

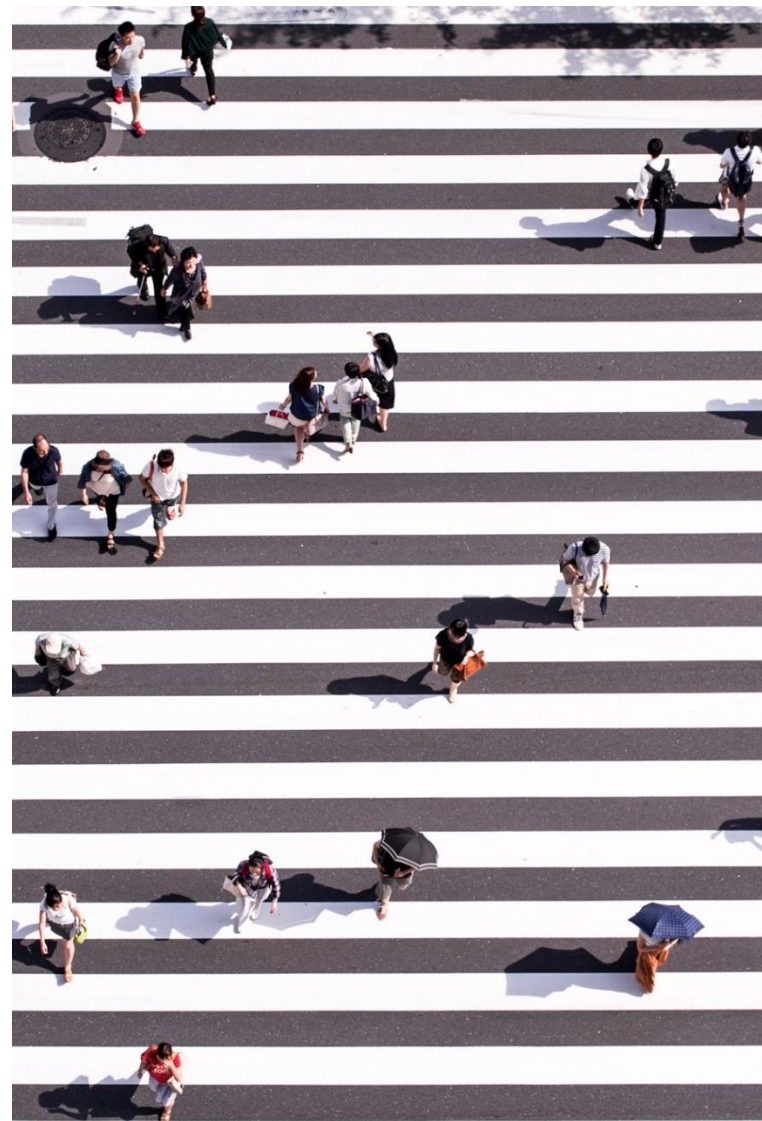
A photograph of two men in a professional setting. One man, wearing a striped sweater, is writing on a whiteboard with a black marker. The other man, in a grey sweater, stands beside him, looking at the whiteboard. The whiteboard contains some faint sketches and text. The background is a bright, modern office with large windows and a potted plant.

niji

# Les annotations

# Sommaire

1. A quoi ça sert ?
2. Types d'annotations
3. AnnotationTarget
4. Processing
5. Réflexion
6. AbstractProcessor
7. Kotlinpoet
8. Sources



## A quoi ça sert ?

- Introduit dans Java 5
- Les annotations permettent d'attacher des métadonnées au code
- Ce sont des classes contenant uniquement des données et ne contiennent aucun code exécutable
- Certaines annotations intégrées ont un effet sur le processus de compilation
- inspectées au moment de l'exécution par le système de réflexion

## Types d'annotations

### 3 types de RetentionPolicy :

- **SOURCE** : C'est comme un surnom quand on est bébé. Après avoir grandi, personne ne s'en souvient, ni vous-même.
- **BINARY** : C'est comme si le surnom de votre famille restait avec vous pour toujours. Mais seul le membre de votre famille y a accès.
- **RUNTIME** : C'est comme votre surnom à l'école, tout le monde le sait s'il demande simplement à votre camarade de classe... ou mieux connu comme «ami»

## AnnotationTarget

Une annotation doit être défini sur quoi elle s'applique :

- Classe
- Fonction
- Champs...

<https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.annotation/-annotation-target/index.html>

## Processing

3 manières différentes de traiter une annotation :

- Par réflexion
- `AbstractProcessor` pour le prétraitement
- `Kotlinpoet` pour générer de nouveaux codes

## Reflexion

```
private fun bindReflectionValue(target: Any) {  
    val declaredFields : Array<(out) Field!> = target::class.java.declaredFields  
  
    for (field : Field! in declaredFields) {  
        for (annotation : Annotation! in field.annotations) {  
            when (annotation) {  
                is ReflectRuntime -> {  
                    field.isAccessible = true  
                    field.set(target, annotation.value)  
                }  
            }  
        }  
    }  
}
```

Utilise la réflexion pour répertorier toutes les variables déclarées dans la classe et recherche toutes celles qui sont annotées avec `ReflectRuntime`.

L'annotation rend la variable accessible en écriture et la définit en fonction de la valeur d'annotation, c'est-à-dire 5. Il s'agit d'une annotation vide de sens, mais elle explique que l'annotation trouve un élément de programmation spécifique pour son traitement.

L'annotation seule ne fait rien.

## AbstractProcessor

- Classe AbstractProcessor : Cette classe examine les valeurs d'annotation pour calculer les options, les annotations et la version source prises en charge par ses sous-types.
- @AutoService : Google AutoService est un outil générateur de code open source. Le but de cette bibliothèque est d'économiser des efforts et du temps et, en même temps, d'éviter les erreurs de configuration.
- GetSupportedAnnotationType() : Enregistre l'annotation qui sera extraite, c'est-à-dire l'annotation CheckCamelSource.



## Kotlinpoet

KotlinPoet est une API Kotlin et Java pour générer des fichiers source .kt.

La génération de fichiers source peut être utile lorsque vous effectuez des tâches telles que le traitement d'annotations ou l'interaction avec des fichiers de métadonnées. En générant du code, vous éliminez la nécessité d'écrire du passe-partout tout en conservant une source unique de vérité pour les métadonnées.

## Sources

- <https://medium.com/@elye.project/annotation-tutorial-for-dummies-in-kotlin-1da864acc442>
- <https://kotlinlang.org/docs/tutorials/kotlin-for-py/annotations.html>
- <https://kotlinlang.org/docs/reference/annotations.html>
- <https://docs.oracle.com/javase/7/docs/api/javax/annotation/processing/AbstractProcessor.html>
- <https://www.baeldung.com/google-autoservice>
- <https://github.com/square/kotlinpoet>

The background of the slide features several incandescent light bulbs hanging from black cords against a dark, out-of-focus background. One bulb in the lower right quadrant is turned on, emitting a warm, yellowish glow, while the others remain unlit.

niji

Questions ?