

Functional Interface

Thursday, September 18, 2025 12:20 PM

Types of Interfaces in Java

1 Regular (Normal) Interface

- The standard interface type.
- Contains **abstract methods** (implicitly public abstract) and constants.
- A class implements it using implements keyword.

2 Functional / SAM (Single Abstract Method) Interface

- Has **exactly one abstract method**.
- Can have any number of default/static methods.
- Supports **lambda expressions**.
- Annotate with `@FunctionalInterface` for clarity.
- Examples: `Runnable`, `Callable<T>`.

```
@FunctionalInterface 2 usages new *  
interface A{  
    void show() ; 1 usage new *  
}
```

3 Marker (Tag) Interface

- An **empty interface** with **no methods**.
- Used to “mark” classes so JVM or frameworks treat them specially.
- Examples: `Serializable`, `Cloneable`.

Bonus:

Definition of Annotation in Java

An **annotation** in Java is a **special form of metadata** (extra information) added to code — classes, methods, fields, parameters, etc.

It **does not affect the execution directly**, but it tells the compiler, tools, or frameworks how to treat the element it's attached to.

They start with `@` symbol.

Example:

```
@FunctionalInterface 2 usages new *
```

Key Points

- Introduced in **Java 5**.
- Provide **instructions** to compiler or frameworks.
- Can be used for **compile-time checks**, **runtime processing**, or **documentation**.