

ASSIGNMENT NO: 5

Problem Statement: Design and develop a context for given case study and implement an interface for Vehicles Consider the example of vehicles like bicycle, car, and bike. All Vehicles have common functionalities such as Gear Change, Speed up and apply brakes. Make an interface and put all these common functionalities. Bicycle, Bike, Car classes should be implemented for all these functionalities in their own class in their own way.

Aim: To understand Interface in Java

Theory: 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 is used to achieve abstraction and multiple inheritance in Java. A programmer uses an abstract class when there are some common features shared by all the objects .A programmer writes an interface when all the features have different implementations for different objects Interfaces are written when the programmer wants to leave the implementation to third party vendors An interface is a specification of method prototypes.

All the methods in an interface are abstract methods

1. An interface is a specification of method prototypes
2. An interface contains zero or more abstract methods
3. All the methods of interface are public, abstract by default
4. An interface may contain variables which are by default public static final
5. Once an interface is written any third party vendor can implement it
6. All the methods of the interface should be implemented in its implementation classes
7. If any one of the method is not implemented, then that implementation class should be declared as abstract
8. We cannot create an object to an interface
9. We can create a reference variable to an interface
10. An interface cannot implement another interface
11. An interface can extend another interface
12. A class can implement multiple interfaces

Syntax:

```
interface <interface_name>{
```

1. Java Interface Example

In this example, the Printable interface has only one method, and its implementation is provided in the A6 class. 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();
```

```
}
```

```
}
```

2. Write an example program for interface

Interface Shape

```
{
```

```
void area (); void
```

```
volume ();
```

```
double pi = 3.14;
```

```
}
```

```
class Circle implements Shape
```

```
{
```

```
double r;
```

```
Circle (double radius)
```

```
{r = radius; }
```

```
public void area ()
```

```
{
```

```
System.out.println ("Area of a circle is : " + pi*r*r );
```

```
}
```

```
public void volume ()
```

```
{
```

```
System.out.println ("Volume of a circle is : " + 2*pi*r);
```

```
}
```

```
}
```

```
class Rectangle implements Shape
```

```
{
```

```
double l,b;
```

```
Rectangle (double length, double breadth)
```

```
{
```

```
l = length; b = breadth;.
```

```

}
public void area ()
{
System.out.println ("Area of a Rectangle is : " + l*b );
}
public void volume ()
{
System.out.println ("Volume of a Rectangle is : " + 2*(l+b));
}
}
class InterfaceDemo
{
public static void main (String args[])
{
Circle ob1 = new Circle (102);
ob1.area (); ob1.volume ();
Rectangle ob2 = new Rectangle (126, 2355); ob2.area
();
ob2.volume ();
}
}

```

Interface vs Abstract Class

An interface is like having a 100% Abstract Class. Interfaces cannot have nonabstract Methods while abstract Classes can. A Class can implement more than one Interface while it can extend only one Class. As abstract Classes come in the hierarchy of Classes, they can extend other Classes while Interface can only extend Interfaces.

The relationship between classes and interfaces

As shown in the figure given below, a class extends another class, an interface extends another interface, but a class implements an interface.

Program:

```

import java.util.Scanner;
interface Vehicles{    void
gear_change(int a);    void
speedup(int b);
    void apply_brakes(int c);
}

```

```

class Bicycle implements Vehicles{
int gear_b,speed_b;   public void
gear_change(int gear){
gear_b=gear;
    }
    public void speedup(int increment){
        speed_b= speed_b + increment;
    }
    public void apply_brakes(int decrement){
speed_b= speed_b - decrement;
    }
    void print(){
        System.out.printf("Current Gear: %d \nSpeed after applying breaks : %d",gear_b,
speed_b);
    }
}

```

```

class Bike implements Vehicles{
int gear_bi,speed_bi;   public void
gear_change(int gear){
    gear_bi = gear;
    }
    public void speedup(int increment){
        speed_bi = speed_bi + increment;
    }
    public void apply_brakes(int decrement){
        speed_bi = speed_bi - decrement;
    }
    void print(){
        System.out.printf("Current Gear: %d \nSpeed after applying breaks : %d",gear_bi,
speed_bi);
    }
}

```

```

class Car implements Vehicles{
int gear_c,speed_c;
    public void gear_change(int gear){
        gear_c = gear;
    }
    public void speedup(int increment){
        speed_c = speed_c + increment;
    }
    public void apply_brakes(int decrement){
        speed_c = speed_c - decrement;
    }
    void print(){
        System.out.printf("Current Gear: %d \nSpeed after applying breaks : %d",gear_c,
speed_c);
    }
}

```

```

class getData{

```

```

    int x;
    int y;    int
    z;
    void get(){
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter Gear : ");
        x = sc.nextInt();
        System.out.print("Enter current Speed : ");
        y = sc.nextInt();
        System.out.print("Enter break Speed : ");
        z = sc.nextInt();
    }
}
public class Interface_5 {
    public static void main(String[] args){
        int choice ;
        getData info = new getData();
        Scanner sc = new Scanner(System.in);
        do {
            System.out.println("Choose type of Vehicle : \n1. Bicycle\n2. Bike\n3. Car\n4. Exit");
            System.out.print("Enter choice : ");
            choice = sc.nextInt();

            switch (choice) {
            case 1:
                Bicycle bc = new Bicycle();
                info.get();
                bc.gear_change(info.x);
                bc.speedup(info.y);
                bc.apply_brakes(info.z);
                System.out.println("\n=====");
                System.out.println("Bicycle Information : ");
                bc.print();
                System.out.println("\n=====");
                break;
            case
            2:
                Bike b = new Bike();
                info.get();
                b.gear_change(info.x);
                b.speedup(info.y);
                b.apply_brakes(info.z);
                System.out.println("\n=====");
                System.out.println("Bike Information : ");          b.print();
                System.out.println("\n=====");
                break;
            case
            3:
                Car c = new Car();
                info.get();
                c.gear_change(info.x);

```

```

        c.speedup(info.y);
        c.apply_brakes(info.z);
        System.out.println("\n=====");
System.out.println("Car Information: ");          c.print();
        System.out.println("\n=====");
break;          case 4:
        System.out.println("\nProgram Exited!!");
break;          default:
        System.out.println("\nEnter choice again!");

```

```

    }
}while (choice!=4);
}
}

```

```

3      import java.util.Scanner;
4      interface Vehicles{
5          void gear_change(int a);
6          void speedup(int b);
7          void apply_brakes(int c);
8      }
9      class Bicycle implements Vehicles{
10         int gear_b,speed_b;
11         public void gear_change(int gear){
12             gear_b=gear;
13         }
14         public void speedup(int increment){
15             speed_b= speed_b + increment;
16         }
17         public void apply_brakes(int decrement){
18             speed_b= speed_b - decrement;
19         }
20         void print(){
21             System.out.printf("Current Gear: %d \nSpeed after applying breaks : %d",gear_b, speed_b);
22         }
23     }
24
25     class Bike implements Vehicles{
26         int gear_bi,speed_bi;
27         public void gear_change(int gear){
28             gear_bi = gear;
29         }
30         public void speedup(int increment){
31             speed_bi = speed_bi + increment;

```

```

32     }
33     public void apply_brakes(int decrement){
34         speed_bi = speed_bi - decrement;
35     }
36     void print(){
37         System.out.printf("Current Gear: %d \nSpeed after applying breaks : %d",gear_bi, speed_bi);
38     }
39 }
40
41 class Car implements Vehicles{
42     int gear_c,speed_c;
43     public void gear_change(int gear){
44         gear_c = gear;
45     }
46     public void speedup(int increment){
47         speed_c = speed_c + increment;
48     }
49     public void apply_brakes(int decrement){
50         speed_c = speed_c - decrement;
51     }
52     void print(){
53         System.out.printf("Current Gear: %d \nSpeed after applying breaks : %d",gear_c, speed_c);
54     }
55 }
56
57 class getData{
58     int x;
59     int y;
60     int z;
61
62     void get(){
63         Scanner sc = new Scanner(System.in);
64
65         System.out.print("Enter Gear : ");
66         x = sc.nextInt();
67         System.out.print("Enter current Speed : ");
68         y = sc.nextInt();
69         System.out.print("Enter break Speed : ");
70         z = sc.nextInt();
71     }
72 }
73 public class Interface_5 {
74     public static void main(String[] args){
75         int choice ;
76         getData info = new getData();
77         Scanner sc = new Scanner(System.in);
78         do {
79             System.out.println("Choose type of Vehicle : \n1. Bicycle\n2. Bike\n3. Car\n4. Exit");
80             System.out.print("Enter choice : ");
81             choice = sc.nextInt();
82
83             switch (choice) {
84                 case 1:
85                     Bicycle bc = new Bicycle();
86                     info.get();
87                     bc.gear_change(info.x);
88                     bc.speedup(info.y);
89                     bc.apply_brakes(info.z);
90                     System.out.println("\n=====");
91                     System.out.println("Bicycle Information : ");

```

```

91         bc.print();
92         System.out.println("\n=====");
93         break;
94
95     case 2:
96         Bike b = new Bike();
97         info.get();
98         b.gear_change(info.x);
99         b.speedup(info.y);
100        b.apply_brakes(info.z);
101        System.out.println("\n=====");
102        System.out.println("Bike Information : ");
103        b.print();
104        System.out.println("\n=====");
105        break;
106
107     case 3:
108        Car c = new Car();
109        info.get();
110        c.gear_change(info.x);
111        c.speedup(info.y);
112        c.apply_brakes(info.z);
113        System.out.println("\n=====");
114        System.out.println("Car Information: ");
115        c.print();
116        System.out.println("\n=====");
117        break;
118
119     case 4:
120        System.out.println("\nProgram Exited!!");
121        break;
122
123     default:
124        System.out.println("\nEnter choice again!");
125
126 }
127 }while (choice!=4);
128 }

```


Output:

```
"C:\Program Files\Java\jdk-16.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA
Choose type of Vehicle :
1. Bicycle
2. Bike
3. Car
4. Exit
Enter choice : 1
Enter Gear : 3
Enter current Speed : 35
Enter break Speed : 10

=====
Bicycle Information :
Current Gear: 3
Speed after applying breaks : 25
=====

Choose type of Vehicle :
1. Bicycle
2. Bike
3. Car
4. Exit
Enter choice : 2
Enter Gear : 2
Enter current Speed : 20
Enter break Speed : 5
```

```
=====
Bike Information :
Current Gear: 2
Speed after applying breaks : 15
=====
Choose type of Vehicle :
1. Bicycle
2. Bike
3. Car
4. Exit
Enter choice : 3
Enter Gear : 3
Enter current Speed : 40
Enter break Speed : 15

=====
Car Information:
Current Gear: 3
Speed after applying breaks : 25
=====
Choose type of Vehicle :
1. Bicycle
2. Bike
3. Car
4. Exit
Enter choice : 7

Enter choice again!
Choose type of Vehicle :
1. Bicycle
2. Bike
3. Car
4. Exit
Enter choice : 4

Program Exited!!

Process finished with exit code 0
|
```

Conclusion: Thus, we have studied interface concept using java