INTERFACES

20384114

CONTENT

What is an Interface

- An **interface in Java** is a blueprint of a class. It has static constants and abstract methods.

- 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.

What is an Interface

- An interface is an abstract "class" that is used to group related methods with "empty" bodies

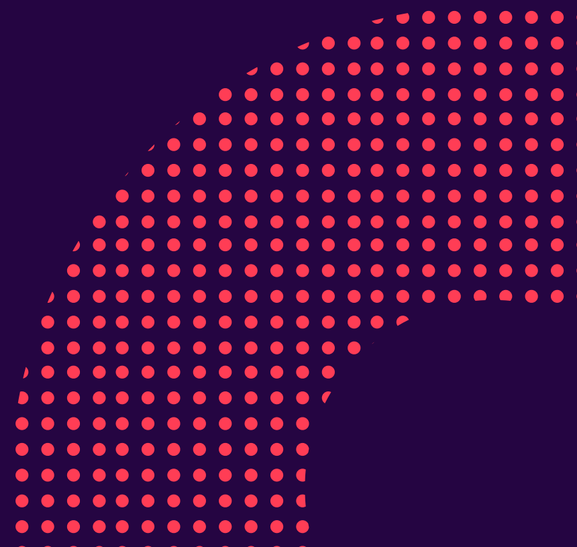
- To access the interface methods, the interface must be "implemented" by another class with the implements keyword

USES

- Interfaces in Java is used to achieve Abstraction

- Support the functionality of Multiple Inheritance

- It can be used to achieve loose coupling



Declaring an Interface

An interface is declared by using the "interface" keyword. It provides total abstraction; means all the methods in an interface are declared with the empty body, and all the fields are public, static and final by default. A class that implements an interface must implement all the methods declared in the interface.

Syntax

```
interface {

    // declare constant fields
    // declare methods that abstract
    // by default.
}
```

Example

```
interface printable{
void print();
}
class A6 implements printable{
public void print(){System.out.println("Hello");}

public static void main(String args[]){
A6 obj = new A6();
obj.print();
 }
}
```

Abstract Class vs Interface

Abstract class	Interface
1) Abstract class can **have abstract and non-abstract** methods.	Interface can have **only abstract** methods. Since Java 8, it can have **default and static methods** also.
2) Abstract class **doesn't support multiple inheritance**.	Interface **supports multiple inheritance**.
3) Abstract class **can have final, non-final, static and non-static variables**.	Interface has **only static and final variables**.
4) Abstract class **can provide the implementation of interface**.	Interface **can't provide the implementation of abstract class**.
5) The **abstract keyword** is used to declare abstract class.	The **interface keyword** is used to declare interface.

Multiple Inheritance by Interface

An interface contains variables and methods like a class but the methods in an interface are abstract by default unlike a class. Multiple inheritance by interface occurs if a class implements multiple interfaces or also if an interface itself extends multiple interfaces.

```java
interface AnimalEat {
   void eat();
}
interface AnimalTravel {
   void travel();
}
class Animal implements AnimalEat, AnimalTravel {
   public void eat() {
      System.out.println("Animal is eating");
   }
   public void travel() {
      System.out.println("Animal is travelling");
   }
}
public class Demo {
   public static void main(String args[]) {
      Animal a = new Animal();
      a.eat();
      a.travel();
   }
}
```

References

www.geeksforgeeks.org

www.tutorialspoint.com

www.javatpoint.com

Thank You