Chapitre 1
Introduction

- Langages informatiques et de programmation
- Pourquoi étudier les langages de programmation ?
- A quoi servent-ils ?
- Historique
- Classification par paradigmes
- Petite bibliographie

Quelques types de langages informatiques

- Modélisation: UML
- **Spécification formelle**: Z, Réseaux de Petri, Types abstraits algébriques
- **Programmation:** assembleurs, Java, Prolog, Lisp
- Script: DOS, macros dans Excel
- Interrogation/mise à jour de bases de données: SQL
- Description de circuits électroniques: VHDL
- Mise en page: HTML, TEX, PostScript
- Méta-langage: XML, souvent associé à XSL pour le web

☛ Ce cours se concentre sur les langages de programmation de haut niveau, de modélisation et comment en donner une sémantique

Pourquoi étudier les langages ?

Pour comprendre:

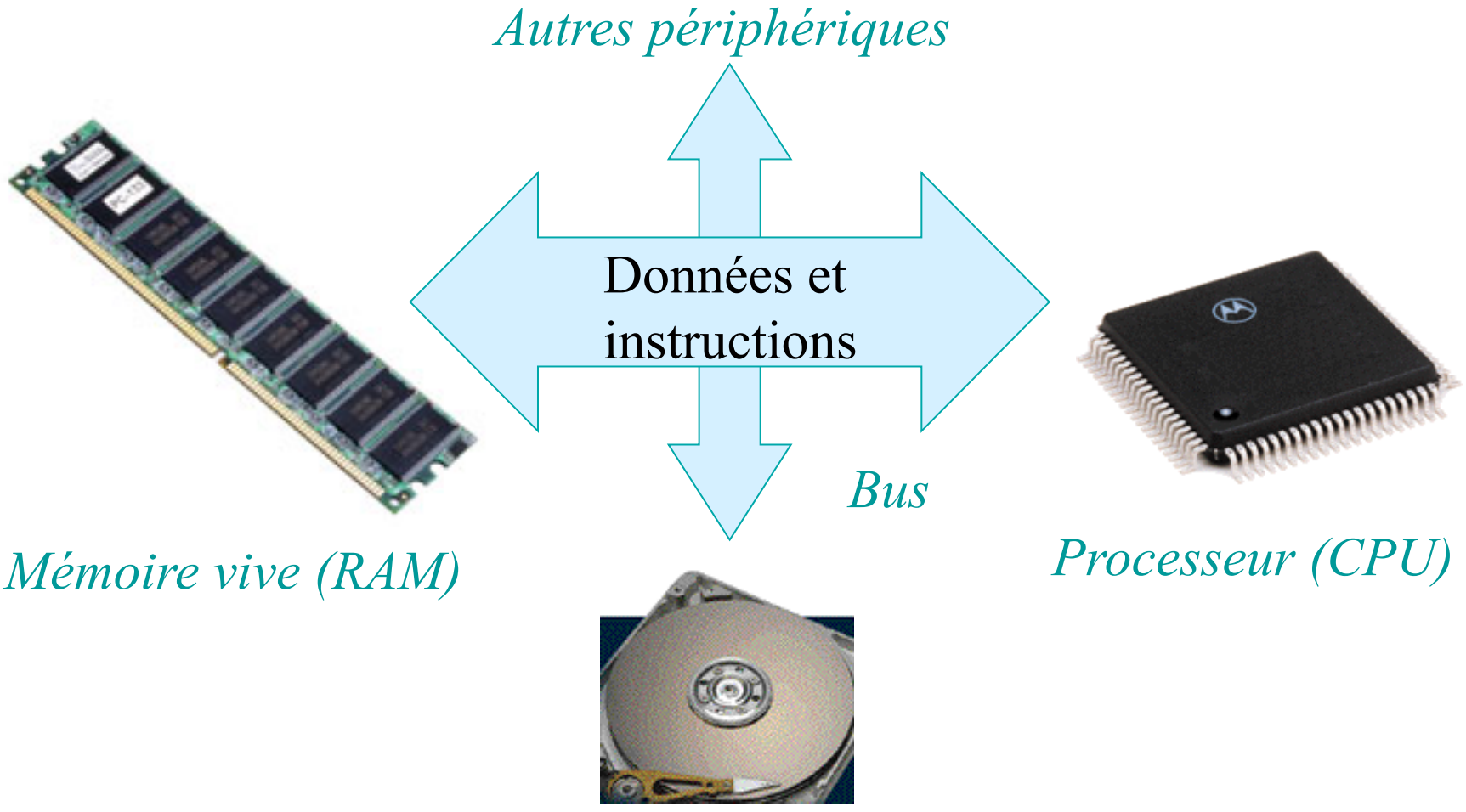
- quel modèle conceptuel est le plus approprié pour traduire un problème du monde réel en un programme
- comment décomposer un programme informatique en éléments plus faciles à gérer
- comment décrire un langage de programmation
- comment un langage devrait être implémenté
- quelles facilités un langage devrait fournir pour simplifier la vie du programmeur (et lui éviter des erreurs)

☛ … et pour avoir le recul nécessaire pour être un vrai professionnel et éviter les guerres de religion à leur sujet !

Pourquoi les langages informatiques ?

- Un langage sert à communiquer
- Un langage informatique doit être précis, non-ambigü
  - Le langage parlé, les textes de lois, les textes sacrés sont truffés d'ambigüités
  - Langage formel = absolument sans ambigüité
- La résolution de problèmes informatiques consiste à:
  - factoriser ce qui est commun à certaines classes de problèmes
  - gérer les variantes, donc les différences entre ces problèmes
- Pour simplifier la tâche du programmeur, les langages de haut niveau cherchent à intégrer ces aspects, par leurs structures de données et de contrôle.

Principe d'exécution d'un programme



Pourquoi des langages de haut niveau ?

- Chaque famille de processeur possède un langage binaire qui lui est propre: le langage machine
  - ```
    00000010101111001010
    00000010111111001000
    00000011001110101000
    ```
- Pour chaque langage machine, on peut avoir une variante symbolique, dite langage d'assembleur
  - ```
    LOAD    I
    ADD     J
    STORE   K
    ```
- … alors qu'il est tellement plus facile et lisible d'utiliser:
  - `k = i + j` (mais il a fallu plus de 10 ans pour y arriver !)

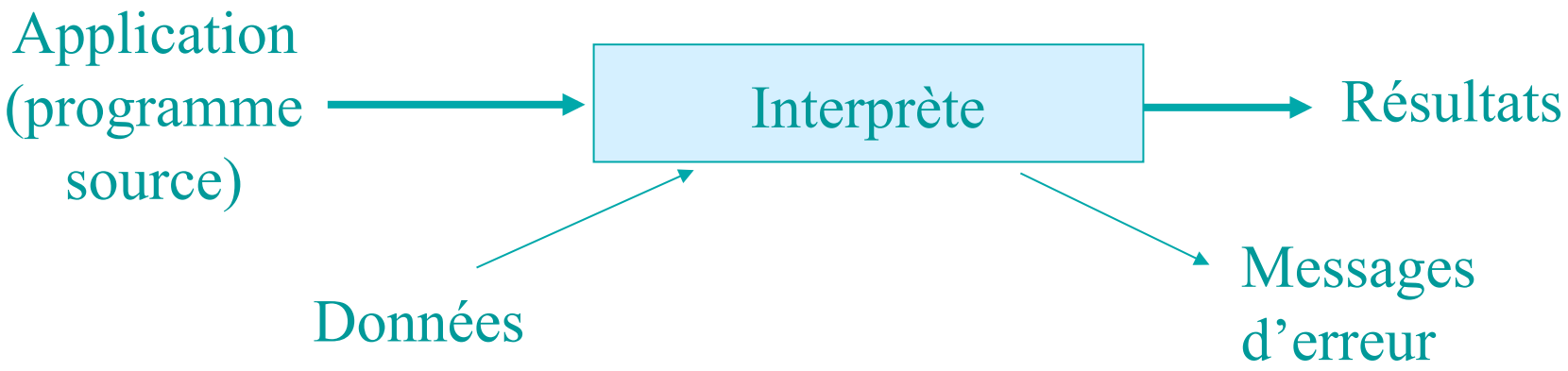
☛ Abstraction = élimination d'erreurs, augmentation de la productivité, programmes plus efficaces, portabilité

Qu'est-ce qu'un langage de programmation ?

- Un *programme* est la spécification d'un calcul (algorithme).
- Un *langage de programmation* est une notation pour écrire des programmes; il permet d'écrire des phrases, formées d'une suite de mots appelés *unités syntaxiques*.
- La *syntaxe* du langage régit la forme des phrases.
  - Pascal correct: If 3+5=9 then write ('bravo')
  - non correct: If 3+5=9 then write 'bravo'
- La *sémantique* du langage est la signification véhiculée par les phrases valides du langage (sémantique statique ici, ou typage).
  - Non correct: if 3=false then write ('bravo')

Interprètes

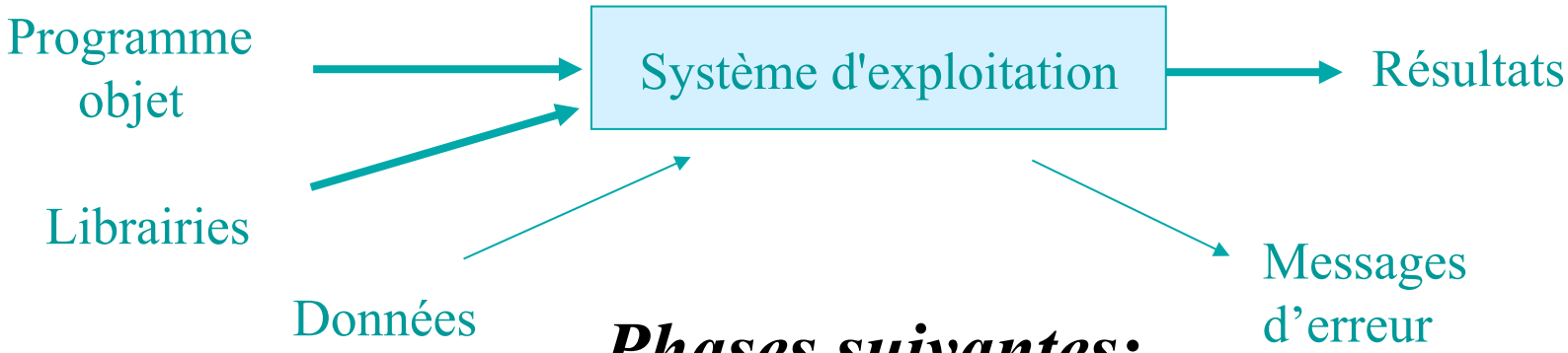
- Un *interprète* (ou *interpréteur*) lit, puis exécute immédiatement le texte source



Compilateurs

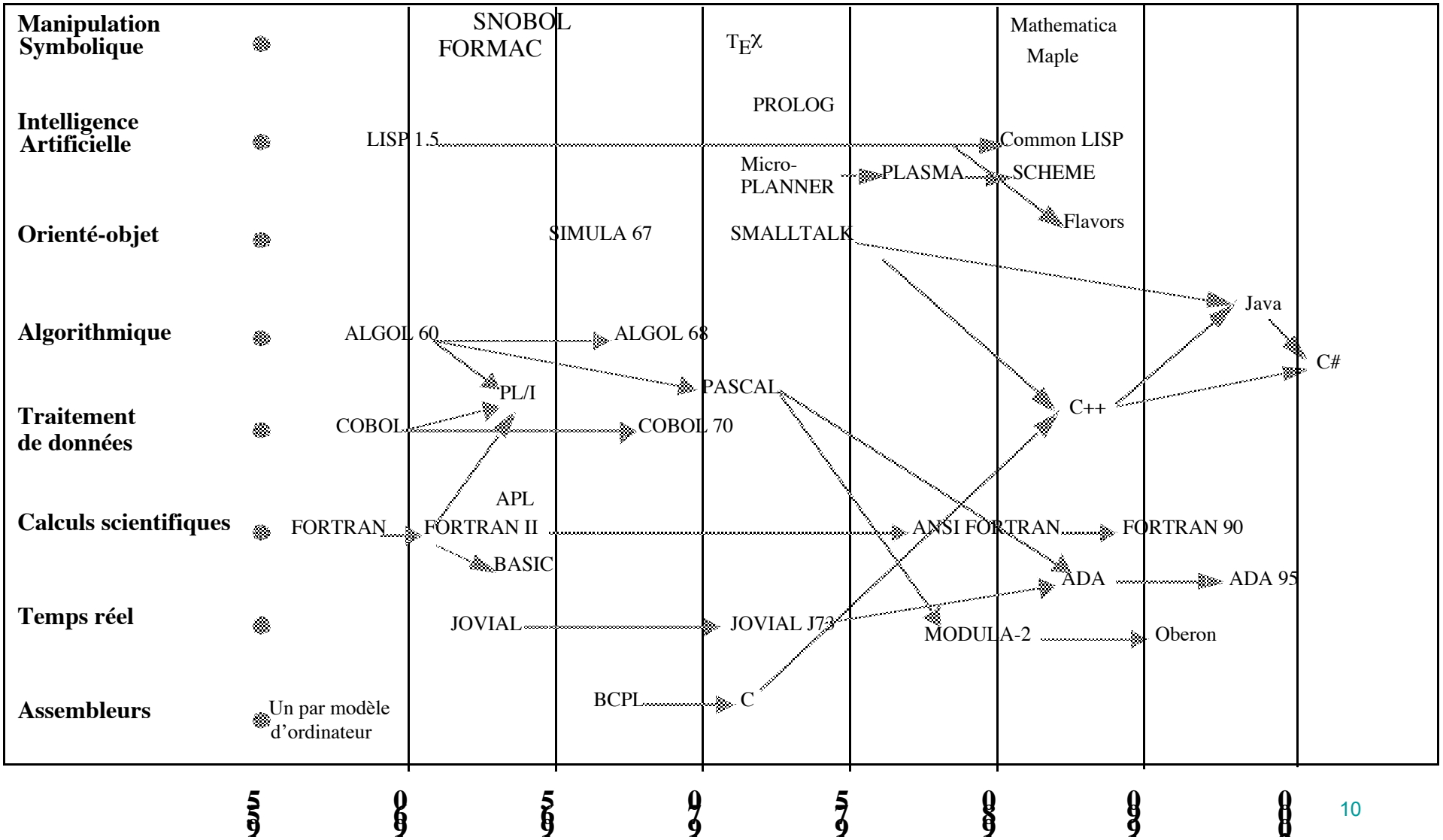
- Un *compilateur* prend un texte source et le traduit dans un langage cible; le résultat est un fichier *objet*, dont l'exécution est remise à plus tard

Phase 1: Compilation

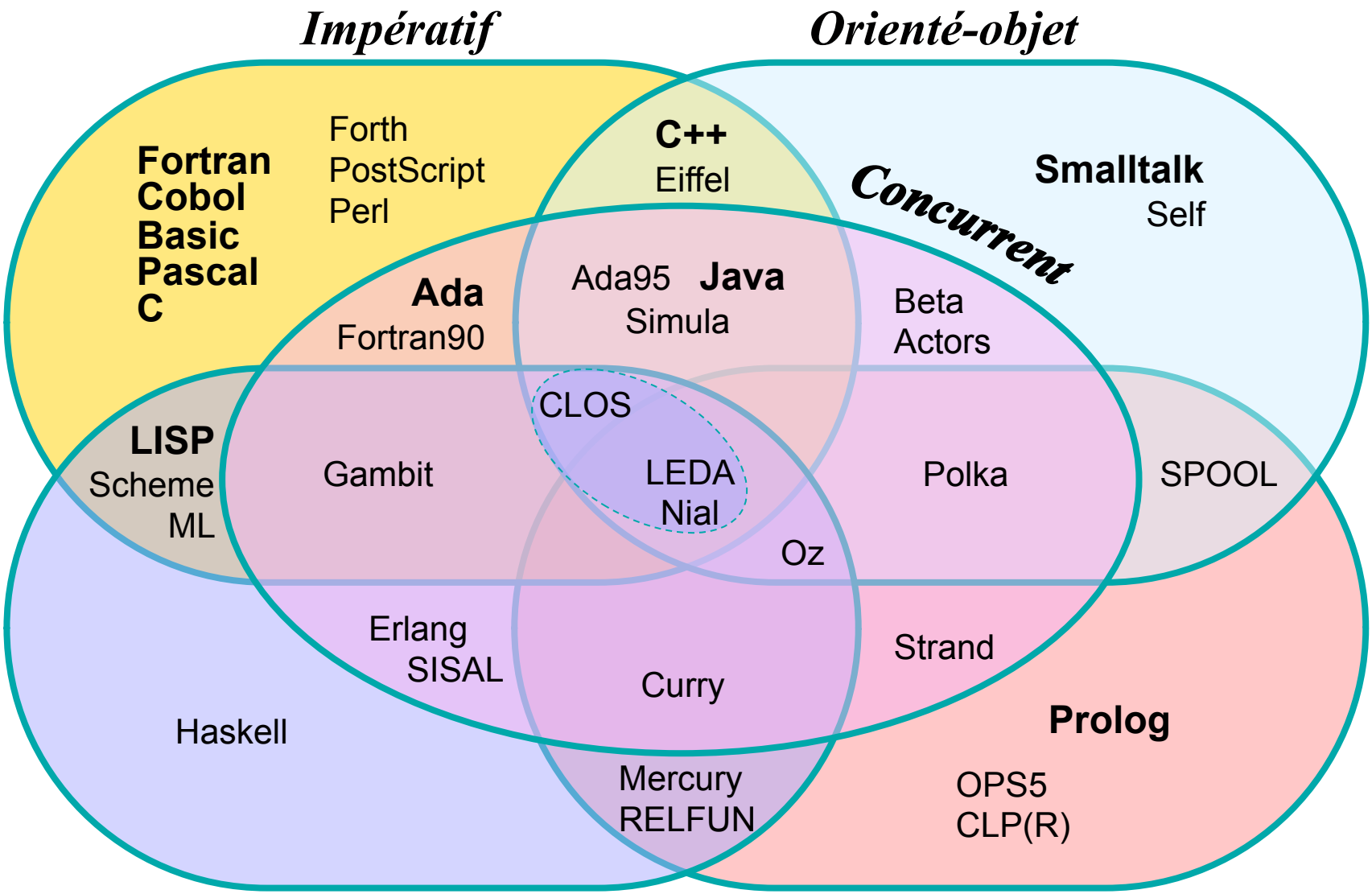


Phases suivantes: Edition de liens & exécution

Petit historique des langages



Panorama des paradigmes courants



Un peu de culture divertissante …

- Une perspective du futur des langages

  http://channel9.msdn.com/Blogs/pdc2008/TL57

- 99 bottles of beer (comparaison humoristique de langages de programmation)

  http://www.99-bottles-of-beer.net

Programme du cours

- Introduction
- Programmation logique
- Sémantique des langages
- Langages spécifiques au domaine (DSL)
- Outils formels de modélisation

Programme, description des objectifs

- Programmation logique
  présentation rapide de prolog, un langage logique utile pour réaliser des exemples et expérimenter la sémantique des langages.
- Sémantique des langages
  Nous présenterons l'intérêt de définir une sémantique et différent moyen d'y parvenir. Ce sont les sémantique opérationnelles, axiomatique, transformationnelles et dénotationelles. Une présentation basée sur des exemples autour de langages très simples sera faite pour différentes sortes de sémantiques opérationnelles.

Programme, description des objectifs

- Langages spécifiques au domaine (DSL)
  En pratique définir son propre langage est souvent nécessaire pour exprimer les problèmes spécifiques à un domaine (comme par exemple HTML, unix command line).
  Nous verrons comment donner une sémantique à ces langages en terme de transformation en un langage abstrait de haut niveau ou un langage formel.

Bibliographie générale

- *Concepts in Programming Languages,* John C. Mitchell, Cambridge Univ Press, 2002, 450 pages, ISBN 0521780985
- *Programming Language Pragmatics*, Michael L. Scott, Morgan Kaufmann Publishers, October 1999, 880 pages, ISBN 1-55860-442-1
- *Comparative Programming Languages*, Robert Clark and Leslie B. Wilson, Addison-Wesley, 3è édition, 2000, 384 pages, ISBN 0-201-71012-9
- *Programming Languages: Structures and Models*, Herbert L. Dershem and Michael J. Jipping, PWS Publishing Co, 1995, 432 pages, ISBN 0-534-94740-9
- *Concepts of Programming Languages*, Robert W. Sebesta, Benjamin/ Cummings, 1989, 500 pages, ISBN 0-8053-7011-0