

# Introduction à l'informatique et la programmation

## IF1 2013-2014

Matthieu Picantin



LIAFA UMR 7089  
CNRS & Université Paris 7 Denis Diderot

09/09/2013

### Tests et examens

- ♦ Td : résultat des 2 tests effectués en Cours-TD
- ♦ Tp : résultat des 2 tests effectués en TP
- ♦ E0 : partiel (samedi 26 octobre)
- ♦ E1 : examen fin décembre ou début janvier
- ♦ E2 : examen en juin

### Notes finales

- ♦ Contrôle continu :  $Cc = (Td + Tp) / 2$
- ♦ Note d'écrit :  $Ne = \max(E1, (E0 + E1) / 2)$
- ♦ Note session 1 :  $(3Ne + Cc) / 4$
- ♦ Note session 2 :  $\max(E2, (3E2 + Cc) / 4)$

### Rappel

pas de note  $\Rightarrow$  pas de moyenne  $\Rightarrow$  pas de semestre

### Organisation générale

**Cours** le jeudi de 14h30 à 16h30 en amphi 1A

MATH+MASS	12/09	19/09	17/10	31/10	14/11	28/11
INFO+MI+CPEI+LI	09/09	26/09	24/10	07/11	21/11	05/12

### Cours-TD

- ▶ 2 heures chaque semaine
- ▶ début la semaine du 16/09

### TP

- ▶ 3 heures chaque semaine
  - ▶ début la semaine du 16/09
- première semaine: 2 x 3 heures d'IS1 pour les groupes INFO et MI  
dernière semaine: 2 x 3 heures d'IF1 pour les groupes INFO et MI

### Tutorat

- ▶ le midi au SCRIPT

### Commission de suivi

- ▶ sur rendez-vous au DSE

### Programme

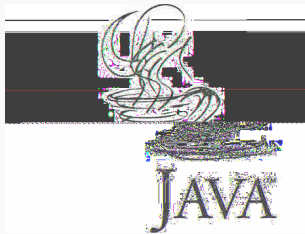
- ♦ notion d'algorithme et son expression en langue naturelle
- ♦ langages de programmation (styles de programmation, compilation/interprétation, code binaire ou code intermédiaire)
- ♦ variables, identificateurs, concept de type
- ♦ affectation, expressions (arithmétiques, booléennes) et leur évaluation
- ♦ structures de contrôle : sélections, itérations
- ♦ tableaux à une ou plusieurs dimensions et imbrications de boucles
- ♦ concept de fonction et transmission de paramètres
- ♦ rapide introduction au concept d'objet

### Objectifs

- ♦ comprendre un certain nombre des concepts généraux des machines et de la programmation
- ♦ réaliser le codage effectif, la compilation et l'exécution d'algorithmes simples dans un environnement de type Unix

## Programme

- ♦ notion d'algorithme et son expression en langue naturelle
- ♦ langages de programmation (styles de programmation, compilation/interprétation, code binaire ou code intermédiaire)
- ♦ variables, identificateurs, concept de type
- ♦ affectation, expressions (arithmétiques, booléennes) et leur évaluation
- ♦ structures de contrôle : sélections, itérations
- ♦ tableaux à une ou plusieurs dimensions et imbrications de boucles
- ♦ concept de fonction et transmission de paramètres
- ♦ rapide introduction au concept d'objet



## Plan possible du cours en amphi

- #1 modèles et algorithmes
- #2 langages de programmation
- #3 compilation et typage
- #4 logique et circuits
- #5 correction et sécurité

## Qu'est ce que l'algorithmique ?

- ♦ Avez-vous déjà ouvert un livre de cuisine pour concocter une bonne ouiche lorraine ?
- ♦ Avez-vous déjà déchiffré une notice de montage d'une armoire suédoise ?
  - ▶ Si oui, sans forcément le savoir, vous avez déjà exécuté des *algorithmes* !
- ♦ Avez-vous déjà indiqué son chemin à un touriste égaré ?
- ♦ Avez-vous déjà écrit une lettre anonyme précisant comment procéder à une remise de rançon ?
  - ▶ Si oui, vous avez déjà fabriqué—et fait exécuter—des algorithmes !

## L'art d'organiser un calcul complexe en partant d'opérations simples

une suite d'instructions compréhensibles par celui qui devra l'exécuter

## Étymologie

Le mot *algorithme* vient du nom du mathématicien et astronome perse Al Khuwarizmi, qui écrivit au IX<sup>e</sup> siècle un traité donnant la première solution systématique des équations quadratiques

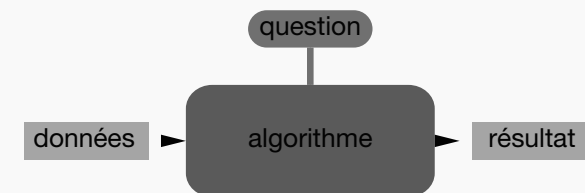
## Bibliographie

- ♦ Concepts fondamentaux de l'informatique  
Alfred Aho & Jeffrey Ullman (une dizaine d'exemplaires à la bu 004 AHO)
- ♦ Concrete Mathematics: A Foundation for Computer Science  
Ronald Graham, Donald Knuth & Oren Patashnik
- ♦ Penser, modéliser et maîtriser le calcul informatique  
Gérard Berry
- ♦ The Art of Computer Programming  
Donald Knuth
- ♦ The Java Tutorial <http://java.sun.com/docs/books/tutorial/>  
Mary Campione, Kathy Walrath & Alison Hulm
- ♦ Thinking in Java <http://www.mindview.net/Books/TIJ/>  
Bruce Eckel

## Informations relatives à l'enseignement

<http://didel.script.univ-paris-diderot.fr>

## L'art d'organiser un calcul complexe en partant d'opérations simples



## Instructions

- ♦ affectation
- ♦ lecture/écriture
- ♦ tests
- ♦ boucles

## Principes

- ♦ approche incrémentale
- ♦ principe du diviser pour régner
- ♦ technique gloutonne
- ♦ exploitation de l'aléa

- ♦ le type de machine
- ♦ le temps de calcul
- ♦ la taille mémoire
- ♦ la consommation d'énergie
- ♦ le type de résultat

## images

texture éclairage  
fusion mise 3D  
contours segmentation  
corrections optiques

## géométrie

trajectoires  
déformations  
tomographie  
surfaces volumes

## réseaux

gestion de trafic  
diffusion  
protocoles  
codage

## mots, textes

grammaires, automates  
croisement  
tri, recherche  
classement

## nombres

éléments finis  
matrices prog. linéaire  
premiers cryptage  
4 opérations

## optimisation

emploi du temps  
ordonnancement  
circulation  
routage

- ♦ algorithmes de tri
  - ▶ tri par insertion
  - ▶ tri par dichotomie-fusion
  - ▶ tri rapide
- ♦ algorithmes arithmétiques
  - ▶ addition et multiplication
  - ▶ pgcd et division euclidienne
- ♦ algorithmes géométriques
  - ▶ enveloppe convexe
  - ▶ diagramme de Voronoï
  - ▶ triangulation de Delaunay
  - ▶ arbre recouvrant
- ♦ algorithmes stochastiques
  - ▶ chifoumi
  - ▶ calcul de  $\pi$
- ♦ modèles abstraits
  - ▶ machine de Turing
  - ▶ Jeu de la Vie

### Définition

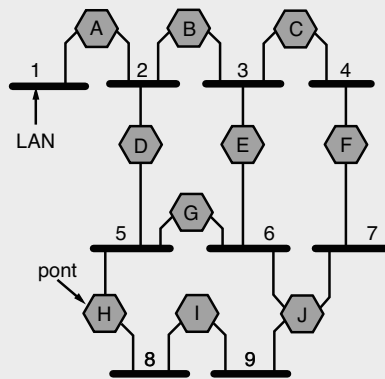
L'algorithme d'Euclide est un algorithme permettant de déterminer le plus grand commun diviseur (pgcd) de deux entiers dont on ne connaît pas la factorisation. Il est déjà décrit dans le livre VII des Éléments d'Euclide.

### Pseudo-code

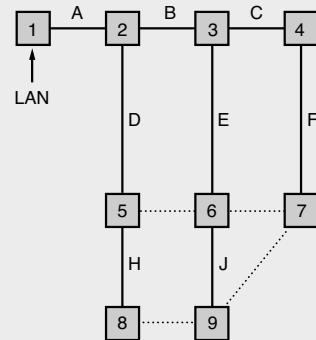
---

Propriété

## Une interconnexion



## Un arbre recouvrant



## Plusieurs usages de l'aléatoire