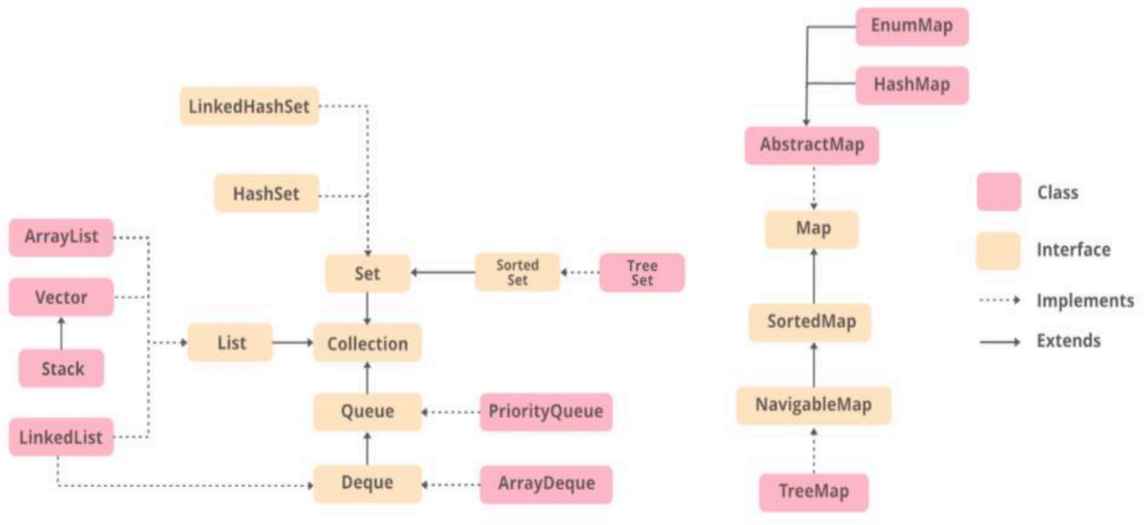
Class - 2

Contents

1. Collections API
2. Types of Interface [Normal Interface, Functional interface, Marker Interface]
3. Abstract classes, Lambdas
4. Anonymous Inner Classes

Collections in JAVA



POINTS TO DISCUSS :

1. Iterable Interface
2. Collection Interface
3. List Interface
4. ArrayList (https://ide.geeksforgeeks.org/VAojWFIbfu)
5. LinkedList (https://ide.geeksforgeeks.org/v4HVcUORsc)
6. Set Interface
7. HashSet (https://ide.geeksforgeeks.org/KgfnJhwz8j)
8. Map Interface
9. HashMap (https://ide.geeksforgeeks.org/L3WyK25UI5)

Refer github repo :
https://github.com/ashishgulatiG4G/JBDL-73/tree/main/Class-2%20Collection

ABSTRACT CLASS :

A class which is declared with the abstract keyword is known as an abstract class in Java. It can have abstract and non-abstract methods (method with the body).

```
abstract class Shape {
        int colour;
        abstract void draw(); // An abstract function
}
```

Challenge 1 : Can we have a function in the abstract class with a body ?

References :
https://www.geeksforgeeks.org/abstract-classes-in-java/

INTERFACE :

The interface in Java is a mechanism to achieve abstraction. There can be only abstract methods in the Java interface, not the method body. It is used to achieve abstraction and multiple inheritance in Java. In other words, you can say that interfaces can have abstract methods and variables. It cannot have a method body. Java Interface also represents the IS-A relationship . That means all the methods in an interface are declared with an empty body and are public and abstract and all fields are public, static, and final by default. A class that implements an interface must implement all the methods declared in the interface.

```
// A simple interface
interface Player {
        final int id = 10;
        int move();
}
```

Challenge 1 : Can we have a function in the interface with a body ?
(HINT : default function)

Challenge 2 : Can we override the default function in the implementing class ?

OR

What do we call the default methods of an interface in the implementation class ?

HINT : If two interfaces have the same name as the default method , we need to override the method in our implementation class. It's a rule .

We override the default function in the class and inside its body use the <interfaceName>.super.<methodName>()

Challenge 3 : If we are overriding a default function in our implementation class, then how can we call the default function of the interface ?
HINT : <InterfaceName>.super.<functionName>(x,y)

Challenge 4 : How to access the variables in our implementation class which have been defined in the interface ? Can we change its value in the implementation class and Why?
(**HINT** : All variables in interface are public, final and static)

Challenge 5 : In the implementation class , in the implementation of an abstract method,can we give access modifiers as protected ?

Challenge 6 : If a class A is implementing an interface I and extending a class C . Which is correct ?

class A extends B implements I

OR

class A implements I extends C

Challenge 7 : If my class A extends interface B and interface C , both have the same default method “power” . Even if my object of class A does not call power and does not overrides the default method power , will there be any issues and why ?

Challenge 8 : Is it possible that in parent class , function A has protected access modifier In child class which extends parent , while overriding , can we give access modifier of A as :
(i) public
(ii) private

HINT : we can't assign weaker privileges while overriding methods .

References :
https://www.geeksforgeeks.org/interfaces-in-java/
https://www.geeksforgeeks.org/static-method-in-interface-in-java/
https://www.geeksforgeeks.org/default-methods-java/

ANONYMOUS INNER CLASS :

An anonymous inner class can be useful when implementing an interface or abstract class with certain “extras” such as overriding abstract methods of interface/abstract class , without having to actually subclass a class.

Challenge 1 : Is multiple inheritance allowed in Java with (i) classes (ii) interface ?

Challenge 2 : Is multiple inheritance allowed in Java ?

Challenge 3 : Does an interface implements another interface or extends another interface ?

Challenge 4 : Can an interface extend a class and why?

Challenge 5 : What is the difference between Abstract Class and Interface ?

Challenge 6 : Can we override default methods of an interface in other interfaces which extend it ?

HINT :
https://www.geeksforgeeks.org/difference-between-abstract-class-and-interface-in-java/

References :
https://www.geeksforgeeks.org/nested-classes-java/
https://www.geeksforgeeks.org/anonymous-inner-class-java/
https://stackoverflow.com/questions/2515477/why-is-there-no-multiple-inheritance-in-java-but-implementing-multiple-interface
https://www.geeksforgeeks.org/java-and-multiple-inheritance/

FUNCTIONAL INTERFACE :

A functional interface is an interface that contains only one abstract method. They can have only one functionality to exhibit. From Java 8 onwards, lambda expressions can be used to represent the instance of a functional interface. A functional interface can have any number of default methods. Runnable, ActionListener, Comparable are some of the examples of functional interfaces.

Challenge 1 : Lets checkout Runnable Interface

Challenge 2 : Let us now create an object of Runnable interface using Anonymous Inner Class

Challenge 3 : Lets now try to write our first lambda by trying to do the Challenge 2 using Lambda .

Challenge 4 : Now we have our object of Function Interface (Runnable) . How do we know which function to call ?

Challenge 5 : Lets now create a custom Function Interface with a function taking 2 args. And now lets create object of this custom functional interface using

(i) Anonymous Inner Class
(ii) Lambda

Challenge 6 : Convert the following to Lambda :

Challenge 7 : If the interface for which we want to write lambda contains 2 abstract methods. Is there a problem ?

Challenge 8 : If the interface for which we want to write lambda contains 2 default methods and a single abstract . Is there a problem? How can we call the default methods ?

Challenge 9 : Why do we use @FunctionalInterface annotation ? Is it mandatory ?
HINT : @FunctionalInterface tells users that this interface is intended to be a functional interface and if there is more than one abstract methods added, then it throws at the interface level at compile time .

Challenge 10 : Is Comparator Interface a Functional Interface ?
HINT : equals method is coming from Object class .

Challenge 11 : If we are creating lambda for a Functional Interface in a class. Does that class need to implement the interface?

References :
https://www.geeksforgeeks.org/functional-interfaces-java/
https://stackoverflow.com/questions/50892117/why-to-use-functionalinterface-annotation-in-java-8