

P.O.O. (Programmation Orientée Objet)

CHOUTI Sidi Mohammed

Cours pour L2 en Informatique
Département d'Informatique
Université de Tlemcen
2017-2018

1. Introduction à la Programmation Orientée Objet
2. Classes et objets
3. Héritage et polymorphisme
4. Interface et implémentation
5. Interface graphique et Applet
6. ...

Objectifs du cours POO

1. Appréhender les concepts de **l'approche** objet afin d'être capable de comprendre toute **technologie** basée sur l'objet.

Objet

Classe

Héritage

...

2. S'initier à la programmation en **java**

Langages machine (0/1) → Langages symboliques
(Assembleur) → Langages de haut niveau ou indépendants du matériel (Basic, Fortran, Cobol, etc.)

Programmation structurée → Programmation procédurale →
Programmation modulaire → Programmation objet → ..

Bref historique

- **Simula** a été le premier langage de programmation à implémenter le concept de classes en 1967 !

Ole-Johan
Dahl



Kristen
Nygaard

Bref historique

- **Smalltalk**, en plus d'implémenter les principaux concepts de base (classe, objet, héritage, polymorphisme), introduit la notion d'objet persistant, de traitement des exceptions et le principe Modèle-Vue-Contrôleur. (les années 70).



“ The best way
to predict
the future is
to invent it. ”

Alan Kay (°1940)
The real inventor of Windows

- Depuis, de nombreux langages orientés objets ont été mis au point

Object Pascal

Objective C

C++

Java

C#

...

Paradigmes de programmation

Un paradigme est une manière de programmer

Procédurale	Orientée Objet
<ul style="list-style-type: none">• Que doit faire ce programme ?• De quelles procédures ai-je besoin ?	<ul style="list-style-type: none">• Quelles sont les entités de ce programme ?• Quels sont les objets fondamentaux de l'application ?

Autres exemples de paradigmes : programmation déclarative

- HTML, XML, etc. (descriptive)
- Lisp, etc. (fonctionnelle)
- Prolog, etc. (Logique)

- La programmation objet systématise la modularité et permet la réutilisation de composants logiciels
- Maîtrise de la complexité d'un système, repose sur :
 - L'abstraction
 - La décomposition
 - La connexion

Les raisons du succès de la POO

- Fondée sur une solide approche génie logiciel ;
- Maîtrise de la complexité de grands systèmes logiciels ;
- Consolidation par les méthodes O.O. et le standard (UML) qu'elles utilisent comme langage de modélisation objet ;
- Ses fondements facilitent la programmation distribuée.

Ce qu'il faut savoir sur java

Fichier de code source: **.java**

Fichier compilé (code intermédiaire ou byte code) : **.class**

Outils nécessaires : **JRE, JDK, etc.**

Environnement utilisé en TP : **blueJ**

Qualités essentielles : **Portabilité (JVM) et Facilité d'utilisation**

Conventions de nommage

- Organisation des fichiers
 - Le fichier .java doit avoir le même nom que la classe publique qu'il décrit
 - 1 fichier .java par classe, même pour celle contenant le main(), exception pour les inner classes
- Éléments de base
 - paquetages minuscules
 - classes MajusculePourLaPremiereLettreDeChaqueMot
 - interface MajusculePourLaPremiereLettreDeChaqueMot
 - méthodes minusculePourLaPremiereLettre
 - constantes MAJUSCULES
 - variables minusculePourLaPremiereLettre

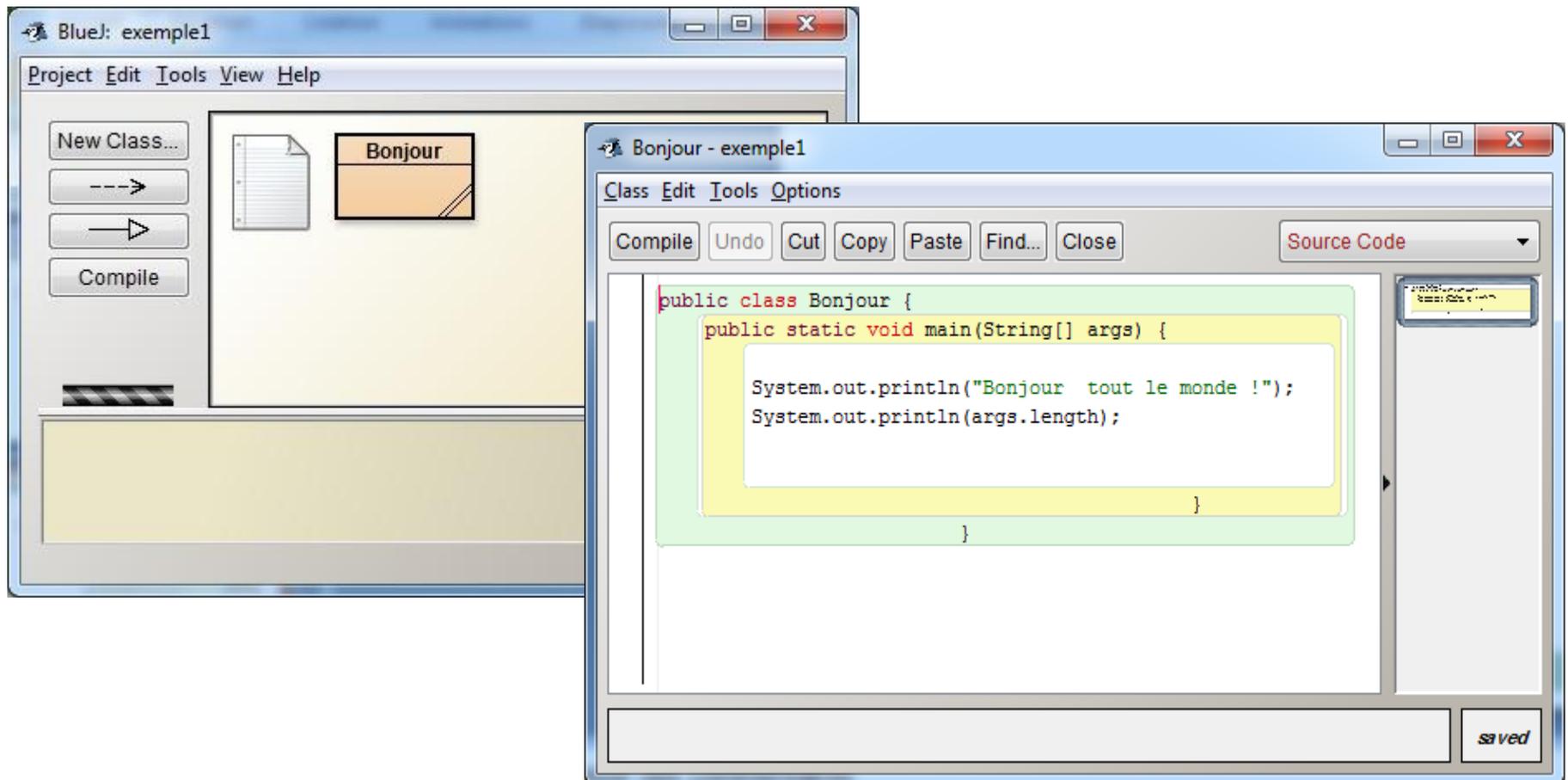
Exemple :

fichier Bonjour.java

```
class Bonjour {  
    public static void main(String[] args) {  
  
        System.out.println("« Bonjour tout le monde !");  
  
    }  
}
```

Ce qu'il faut savoir sur java

blueJ



Références bibliographiques

1- Penser Java :

bruce-eckel.developpez.com/livres/java/traduction/tij2/

2- Conception objet en Java avec BlueJ : une approche interactive, David J. Barnes et Michael Kölling

1- Sites officiels :

Téléchargement <https://www.java.com/fr/>

Documentation

<https://docs.oracle.com/javase/7/docs/api/index.html>

<https://docs.oracle.com/javase/8/docs/api/index.html>

2- Penser Java :

bruce-eckel.developpez.com/livres/java/traduction/tij2/

3- EDI blueJ : <https://www.bluej.org/>

4-La page de ce cours est à : <https://gl2site.wordpress.com/>