Lesson-7-Class Notes

Java 8 – Interface default implementation

default keyword for Java 8 interface implemented method meant for we are providing default implementation can access from all the classes which implements that interface.

default keyword not meant for package level access. It is by default public visibility like all other Interface methods.

Default implementation of the Interface behavior can able to override in sub class.

Functional Interface

An interface with only one unimplemented method. It can be implemented using Lambda.

For each method from the List interface has the default implementation uses Functional Interface Consumer.

Consumer Interface accept a Input and no return data

Lambda Signature

(arg) -> body

```
(x) -> System.out.println(x);
```

// Implementing Consumer in the traditional approach

```
Class MyConsumer implements Consumer{

        void accept(String t){
    if(t.startWith("A")
      System.out.println(t);
  }
}

  Main() {

      List<String> list = new ArrayList<>();
      list.add("Java");
      list.add("Java Spring");
      list.add("Android");
      list.add("Java Hibernate");
      list.add("Kotlin");
```

```
            list.forEach(new MyConsumer());
            // Replace the above line is a functional way as mentioned below
            list.forEach(x->System.out.println(x));
    }
```

Template to implement Iterable in your class

```
        Class MyClass implements Iterable{
            public Iterator<T> iterator(){
         return new Iterator();
        }
        class MyIterator implements Iterator {
            boolean hasNext(){
           }
            String next(){
         }
    }
}
```

When to go for Enum class implements Interface

- The choice between using enum classes that implement interfaces and regular classes implementing interfaces depends on your specific use case.
- If you need a fixed set of distinct, predefined types with specific behaviors, enums are a good choice.
- If you require more flexibility in creating instances or extending behavior, regular classes implementing interfaces might be a better fit. Because Enum cannot inherit from other class, itself inherited from Enum.

Advantages of Enum

1. Enum constant as Singleton Instance
2. Compile time safety
3. Enum can use with Switch statement
4. It represents a distinct, predefined type with specific behavior

Example Scenario

You can have a Payment interface with pay method.
Enum class can implements Payment interface and override the behavior for three types payment like Cash, Credit Card and Debit card.

This is helpful when you want to encapsulate different behaviors within an enum, and clients can choose a strategy by selecting the appropriate enum constant.

Also useful in different design pattern like Strategy pattern, Factory pattern(singleton).

Importance of Overriding equals and Hashcode

→ Always override equals() and hashCode()
→ If equals says true hashcode of those objects must be same.
→ Always make the Key type as immutable for your hash collection