DAY-21

# JAVA

## JDK API

### STRING BUFFER:

* String Buffer is a class in java that represents a mutable sequence of characters. It provides an alternative to the immutable string class, allowing you to modify the contents of a string without creating a new object every time.
* It is mutable.
* Thread safe
* And it is efficient for frequent modifications.

STRINGBUFFER METHODS:

Append, insert, delete, reverse, charAt, deleteCharAt, replace

StringBuffer sb=new StringBuffer(“Simmuu”);

Sb.replace (1,2,” Nani”);

O/P: SNaniu

### STRING BUILDER:

* String Builder class is a part of the java.lang package that provides a mutable sequence of characters.
* It allows for changes in place, making it memory efficient and faster for large frequent string manipulations.
* It doesn’t guarantee synchronization.
* It is faster than the string buffer in most implementations.

## JAVA.UTIL. OPTIONAL:

* It is a container object that may or may not contain a non-null value. It was introduced in Java 8 to avoid the NullPointerException and make code readable and expressive when dealing with potentially null values.
* It avoids null checks.
* It is used foe encourages functional programming style.
* By using optional, it makes API more expressive (like return Optional<String> instead of null).

CREATING AN OPTIONAL:

* Optional <String> empty = Optional.empty(); //No value
* Optional <String> name = Optional.of(“Nani”); //non-null value
* Optional <String> nullAbleName = Optional.ofNullable (“Nani”); //can be null

## REACTIVE PROGRAMMING:

* It provides a stream-based processing model, where data flows in a continuous stream of events. This enables developers to easily compose and transform data in a declarative and concise manner, allowing for efficient data processing and manipulation.
* It is a powerful approach that provides several benefits for modern applications.
* Scalability
* Flexibility
* Error-handling

MODULE IN JAVA:

* The module system is in Java 9. It helps organize large applications into smaller, manageable pieces, and it improves encapsulation, security, and performance.
* A module is self-contained unit of code that contains:
* Module-info.java file.
* Packages and classes.

### LOCALIZATION:

* It is the process of adapting your internationalized application to a specific locale (language+region)
* It translates resource bundles
* It can be used for Format dates, numbers, and currencies according to the locale.

INTERNATIONALIZATION:

* It is the process of designing your application. So, it can be easily adapted to various languages and regions without engineering changes.
* It uses resource bundles to store locale-specific data.
* It avoids hardcoding strings, dates, currencies, etc.
* It uses Java APIs for formatting and locale handling.

CODE STRUCTURE:

* It improves clarity and maintainability.
* It has some important guidelines like:
* Use meaningful package names.
* Keep classes small and focused.
* Limites method size.

COMMENTS:

* It can be mainly used in Javadoc for public APIs.
* We can add comments for complex logic.

### FORMATTING:

* It improves readability and workload.
* Formatting rules are:
* Indentation
* Braces
* Line length
* Blank lines: using the separate logical blocks