

Nouvelles Fonctionnalités Java

Pattern Matching (Switch Avancé)

Introduction

- Introduit dans Java 17+ pour rendre `switch` plus puissant.
- Permet d'éviter les castings explicites et simplifie la logique conditionnelle.

Exemples :

```
static void process(Object obj) {  
    switch (obj) {  
        case Integer i -> System.out.println("Nombre: " + i);  
        case String s   -> System.out.println("Texte: " + s);  
        case null       -> System.out.println("Valeur null");  
        default         -> System.out.println("Type inconnu");  
    }  
}
```

- Plus concis et sûr.
- Meilleure lisibilité du code.

Strings et Texte Enrichi (Text Blocks)

Introduction

- Introduit en Java 13 et stabilisé en Java 15.
- Facilite la manipulation des chaînes multilignes sans concaténation.

Exemple :

```
String json = """
{
    "name": "ChatGPT",
    "type": "IA"
}
""";
```

- Plus lisible.
- Évite les échappements inutiles (`\n`, `\"`).
- Idéal pour JSON, HTML, SQL.

Extensions des Classes Scellées

Introduction

- Introduit en Java 15 (préversion) et stabilisé en Java 17.
- Permet de restreindre les classes qui peuvent hériter d'une classe mère.

Exemple :

```
sealed class Shape permits Circle, Rectangle {}  
final class Circle extends Shape {}  
final class Rectangle extends Shape {}
```

- Améliore la sécurité.
- Facilite l'évolution du code.
- Optimisé pour le pattern matching.

Utilisation dans Spring

Intégration des nouvelles fonctionnalités

- Pattern Matching : Simplification des conditions dans les contrôleurs.
- Text Blocks : Meilleure gestion des requêtes SQL en Spring Data.
- Classes Scellées : Modélisation plus stricte des entités métier.

Exemple d'utilisation de Text Blocks dans Spring Data :

```
@Query("""  
    SELECT * FROM users WHERE name = :name  
    """)  
List<User> findByName(@Param("name") String name);
```

- Code SQL plus lisible.
- Moins de risques d'erreurs d'échappement.

Conclusion

Les nouvelles fonctionnalités de Java permettent :

- Un code plus lisible et concis.
- Une meilleure gestion des types et de la sécurité.
- Une intégration fluide avec Spring pour des applications modernes et maintenables.

Prêt à les intégrer dans vos projets ?

