

**Programmation et modélisation orientées objet
(USAL5A & USAL2H)**

Programmation Objet en Java

Tristan Crolard

Laboratoire CEDRIC
Equipe « Systèmes Sûrs »

`tristan.crolard@cnam.fr`

`cedric.cnam.fr/sys/crolard`

Programmation en Java « moderne »

La plupart des langages *modernes* reprennent plusieurs principes qui permettent au compilateurs de garantir plus de **sûreté** :

- ▶ l'encapsulation des données privées (**Java**, Scala, Kotlin, Swift, C#)
 - classes et interfaces
- ▶ le typage statique et la factorisation du code (**Java**, Scala, Kotlin, Swift, C#)
 - classes abstraites
 - héritage
 - généricité (**Java 5**)
- ▶ l'inférence de types (**Java 11**, Scala, Kotlin, Swift, C#)
- ▶ **le paradigme fonctionnel** (**Java 21**, Scala, Kotlin, Swift, F#)
 - fonctions anonymes (**Java 8**, Scala, Kotlin, Swift, F#)
 - filtrage ou *pattern-matching* (**Java 21**, Scala, Swift, F#)
 - collections immutables (**Java ??**, Scala, Kotlin, Swift, F#)

Paradigmes de programmation

Style	Collections	
	mutables	immutables
impératif	idéal	ok
impératif concurrent	ko	ok
fonctionnel	/	idéal
fonctionnel concurrent	/	idéal

Style impératif : variables (mutables), instructions (séquence, boucles, ...)

Style fonctionnel : constantes, fonctions et expressions (composition, récursion, ...)

Remarques :

- ▶ il faut éviter d'appeler du code impératif depuis le fonctionnel (l'inverse ne pose pas de problème)
- ▶ le paradigme « orienté objet » peut compléter les deux styles, impératif et fonctionnel.

Bibliographie

Effective Java, 3rd Edition

Joshua Bloch

December 2017

Publisher: Addison-Wesley Professional

ISBN: 9780134686097

Modern Java in Action: Lambdas, streams, functional and reactive programming

Raoul-Gabriel Urma, Mario Fusco, Alan Mycroft

November 2018

Publisher: Manning Publications

ISBN:978-1-61729-356-6