SNS COLLEGE OF TECHNOLOGY

(An Autonomous Institution)
Coimbatore - 641035.

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A++" Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

Department of Computer Applications

Course Code: 23CAT606

Course Name: Java Programming

Unit I: Java Fundamentals

Topic 6: Interface

Recall

01 Package

02 Types

03 Built in Package

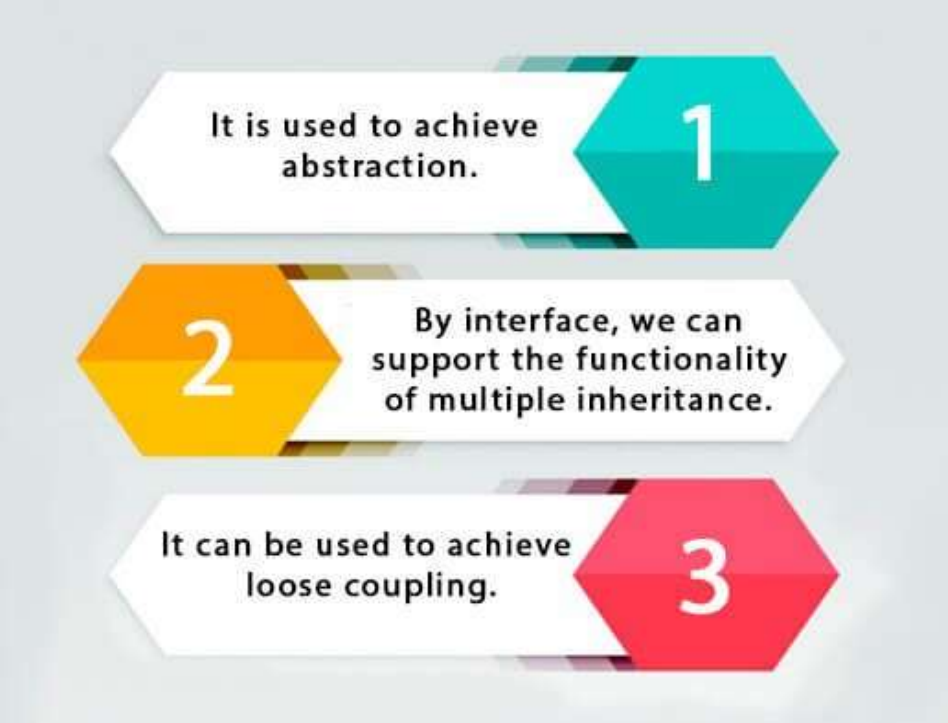
04 User defined Package

05 Accessing the Package

06 Example

07 Assessment

INTERFACE



1 It is used to achieve abstraction.

2 By interface, we can support the functionality of multiple inheritance.

3 It can be used to achieve loose coupling.

Use of Interface in Java Application with Realtime Examples

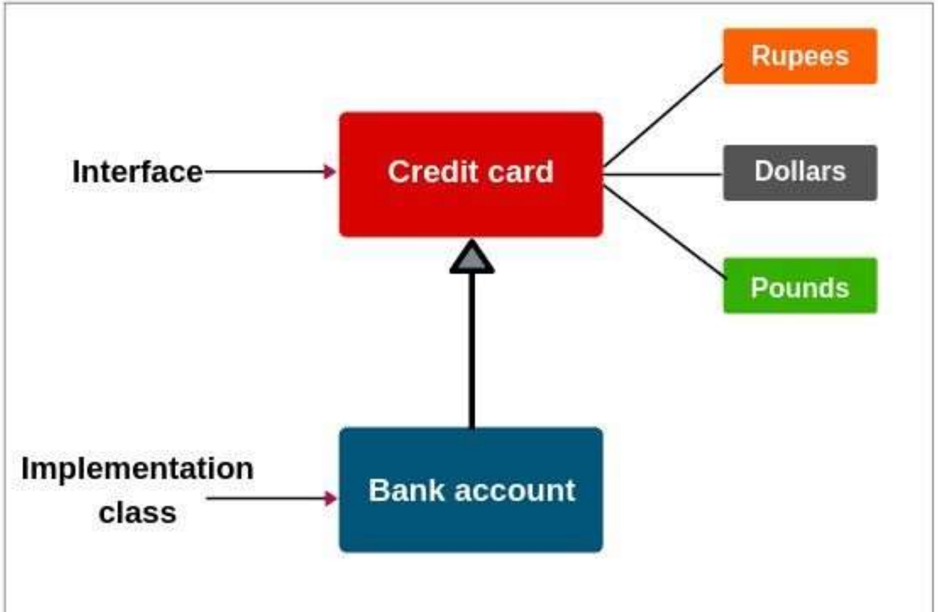


Fig: Interface and Implementation class

INTRODUCTION TO 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 method body.
- It cannot be instantiated just like abstract class.
- An interface is not extended by a class; it is implemented by a class.
- An interface can extend multiple interfaces.

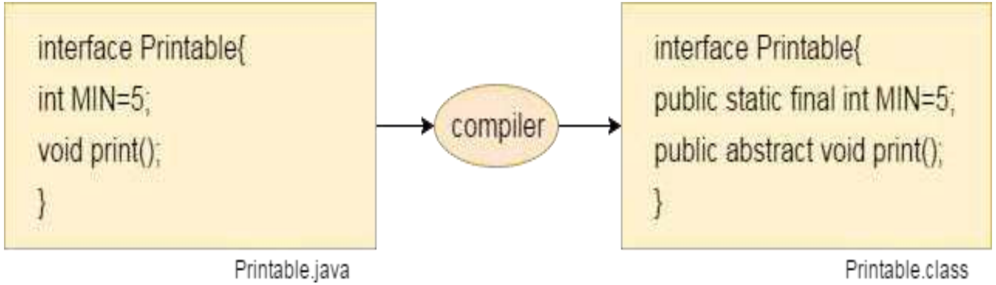
DECLARING INTERFACE

- The **interface** keyword is used to declare an interface.
- Syntax:

```
public interface NameOfInterface
{
      // Any number of final, static fields
      // Any number of abstract method declarations
}
```

Interface

- An interface is implicitly abstract.
- So not need to use the **abstract** keyword while declaring an interface.
- Each Fields in an interface is also implicitly **static** and **final**, so the static and final keyword is not needed (Refer below diagram).
- Methods in an interface are also implicitly public.



```
interface Printable{
int MIN=5;
void print();
}
```

```
interface Printable{
public static final int MIN=5;
public abstract void print();
}
```

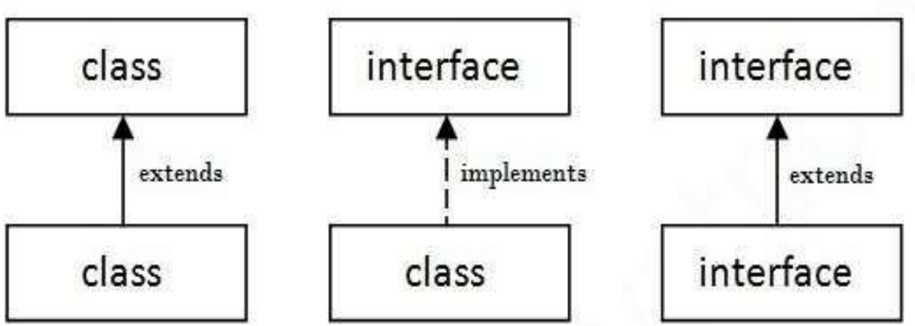
Interface example

Example:

```
 Interface ItemConstants                      //interface declared
 {
       int code = 1001;               // Variable declared in interface
       string name = “Fan”;
       void display();                // Method declared in interface
}
```

Extending interface

- An interface can extend another interface like class.
- The **extends** keyword is used to extend an interface.
- A **class implements an interface**.



```
interface Name2 extends Name1
{
                Body of Name2
}
```

Implementing interface

- Interfaces are used as “Super classes” whose properties are inherited by classes.
- syntax

```
Class className implements interfacename
{
            Body of classname
}
```

Implementing Interface

Example:

```
interface Drawable                  // Interface declared
{
void draw();
}
   class Rectangle implements Drawable                //implementing
   {
   public void draw()
   {
   System.out.println("drawing rectangle");
   }}
```

OUTPUT: drawing rectangle

Implementing Interface

```
  interface Printable
   {
  void print();
  }
  interface Showable
  {
  void show();
  }
 class A7 implements Printable,Showable

   {
  public void print()
{
System.out.println("Hello");
}
  public void show()
{
System.out.println("Welcome");
}
   public static void main(String args[])
{
   A7 obj = new A7();  obj.print();



   obj.show();
    } }
```

OUTPUT:

```
Hello
Welcome
```

Reference

1. Herbert Schildt “ The Complete Reference Java 2, 8th edition , Tata McGraw Hill, 2011
2. Ralph Bravaco, Shai Simonson, “Java Programming: From the Ground up Tata McGraw Hill, 2012
3. https://www.scientecheasy.com/2020/06/packages-in-java.html/#2_Predefined_Packages_in_Java_Built-in_Packages
4. https://www.scientecheasy.com/2019/06/java-interface-use.html/

Summary
1
2
3