INFORMATIQUE
Sciences
Université Paris Cité

Introduction à la Programmation

Langage: Java

IP1 Java

Lélia Blin

lelia.blin@irif.fr

Septembre 2023

But du cours

> Apprendre les bases de la programmation
> Être capable de comprendre des programmes
> Être capable d'écrire des programmes simples
> Langage utilisé : Java

Apport du cours

Savoir programmer et comprendre l'approche de la programmation est un atout important

Les clés pour réussir

- Travail régulier nécessaire
  - N'hésitez pas à refaire les exercices chez vous
- Ne pas rater de cours-Td et de TP
- Besoin de beaucoup de rigueur dans l'écriture des programmes

Remarque : Cours sans difficulté théorique

Organisation des enseignements

- Cours de présentation (2h)
- **Toutes les semaines (6 h)**
  - Séance de 2h : Cours/Travaux dirigés (Td)
  - Séance de 2h : Travaux pratique 1
  - Séance de 2h : Travaux pratique 2

Important : Respecter votre groupe

Tutorat (si possible)

- Accès libre
- Certains jours entre 12h et 14h (horaires et salles à préciser)
- Début : vous sera indiqué

Évaluation

Contrôle continu (Cc): 2 épreuves en TP

Partiel : 2h, QCM avec questions ouvertes (P)

Examen 2h avec questions ouvertes 5(E)

Note

Ne : Max (E, (E + P) / 2)

Note session 1 : (3*Ne + Cc) / 4

Remarques :

- Absence aux CC : 0
- Absence au partiel : 0
- Absence à l'examen : pas de note
- Une mauvaise note au partiel est rattrapable

Points sur le contenu

- Les bases de la programmation seront présentés en cours/td
    - Des supports vous seront distribués
- Les Tp servent à mettre en pratique ces bases
    - Les énoncés seront sur Moodle

Page Moodle du cours (toutes les infos y sont données) :
https://moodle.u-paris.fr/course/view.php?id=1620

Communication

N'hésitez pas à communiquer avec:

> Vos chargé.e.s de cours/td et tp
>
> Vous pouvez aussi m'écrire : lelia.blin@irif.fr
>
> Nous lisons toutes et tous nos mails régulièrement

Ecriture du mail:

> Respectez les règles de politesse
>
> Précisez le nom du cours et votre groupe
>
> Vérifiez votre orthographe et n'utilisez pas d'abréviation
>
> N'oubliez pas de signer votre mail
>
> > N'envoyer pas un programme tapé dans un mail ou dans un document Word !!!!

Comment travailler vos cours :

- Écrire les programmes sur feuille sans les tester n'est pas suffisant
- Il faut écrire des programmes chez vous ou en salle de TP et tester qu'ils fonctionnent bien
- La voie vers le succès pour ce cours : programmer encore et encore

Programmer

- Pour les TPs, il vous faut un login et un mot de passe pour pouvoir vous connecter aux machines de l'université
- Pour les obtenir :
  - Il faut activer votre compte Université de Paris
    - https://u-paris.fr/activation-de-votre-compte-universite-de- paris/
  - Activer son compte machien :
    - http://comptes.script.univ-paris-diderot.fr/activation_compte/compte_up75.php

Important : FAIRE CELA AVANT LE PREMIER TP !!!!!

Qu'est ce qu'un programme ?

Un programme est une suite d'instructions qui pourra être 'exécutée' par la machine

Quelles sont les instructions disponibles

> Faire un calcul arithmétique (par ex. 12 * 5)
> Afficher une chaîne de caractères
> Déplacer la souris
> Lancer un autre programme
> Manipuler des données
> Jouer un son
> etc

Où trouve-t-on les programmes ?

Tout ce que vous exécutez sur une machine de calcul telle qu'un ordinateur, une tablette ou un smartphone est un programme

> Les applications
> Les logiciels
> Mais aussi le système qui fait fonctionner votre appareil
> - Androis, IOS, Windows, Linux,...

Comment écrit-on un programme ?

- Un programmeur écrit un programme dans un langage de programmation
- Il existe plusieurs langages de programmation et plusieurs familles de langage de programmation
  - Langages Orientée Objets : Java, Python
  - Langage Impératif : C, ...
  - Langage Fonctionnelle : CAML, OCAML, ...

Comment la machine comprend tous les langages ?

- Le langage de programmation est un langage 'compréhensible' par les humains
- Les instructions sont un mélange d'anglais et d'opérations mathématiques, plus certaines instructions spécifiques à chacun des langages
- Le programme écrit par le programmeur est contenu dans un fichier, on parle de code source

Et après le code source

- Le code source est ensuite
  - soit traduit vers un langage compréhensible par la machine (langage binaire), on parle de compilation,
  - soit il est interprété par un interpréteur qui exécute ces instructions (langage interprété)

> Pour pouvoir exécuter un programme, il faut donc soit avoir le compilateur (Java) ou l'interpréteur (Python, OCaml)

> **Remarques** : L'interpréteur et le compilateur sont eux-mêmes des programmes

Schéma d'exécution d'un code source avec compilation

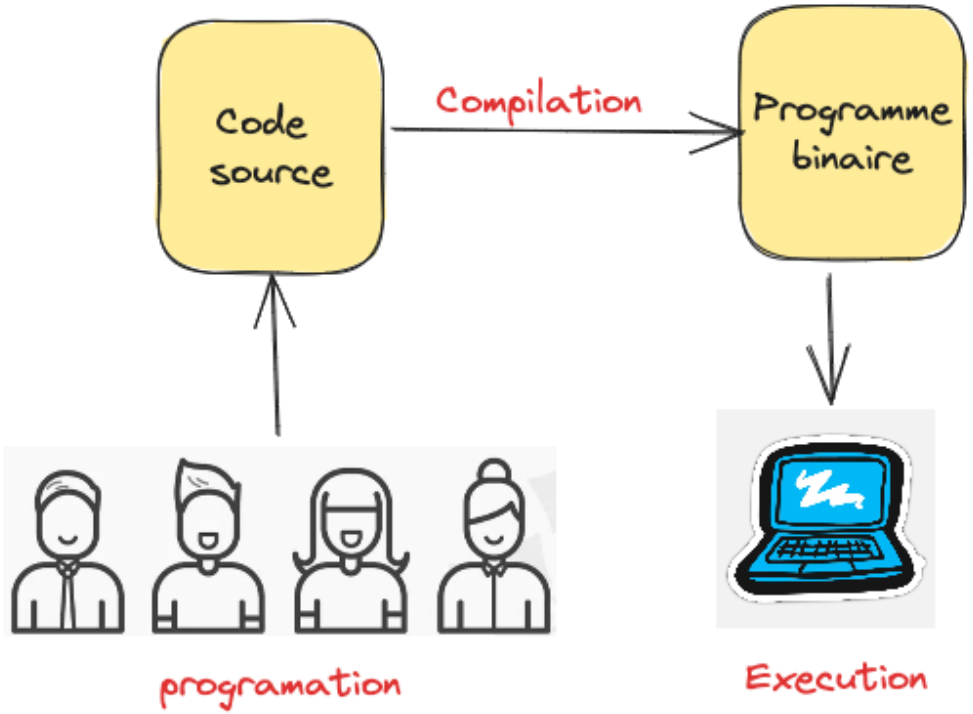
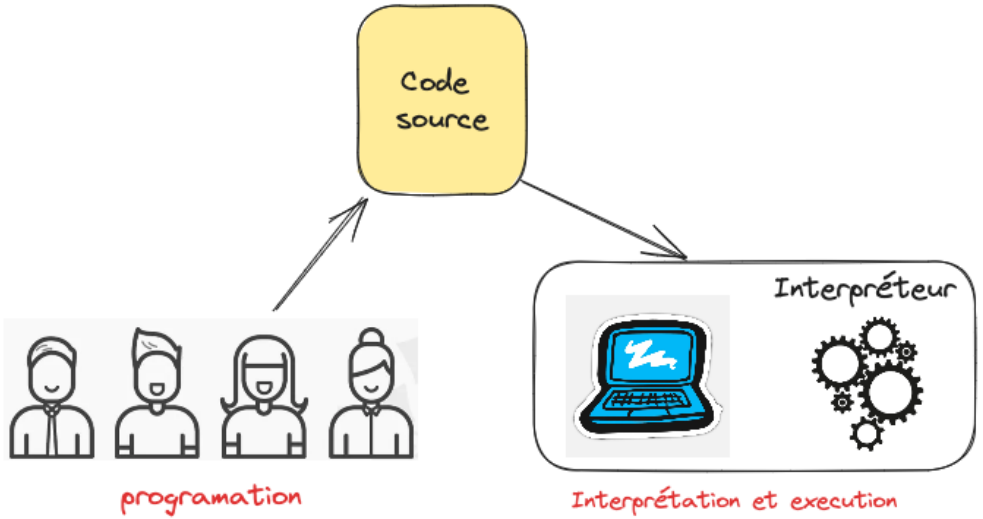


Schéma d'exécution d'un code interprété



Lélia Blin

A qui appartient l'intelligence

Un programme ne fait que ce que lui demande la programmeuse ou le programmeur

L'implicite n'existe pas

Langage étudié: JAVA

- En fait un sous langage
  - Réalisation de calculs arithmétiques
  - Manipulation de chaînes de caractères
  - Lire/écrire/modifier des variables
  - Boucler sur des instructions
    - Faire des tests sur le variables
    - Écrire et appeler des fonctions
    - Manipulation de tableaux

Données manipulées

- Un programme manipule des données
  - appelés aussi **variable**
- Ces données peuvent être de différentes sortes
- On parle en fait de type

Les différents types

- **int** : il s'agit des données entières (par exemple : 3, 4, 5000, etc.)
- **double** : il s'agit des nombres réels avec virgules (par exemple : 3.5, 2.4,etc)
- **String** : il s'agit de chaînes de caractères (par exemple ''Hello World'', ''345'', ''Un message !'', etc)

Il existe d'autre types

Quelques remarques sur les données

- Dans certains langage comme java une donnée ou variable de peut pas changer de type
- Il faut toujours avoir en tête quel type de données on manipule
  - Ex: La division entière 2 / 4 donne 0 alors que la division réelle 2.0 / 4 donne 0.5
- À quoi sert le type String ?
  – Typiquement à stocker des données correspondant à des chaînes de caractères, mais aussi des messages que l'on souhaite afficher

Une machine n'a pas une précision infinie

- Ainsi, on ne peut pas compter jusqu'à l'infini
- Pour les nombres réels, on ne dispose pas d'une précision infinie
  - Par exemple : 1 / 3 est interprété en Java comme 0.3333333333333333
  - Il n'y a pas un nombre infinie de chiffres après la virgule !

Opération sur les données

- Un programme peut faire des opérations sur les données
- Sur les données entières, comme un calculatrice :
  - addition (2 + 5), soustraction, division entière (3 / 4), multiplication, etc
- Sur les chaînes de caractères :
  - concaténation ''Hello'' + ''World !'' donne la chaîne ''Hello World !''

Rappel pas de mélange de type

Afficher

```
''Hello'' + 3
3 * ''Bob''
```

donne un résultat mais n'est pas recommandé

Voir le résultat d'une opération

- Un programme n'affiche rien si on ne lui demande pas
- Un programme qui fait des opérations le fait silencieusement (on ne voit pas l'effet)
- Pour voir le résultat d'une opération, il faut demander au programme de l'afficher
- On utilise la fonction System.out.print ou System.out.println

Voir le résultat d'une opération (Exemple)

- L'argument donné est affiché sur le terminal, par exemple

```
System.out.println (6*7)
```

affiche 42

```
System.out.println (''Hello ! '')
```

affiche Hello !

Voir le résultat d'une opération (Exemple)

```
System.out.println(''Un'' + ''Message'')
```

affiche UnMessag

```
java System.out.println(''6*7'')
```

affiche 6*7 (et pas 42)

Les variables

- Un programme peut stocker les données
  - pour faciliter leur manipulation
  - pour abstraire leur valeur
  - pour les réutiliser plus tard
  - pour faire des calculs complexes
- Il dispose de sa mémoire (pensez à un ensemble de cases)

Les variables

- Une variable indique un endroit de la mémoire où est stocké une donnée
- Une variable a :
  - un nom, par exemple x, y, z, var, z3
  - un type
- Pour utiliser la variable, on utilise son nom
- Opérations sur les variables : Déclaration, Affectation, Lecture et Modification

Opération sur les variables

Déclaration et affectation (Donner le type de la variable et mettre une donnée)

Attention on utilise le symbole = , mais qui ne veut pas dire égalité

On déclare qu'une seule fois le type Par exemple : int x = 3

Lecture (lire la donnée d'une variable)

- On utilise le nom de la variable à la place de la donnée
- Par exemple : System.out.println (x + 2) affiche 5
- Modification (modifier la valeur d'une variable) Comme l'affectation : x = 8

Opération sur les variables

```
int x = 3
int y = 2
int z = x + 1
x = 6
y = 2 * x
System.out.println(x)
System.out.println(y)
System.out.println(z)
```

```
6
12
4
```

Quelques règles de bonne conduite

- Toujours initialiser une variable, par exemple au début du programme
- On déclare **qu'une seule fois** le type d'une variable
- Exemples à ne pas faire

```
int x = 1
int x = 3
```

```
int x = 1
int z = int x
```

Syntaxe non correcte

- Interdit de mettre à gauche de = une valeur et à droite une variable

```
2 = x
```

Que fait la machine ?

- Exemple on a le programme suivant

```
z = (x * x) + 2
```

- Opération effectué par la machine
    i. Récupère la valeur de la variable x
        - Si x n'a pas de valeur → Erreur
    ii. Calcule (x * x) + 2
    iii. Stocke la valeur obtenue dans la variable z

    On calcule d'abord ce qui se trouve à droite du symbole =

Exemple de programme I

```
int x = 3 ;
int y = 2 ;
int z = x * x ;
y = 3 *z ;
System.out.println (x) ;
System.out.println (y) ;
System.out.println (z) ;
```

Affiche

```
3
27
9
```

Exemple de programme II

```
int x = 3 ;
x = x+ x ;
x = x- 1 ;
System.out.println (x) ;
```

Affiche

```
5
```

Suite en Cours-TD ...