Tronc Commun X2005

INF311 : Introduction à l'Informatique

F. Morain

morain@lix.polytechnique.fr

DÉPARTEMENT D'INFORMATIQUE

Amphi 1 : introduction à Java

I. Organisation générale du cours.

II. Un ordinateur, comment ça marche?

III. La programmation en Java.

IV. Au travail!

V. Derniers mots.

I. Organisation générale du cours

Cours fait pour les **débutants** en informatique

- Faire connaissance avec l'informatique (science + techniques).
- Se familiariser avec un système d'exploitation (Unix), quelques outils (`kde`, `mozilla`, mail, `nedit`, etc.).
- Apprendre la programmation, vérifier la correction des programmes, les valider par des tests. Le langage support est Java.

Quels moyens?

- **10 blocs :** 1 bloc = 1 amphi de 1h30 suivi de TP de 2h le jour même ou le lendemain ; à compléter par deux heures de travail personnel.
- **Tutorat:** encadrement disponible pour les 2h de travail perso pour aider à régler les bloquages éventuels.
- Tout est sur le web: poly, exercices, etc.
- **Matériel :** stations de travail PC sous GNU/Linux (kde).

Les participants

L'équipe enseignante: 1 Chargé d'enseignement + 1 vacataire + 1 pour aider pendant les trois premiers blocs.

1-7	T. Clausen + M. Cluzeau + C.-P. Gwiggner
2-8	É. Duris + T. Houtmann + O. Bouissou
3-9	A. Enge + Q. Thai + F. Kirchner
4-10	F. Magniez + S. Ben Mokhtar + C. Lauradoux
5-11	E. Waller + A. Ziegler + J.-R. Reinhard
6-12	P. Chassignet + M. Quisquater + B. Cautis

Délégués :

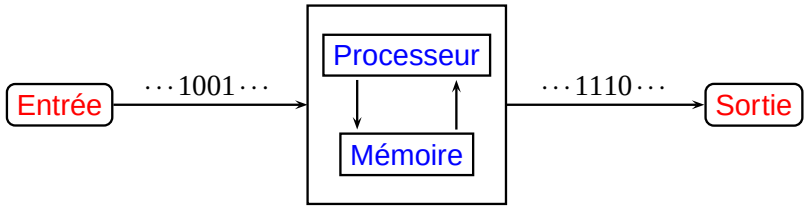
- Un délégué(e) pour chaque groupe avec parité; un délégué et une déléguée pour le niveau élu(e)s parmi ceux-ci.
- Pour quoi faire? Faire remonter les problèmes; sondage; réunions de département; déjeuner de temps en temps.
- **Je veux les noms mardi prochain...**

Notation

- bloc 5 : annotation des programmes par les enseignants ;
- bloc ? ? : composition sur machines ;
- composition Hors Classement le mardi 11 juillet sur papier.

La notation du module se fait sur ces deux notes + une note de participation.

II. Un ordinateur, comment ça marche?

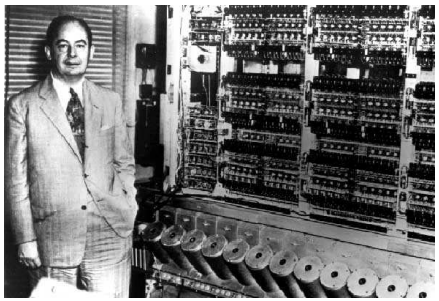


Un ordinateur est une machine qui exécute des programmes qui résolvent un problème.

À un très bas niveau, le processeur transforme des 0 et des 1 en d'autres 0 et d'autres 1. Leur interprétation se fait suivant des conventions précises ($1011000 =$ 'X').

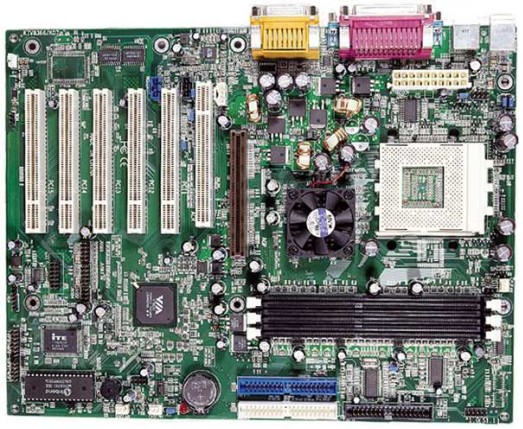
Les pionniers

Alan Turing
1912 – 1954



John von Neumann
1903 – 1957

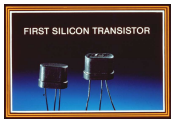
Une carte PC



Réponse à Claude Allègre

Le langage binaire est longtemps resté l'apanage de quelques mathématiciens, le commun des mortels préférant le calcul à base dix, plus économe de chiffres et qui permet d'utiliser ses doigts pour compter. Comment a-t-il fait pour s'imposer comme langage universel des ordinateurs, alors qu'il est si encombrant ?

LA DÉFAITE DE PLATON OU LA SCIENCE DU XXÈME SIÈCLE, Fayard 1995. Chapitre "0-1...et l'infini"

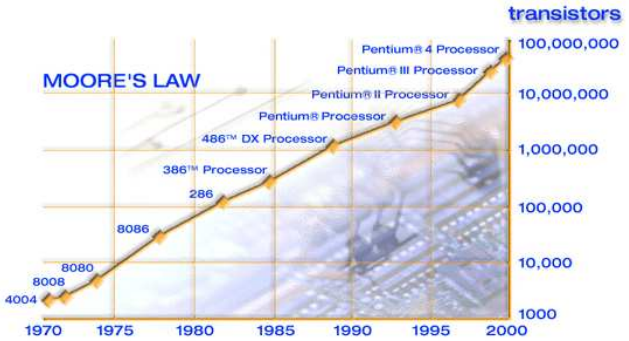


La machine la plus rapide du moment

On mesure la puissance d'un ordinateur par le nombre d'instructions qu'il peut exécuter par seconde (Mips, flop/s).
Cf. la page des top 500.

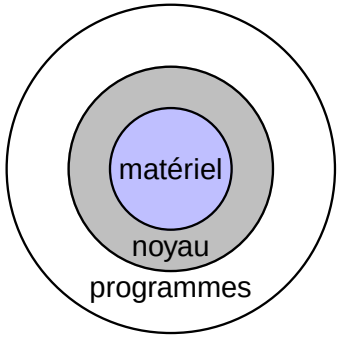
La plus grosse machine: IBM BlueGene/L (131072 processeurs, ≈ 280.6 Teraflop/s sur le benchmark LINPACK – factorisation LU avec pivot partiel d'un système linéaire dense).

La loi de Moore



Le système d'exploitation

Un intermédiaire nécessaire entre le processeur et l'utilisateur.



Unix

Un système multi-utilisateurs en temps partagé.

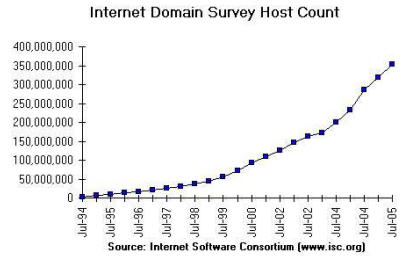
Bienvenue dans un monde dans lequel les ordinateurs ne s'arrêtent jamais et . . . ne doivent jamais être arrêtés.

- mémoire, temps de calcul : plusieurs programmes tournent en même temps ;
- la mémoire est protégée : impossible de planter l'ordinateur !
- découpage des disques en zones séparées par compagnie, puis utilisateur (structure arborescente) ; un élève `toto` a comme répertoire (homedir) `/users/A1/toto` ; ce répertoire est protégé.

Des ordinateurs reliés entre eux

via Internet (réseaux locaux ou globaux).

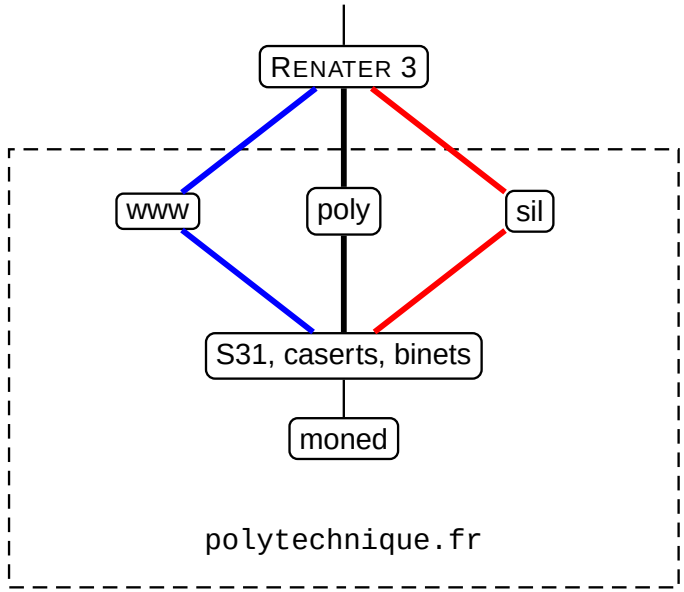
Nombre de machines routables :



Domaine `.fr` : au 23 mars 2006, 456918 sous-domaines répertoriés par l'AFNIC (resp. 346126 au 25 mars 2005).

Comment ça marche? chaque ordinateur dispose d'un numéro IP qui permet de le localiser de par le vaste monde (`sil` a pour adresse IP `129.104.247.3`).

L'environnement réseau à l'X



Comment parle-t-on à un ordinateur?

Java	**a+b**
langage assembleur	**add A, B**
langage machine	**1000110010100000**

Historiquement: apparition des langages dans l'ordre inverse.

Étape ultime : l'ordinateur qui comprend le langage humain directement . . . ou inversement?

Qu'est-ce qu'un calcul ?

Modèle de base : séquentiel – les instructions sont exécutées l'une après l'autre), calculs internes sur des entiers:

$$\begin{array}{rl} & 00000000101000010000000000011000 \\ \rightarrow & 00000000100011100001100000100001 \\ & 10001100011000100000000000000000 \end{array}$$

Algorithmes: déterministes (311/421) ou non (431/majeures) ; probabilistes/randomisés.

Autres modèles : vectoriels, parallèles (SIMD, MIMD), distribués.

Modèles du futur : quantique, biologique, etc.

III. La programmation en Java

A) Qu'est-ce qu'un programme ?

C'est la traduction dans un langage compréhensible pour l'ordinateur d'un ensemble d'algorithmes qui permettent de résoudre des problèmes.

Pourquoi programmer ?

- point commun à tous les informaticiens ;
- résoudre des problèmes toujours nouveaux (e.g., réseaux) ;
- informatique expérimentale.

Pourquoi Java ?

- relativement simple, très répandu, conçu pour le WWW ;
- typé, avec des objets *(object oriented)*, gestion mémoire ;
- quand on connaît un langage, il est facile d'en apprendre d'autres.

Pierre François Xavier Bouchard (X96)

À quoi ressemble un programme Java ?

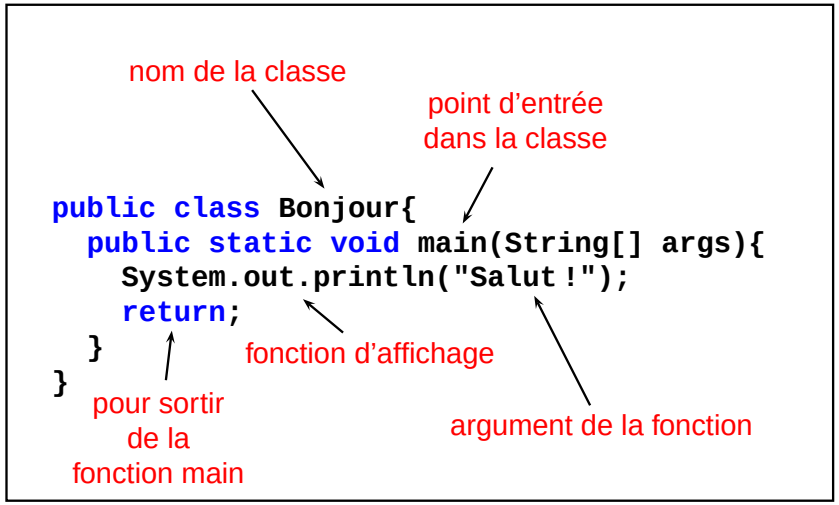
```
public class Wagon{
  final static int WMAX = 100; // constante
  static int n; // variable de classe
  String nom;   // champ d'un objet
  // une fonction (méthode de classe)
  public static void print(Wagon w){
    System.out.println(w.nom);
  }
  // fonction principale
  public static void main(String[] args){
    Wagon w = new Wagon();
    w.nom = "Thalis";
    print(w);
  }
}
```


Le premier programme

```
public class Bonjour{
  public static void main(String[] args){
    System.out.println("Salut !");
    return;
  }
}
```

Mots en bleu : mots-clefs de Java

Le premier programme (suite)



Compilation, interprétation

On écrit le programme dans un fichier `Bonjour.java`, puis on le compile par :

```
unix% javac Bonjour.java
```

et on l'exécute avec :

```
unix% java Bonjour
```

En vrai : `javac` vérifie la syntaxe et le typage, puis traduit le programme dans un langage intermédiaire (byte-code portable) qui est ensuite traduit pour l'ordinateur par `java`.

B) Constituants de base du langage

- Identificateurs : séquences de lettres et de chiffres commençant par une lettre, séparées par des espaces, caractères de tabulation, retours à la ligne ou des caractères spéciaux (**`+`**, **`-`**, **`*`**, etc.).
- mots-clefs : **`class`**, **`public`**, **`static`**, etc.
- types primitifs: entiers (**`int`**), réels (**`double`**), caractères (**`char`**).
- Opérations arithmétiques: **`+, -, *, /, %`** (modulo).
- Bibliothèque mathématique : **`Math.sqrt()`**, **`Math.PI`**, etc.


```
public class Calculs{
  public static void main(String[] args){
    System.out.println(2 + 2);
    System.out.println((13 * 2) % 5);
    System.out.println(2. * Math.PI * 1.5);
    return;
  }
}
```

```
unix% java Calculs
4
1
9.4247779607693793
```


Variables

Une variable est représentée par un identificateur, elle a un type (c'est-à-dire qu'on doit savoir dans quel ensemble elle prend ses valeurs).

On peut utiliser sa valeur, lui en affecter une (nouvelle).

Une variable doit toujours être déclarée :

```
int x;
double u;
```

C) Affectation

```
u = e;
```

- u est une variable et e une expression ;
- les deux ont même type ;
- l'expression e est évaluée, puis la variable u prend pour valeur le résultat de cette évaluation : on a affecté la valeur de e à u.

```
x = -155;
z = x + 3;
```

Une variable doit toujours être initialisée.

On peut condenser déclaration et initialisation :

```
int x = -155;
```

Une instruction idiomatique : avec le mécanisme d'affection, les instructions :

```
int i = 3;
i = i + 1;
```

sont valides, car on calcule d'abord **i+1** et le résultat 4 est mis dans **i** à la place de 3.

D) Un peu de convivialité

Chaînes de caractères : pour rendre un programme plus convivial, on affiche des chaînes de caractères, comme `"Bonjour"`.

```
public class Calculs2{
  public static void main(String[] args){
    int n = 10, b;

    b = n*(n-1)/2;
    System.out.print("résultat=");
    System.out.println(b);
    return;
  }
}
```


Intéractions avec le programme

Ex: demander un entier à l'utilisateur et afficher le carré de l'entier à l'écran.

```
public class Calculs{
  public static void main(String[] args){
    System.out.print("Entrer n=");
    int n = TC.lireInt();
    System.out.print("n^2=");
    System.out.println(n * n);
    return;
  }
}
```

La classe TC: écrite spécialement pour le Tronc Commun. C'est une classe comme **System**. Documentation: poly ou web.

E) Quelques types primitifs

a) les entiers

Un entier $n \geqslant 0$ s'écrit en binaire sous la forme:

$$n = b_p 2^p + b_{p-1} 2^{p-1} + \cdots + b_0$$

avec $p \geqslant 0$, $b_i \in \{0,1\}$.

Ex. $29 = 1\cdot 2^4 + 1\cdot 2^3 + 1\cdot 2^2 + 0\cdot 2^1 + 1\cdot 2^0 = \overline{11101}_2$.

Les entiers en Java: `int` (le plus courant, 32 bits), **`long`** (64 bits), **`short`** (16 bits). On choisit en fonction du problème.

Arithmétique signée

Le type **`int`** permet de coder les entiers de $[-2^{31}, 2^{31}-1] = [-2147483648, 2147483647]$.

$$-2^{31} \leqslant x < 2^{31} \to b_{31} b_{30} \ldots b_0$$

avec $b_i \in \{0,1\}$, et le bit b_{31} est interprété comme bit de signe:

- si $0 \leqslant x < 2^{31}$, $b_{31} = 0$ et

$$x = b_{30} 2^{30} + b_{29} 2^{29} + \cdots + b_0.$$

- si $-2^{31} \leqslant x < 0$, $b_{31} = 1$ et

$$x + 2^{31} = b_{30} 2^{30} + b_{29} 2^{29} + \cdots + b_0.$$

$$+ \neq +$$

Mathématiquement: si **`x`** et **`y`** sont de type **`int`**, donc éléments de $[-2^{31}, 2^{31}-1]$, alors **`x+y`** $\overset{\text{déf}}{=} x + y \bmod 2^{32}$.

Donc: **`2000000000+2000000000 = -294967296`**.

Pas très grave pour des indices de boucle, etc. En cas de besoin, utiliser des **`long`**
$[-2^{63}, 2^{63}-1] = [-9223372036854775808, 9223372036854775807]$.

Opérations classiques

`+, -, *, /, %` (modulo).
Ex. **`int q = 5 / 2; int r = 5 % 2;`**.

Décalages: `n « e` multiplie **`n`** par 2^e, **`n » e`** divise **`n`** par 2^e:

$$0 b_{30} \ldots b_0 << e \to b_{31-e} b_{30-e} \ldots b_0 \underbrace{0 \ldots 0}_{e \text{ zéros}}$$

$$0 b_{30} \ldots b_0 >> e \to \underbrace{0 \ldots 0}_{e \text{ zéros}} 0 b_{30} \ldots b_e$$

et **`(-n) « e = - (n « e)`**, etc.

Instructions compactes

`i = i+1;` ou `i += 1;` (idem pour `-, *, /`).
`i++;` ou `++i;` (idem avec $-$).

Post-incrémentation

```
n = 5;
m = n++;
```

```
m = n;
n += 1;
```

Pré-incrémentation

```
n = 5;
m = ++n;
```

```
n += 1;
m = n;
```


b) Booléens

Déf. un **`boolean`** a pour valeur **`true`** ou **`false`**.

Expressions logiques: évaluées à **`true`** ou **`false`**, combinaisons d'opérateurs de comparaison arithmétiques:
`<, <=, >, >=, ==, !=`
et d'opérateurs logiques: `!` (négation), `&&` (et), `||` (ou).

```
x > 8;
x == 7;
y != ((x == 1) && (z > 9));
```

Attention à la différence entre **`=`** affectation, et **`==`** comparaison.

Tables de vérité

NOT	
V	F
F	V

`!x`

ET		
	V	F
V	V	F
F	F	F

`x && y`

OU		
	V	F
V	V	V
F	V	F

`x || y`

XOR		
	V	F
V	F	V
F	V	F

`x ^ y`

Rem. prolongation aux entiers, en opérant bit à bit:

$$11 \& 10 = 10, \quad 01 | 00 = 01$$

Règles d'évaluation: soient **`C1`** et **`C2`** deux expressions booléennes, dans l'instruction

```
boolean b = (C1 || C2);
```

Java évalue **`C1`**; si **`C1`** est vraie, **`C2`** n'est pas évaluée et **`b`** vaut **`true`**.

De même, dans:

```
boolean b = (C1 && C2);
```

si **`C1`** est faux, **`C2`** n'est pas évaluée, et **`b`** vaut **`false`**.

F) Instructions conditionnelles

But: faire prendre des décisions simples par l'ordinateur.

Syntaxe:

```
if(E)
  I
else
  J
```

I et **J** sont des blocs d'instructions, **E** une expression logique.

J peut ne pas être présente, on écrit alors simplement:

```
if(E)
  I
```



```
public class Ifelse{
  public static void main(String[] args){
    int devinette = 3, n;

    n = 4;
    if(n == devinette){
      System.out.println("Gagné");
    }
    else{
      System.out.println("Perdu");
    }
    return;
  }
}
```



```
public class Ifelse{
  public static void main(String[] args){
    int devinette = 3, n;

    n = 4;
    if(n == devinette)
      System.out.println("Gagné");
    else
      System.out.println("Perdu");
    return;
  }
}
```

Vivement recommandé quand il y a une seule instruction dans le bloc.

Plus intéressant:

```
public class Ifelse{
  public static void main(String[] args){
    int devinette = 3, n;

    System.out.print("Entrer un entier : ");
    n = TC.lireInt();
    if(n == devinette)
      System.out.println("Gagné");
    else
      System.out.println("Perdu");
    return;
  }
}
```

Comment ça marche dans le processeur ?

```
      011  ···
→     012  si n == 3 aller à la ligne 20
      013  ···
      ···  ···
      020  suite des instructions
```

```
      011  ···
      012  si n == 3 aller à la ligne 20
      013  ···
      ···  ···
→     020  suite des instructions
```


IV. Au travail!

Chaque élève a **trois comptes** :

- `moned` : serveur de fichiers contenant les répertoires utilisateur ;
- `poly` : courrier électronique ;
- `sil` : entrer ou sortir du campus (par `slogin`).

À quoi servent ces comptes ?

- travailler pour l'informatique et les autres matières, depuis les salles info ou sa chambre ;
- web, email.

Mots de passe

- ne pas oublier la feuille pour aller en TP.
- **Ne jamais oublier ses mots de passe ou les donner à quelqu'un.**

V. Derniers mots

Ordinateur, etc.

Résumé de l'introduction à Java :

- identificateurs, variables ;
- types primitifs : **`int`**, **`boolean`**;
- premiers programmes.

Prochains rendez-vous

Groupes	TD 1
1–6	10h15–12h15
7–12	13h30–15h30

Élection des délégué(e)s.

Prochain amphi: mardi 2 mai à 10h30.

Tutorat: en salles info le mercredi 3 mai. Inscription à la scola à la sortie des urnes.

Où sont les salles info ?

- Au rdc, à côté du couloir des langues.
- Pour y aller :
  - passer à côté de la scola, prendre le couloir d'accès aux laboratoires ;
  - descendre le premier escalier sur votre gauche ;
  - arrivé(e) en bas, passer dessous, c'est la première porte à droite après les portes vitrées ; elle s'ouvre avec le badge ;
  - les noms des élèves sont affi chés sur les portes des salles.