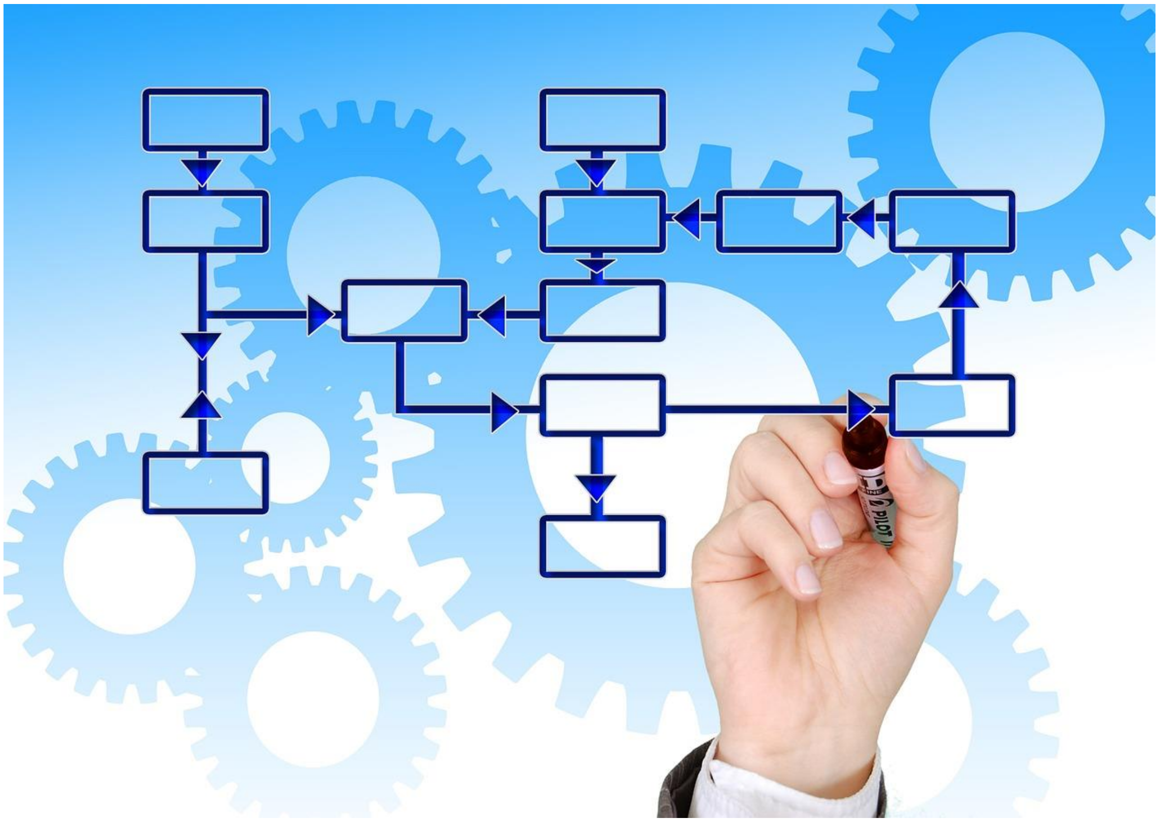

JAVA 8 FEATURES AND FUNCTIONAL PROGRAMMING

OBJECT ORIENTED PROGRAMMING I

Sercan Külcü | Object Oriented Programming I | 10.01.2023

Contents

Introduction

Java 8 introduced several new features and improvements, including functional programming capabilities. In this chapter, we will explore these new features and how they can be used in Java applications.

Lambda Expressions

Lambda expressions are a new syntax for defining anonymous functions in Java. They allow you to write code that is more concise and expressive, especially when working with collections and streams. A lambda expression is defined using the -> operator, and it can be assigned to a functional interface, which is an interface with a single abstract method.

Streams

Streams are a new API for processing collections in Java. They allow you to perform common operations such as filtering, mapping, and reducing on collections using a functional programming style. Streams can be parallelized, which means they can take advantage of multiple cores in a processor to improve performance.

Method References

Method references are a shorthand syntax for referring to methods or constructors in Java. They can be used instead of lambda expressions when the target method has the same signature as the abstract method of a functional interface. Method references make code more readable and easier to maintain.

Optional

Optional is a new class in Java 8 that represents a value that may or may not be present. It is designed to help eliminate NullPointerExceptions in Java code. Optional can be used to wrap values that might be null, and it provides a set of methods for working with nullable values in a safer and more concise way.

Functional Interfaces

Functional interfaces are interfaces with a single abstract method. They are used to represent functions and provide a way to define lambda expressions in Java. Java 8 introduced several new functional interfaces in the java.util.function package, including Predicate, Function, and Consumer.

Parallel Streams

Parallel streams are a new feature in Java 8 that allow you to process collections in parallel, using multiple threads. Parallel streams can significantly improve the performance of collection processing, especially for large collections.

Conclusion

Java 8 introduced several new features that make it easier to write concise, expressive, and efficient code. The new functional programming capabilities, including lambda expressions, streams, method references, and functional interfaces, allow you to write code that is more functional and modular. Optional and parallel streams provide additional benefits, including safer and more efficient code. By mastering these new features, you can become a more productive and effective Java developer. However, it's important to remember that functional programming is not always the best approach for every problem, and you should carefully consider the needs and requirements of your project when deciding whether to use these new features.