23/06/2025

Monday

JAVA 8 Features

1. Lamda Expressions
2. Function Interfaces
3. Default and static method in interface
4. Method references
5. Stream API
6. Optional Class
7. New Date and Time API
8. Collectors

Lambda Expressions:

* Represents methos using an expression
* It is used to implement **Functional Interfaces**(an interface with single abstract method)
* Useful in Collections and threads etc
* No need to declare parameter types:
* Syntax:

(paraemters) -> {body}

* It is an anonymous function (a function without function name)
* Rules for Lambda Expressions:

1. Lambda can be assigned to functional interfaces only.
2. You can omit parenthesis if there is only one parameter.
3. Curly braces {} and return keyword are optional if there is only one statement.
4. Generally used with Collections and multithreading.

Stream API

* The stream API in java8 allows functional style processing of collections like List, Set etc

Wrapper Classes

* A Wrapper Class in Java is a class that wraps (encapsulates) a primitive datatype into an Object
* Methos parse(), compare().etc
* Supports autoboxing and unboxing
* Autoboxing:- automatically converts primitive to object conversion Ex: int to Integer

Ex : int x =5;

Integer y =x;

* Unboxing:- Automatically converts object to primitive

Ex: Integer x = 5;

int y = x;

|  |  |
| --- | --- |
| **Primitive Types** | **Wrapper Class** |
| byte | Byte |
| short | Short |
| int | Integer |
| long | Long |
| float | Float |
| double | Double |
| char | Char |
| boolean | Boolean |

* compare() :- compares two wrapper objects
* equals() :- compares wrapper object values.