1949 : Le MIT construit la première machine qui joue aux échecs.
- 1949 : Premier ordinateur commercialisé (demi-pièce) : Univac II.
- 1952 : RCA développe le Bizmac. Première base de données.
- 1952 : Imprimante à matrice pour l'Univac 1107 dans les années 60.
- 1953 : Bande magnétiques.
- 1953 : IBM 701. Machine à louer. Addition en 5 cycles, comme maintenant. Fortran est créé sur ces machines par John Backus. Révolution. On passe de zéros et de uns à une feuille de papier avec variables et commentaires.
- 1954 : Premier OS pour l'IBM 704, un mainframe.
- 1954 : RAND corporation : modèle pour illustrer l'ordinateur en 2004.
- 1955 : Premier groupe d'utilisateur : SHARE (IBM 701).
- 1955 : IBM demande à chaque pays de traduire computer. Jacques Perret : "ordinateur".
- 1957 : Film "Desk Set". Ressemble aux assistants vocaux actuels.
- 1959 : 8 traîtres. Bossent sur tubes à vide nouvelle génération. Partent tous en même temps.

- 1959 : Texas : Premier circuit intégré.
- 1959 : COBOL
- 1959 : IBM 401 : 10 000 machines vendues.
- 1960 : Téléphone automatisé.
- 1961 : Premier micro-ordinateur. Fourni avec écran.
- PDP-1 : Space war. Premier jeu vidéo.
- 1963 : Tandy achète Radio Shack. Lancement pour le grand public.
- 1964 : Un étudiant du MIT joue de la musique sur le PDP-1.
- 1964 : CDC 6000. Ordinateur le plus puissant de son époque.
- 1964 : Création du BASIC. Language interprété. Interpréteur écrit en Assembleur.
- CDC : Lance des études d'informatique.
- 1965 : Douglas Engelbart : Créé la première souris. Génie. Première démonstration enregistrée.
- Herb Sutter (gourou du C++). Résume toutes les contributions d'Engelbart. A aussi inventé le pattern matching.
- 1965 : IBM 360. Couvrait tous les besoins de l'informaticien.
- 1965 : Premier doctorant en informatique.
- 1968 : Première calculette par Texas et HP.
- 1968 : Fondation d'Intel.
- 1969 : Fondation d'AMD.
- 1969 : IBM intrduit System-3.
- 1969 : Bell Labs arrête MULTICS. Ken Thompson implémente alors UNICS sur un PDP/7. (4K mots de 18 bits). Codé en P. Jeu de mots avec Eunuque car sa femme était partie.
- 1970 : IBM vend
- 1971 : IBM lance les disquettes.
- 1971 : Intel 4004.
- 1971 : Livres grand public.
- 1971 : Kenbak I. 256 bits de RAM. Pour 750\$. Se branche sur la TV.
- 1972 : Atari lance Pong.
- 1975 : MITS lance l'Altair.
- 1975 : Ouverture de la première boutique d'ordinateurs.
- 1975 : IBM 5100. Flop.
- 1976 : Z80.
- 1976 : Apple I.
- 1977 : Atari VCS 2600.
- 1977 : Bill Gates arrêté pour avoir grillé un feu rouge.
- 1977 : Apple II.
- 1977 : Apple, Comodore et Tandy commencent à vendre des ordinateurs personnels.
- 1978 : Premier demi-million d'ordinateurs vendus.
- Ouverture de The Source et de Compulab.
- 1980 : Sinclair ZX 80 pour 99\$.
- 1981 : IBM PC 5150. Appel d'offre pour les OS. Prend Microsoft avec QDOS.
- 1981 : Premier ordinateur "portable". 25kg. Osborne 1.
- 1981 : Début des interface manager (ancêtre Windows).
- 1983 : Times déclare l'ordinateur le titre de personnalité de l'année.
- 1983-09-27 : Début de GNU par Richard Stallman. Problème d'imprimante Xerox. Cause de logiciel propriétaire. Développe GNU.
- 1984 : Lancement du Macintosh.
- 1985 : Windows 1.0. Ils ont juste ajouté la couleur.
- 2011 : Watson bat un être humain à Jeopardy.
- 2015 : AlphaGo bat les meilleurs joueurs de Go.
- Chanson "Changing minds" par le groupe 16 bits.
- ENIAC
- The baby : la machine avec laquelle on est allés sur la lune.

— Ferranti pagasus. Kreftwerk. Ils ont inventé la musique électronique.

2 Une histoire des langages de programmation

Gros problème : les langages de programmation sont une tour de Babel.

2.1 Première partie

2.1.1 FORTRAN

John Baccus en a marre de faire de l'octal.

Va voir Watson (IBM, CEO) pour lui demander de faire un langage de programmation. Va voir le fils du CEO qui lui dit oui.

Ils travaillent sur l'IBM 704. Ils font le langage en 2 ans.

Le présente au père. le vend et fait un flop.

Donc open bar avec correction de problèmes gratuite.

Ils inventent une bonne partie des fonctionnalités actuelles.

De très bonnes performances pour le calcul scientifique.

En C, les digraphes et trigraphes, comme en fortran : `.LE.` au lieu de `<=`.

2.1.2 ALGOL

Anciennement AIL.

4 objectifs :

1. Faire un support pour les publications et les revues scientifiques.
2. Doit donc être proche des notations mathématiques.
3. Doit être lisible sans aide
4. Doit être traduisible en code machine.

On a les mêmes droits de partout. Donc compilo pourri et perfs pourries.

A servi pour faire JOVIAL pour programmer des missiles.

A été fait (entre autre) par John Backus, qui a inventé la Backus inner form :

`<stmt> : :=<ifthenelse>|<unsigned>`

On a intérêt à lire cette représentation quand on se lance dans un nouveau langage.

Plus gros problème : pas d'IO standard.

Intéressant : priorités aux fonctions.

2.1.3 COBOL

Fait par Grace Murray Hopper.

Ils ont fait le langage et le compilo en 3 mois. Avec les résultats qu'on attend.

Surtout pour le business.

Imposé par l'armée américaine dans ses contrats.

2.2 Seconde vague

2.2.1 APL

Illisible mais génial. Besoin d'un clavier spécial.

Ancêtre de Go et R.

2.2.2 PL/I

Fourni avec l'IBM 360. Objectif : faire tout.
Et il se démarque avec l'invention de la notion d'exception.
Mais c'est incompatible avec le temps réel.

Easter egg :
Pas de mot-clé réservé. On peut nommer des variables "if"...
Abbreviations non-homogènes...
 $25 + 1/3 = 5.3333333333333333333333333333...$
 $25 + 01/3$ se passe comme prévu...

2.2.3 BASIC

Très populaire.
Il doit s'exécuter jusqu'à omber sur une erreur.

2.2.4 Pascal & Heirs

Fin des années 60.
Boucles repeat et until.
Types énumérés.
Type interval.
Garde la structure de l'ALGOL.
Veut les performances de FORTRAN.

Va donner Ada. Résultat d'un concours du DoD.
Temps réel et exceptions.
Super opportunités en Ada. Très propre.

2.3 The final wave

2.3.1 K. N. King

Juste un opportuniste. Prend des photos de partout.

3 Histoire de l'orienté objet

3.1 Early languages

3.1.1 Simula

Turing awards : Prix Nobel de l'informatique.
Permet de faire des thread pour faire varier la vitesse d'exécution des objets dans une centrale.
Introduit :
Object, class, constructors, inheritance, virtual methods, attribute hiding, etc.

Impact :
Smalltalk

Objective-C, Objective-Pascal, Objective-C++, etc. (Problème : lent, Avantage : on peut avoir des objets sur des machines différentes.)

CLOS (Objectif : dispatch dynamique sur tous les paramètres, basé lisp)

Eiffel

C++

3.1.2 Smalltalk

Langage non révolutionnaire, petite échelle.

Tout est objet.

Simple à comprendre. Objectif ; faire une tablette pour les enfants.

La seule interface est de passer des messages.

Gros problème de performance.

Règles pour décrire le fonctionnement des métaclases de Smalltalk :

1. Tout est objet.
2. Tout est une instance de classe
3. Toute classe possède une super-classe (contrairement à Java, avec Object, par exemple.)
4. Tout fonctionne au travers de passage de messages.
5. Le lookup doit se faire selon la chaîne d'héritage. (Donc pas d'héritage multiple.)
6. Chaque classe est une instance d'une méta-classe. (Donc générateur de méta-classe pour simplifier la vie car on a besoin de travailler sur les deux simultanément).
7. La hiérarchie des méta-classes est parallèle à la hiérarchie des classes.
8. Chaque classe hérite de "Class".
9. Chaque méta-classe est une instance de "Metaclass".
10. La méta-classe de "Metaclass" est une instance de "Metaclass".

Donc en C++ :

class

struct

interface

Intérêt d'une métaclasse : dire comment struct et class se comportent.

Voir Squeak, une implémentation moderne de Smalltalk.

Ajouter le schéma de l'héritage de Squeak.

3.1.3 C++

Par Stroustrup.

Ajouts :

- const
- reference
- Prototypes
- Booleans
- Declaring variables anywhere
- void
- inline
- namespace
- overloading

3.2 Propriété de l'OO

Rica Cardelli : Travaille sur le typage.

Pour lui, un bon langage doit avoir ces caractéristiques :

- Économie de l'exécution
- Économie de compilation
- Économie de développement à petite échelle

- Économie de développement à grande échelle
- Économie de fonctionnalités

4 Paramètres de fonctions

Exemples :

- Valeur
- Référence
- Name (Équivalent remplacement par macro dans une fonction uniquement)

4.1 (N)RVO Return value optimization

```
C++  
g()  
std::vector<int> res;  
return res;
```

Le compilateur ne va pas appeler le constructeur par copie pour des raisons de performance.

En revanche, Ada fait des copies à l'appel et au retour. C'est du "value result".

4.2 Call by name

On remplace toutes les occurrences des variables par les paramètres de l'appel.
Plus personne le fait : on a les template, maintenant.

4.3 Default parameters

Introduit par Ada.

4.4 Subprograms as arguments

```
C++ : lambda, std::function  
C : pointer on function
```

4.5 Conclusion

Lire Conception et programmation objet.
Il parle d'Eiffel.

5 Technologie objet Eiffel

Bertrand Mayer (MIT)

5.1 Introduction

Il y a des classes et des objets.
Closes d'adaptation d'héritage.

5.2 Des concepts originaux

Programmation contractuelle.
3 modes de développement (Optimisation, Débug, Production).

5.3 Heritage

Smalltalk, Java et Ada n'implementent pas l'heritage multiple.

Eiffel propose de renommer les primitives dont on herite. On peut changer la visibilite de ce qu'on herite.

Revoir covariance/contravariance des types de retour/arguments.

6 Garbage Collector

Introduit par LISP avec Mc Carthy.

Types de GC :

- Reference counter : Probleme reference cyclique
- Mark and sweep : resolution probleme cyclique, mais creation et deletion $O(n)$