Practical 05

28994-TRDT Dulshan

OOP-JAVA

//E01-1

```
package com.mycompany.practical_05;
public interface MyFirstInterface {
   int x=9;
}
```

//E01-1

```
package com.mycompany.practical_05;
public interface MyFirstInterface {
   public int x=9;
}
```

//E01-1

```
package com.mycompany.practical_05;
public interface MyFirstInterface {
   final x=9;
}
```

//E01-1

```
package com.mycompany.practical_05;
public interface MyFirstInterface {
   static int x=9;
}
```

- There is not any difference between these two approaches because interface doesn't allow to declare any instance variables. It allows declare constant variables only.

//E01-2

```
package com.mycompany.practical_05;
public interface MyFirstInterface {
   int x=9;

   public void display();

}
```

```
package com.mycompany.practical_05;
public interface MyFirstInterface {
   int x=9;

   public abstract void display();

}
```

- There is not any difference between these two approaches because All methods inside an interface are implicitly abstract, and implementing classes must provide concrete implementations for these methods to fulfill the interface contract.

//E01-3

```
package com.mycompany.practical_05;
public class InterfaceImplemented implements MyFirstInterface{
  @Override
  public void display(){
     x=9;
     System.out.println(x);
  }
}
```

Its not possible to change x because It is a constant variable.

//E02

//Interface

```
package com.mycompany.interfaceexample;
public interface Speaker {
    public void speak();

}
```

//Politician

```
package com.mycompany.interfaceexample;
public class Politician implements Speaker{
    @Override
    public void speak(){
        System.out.println("politician");
    }

}
```

//Priest

```
package com.mycompany.interfaceexample;
public class Priest implements Speaker{
    @Override
    public void speak(){
        System.out.println("Priest");
    }
}
```

//Lecturer

```
package com.mycompany.interfaceexample;
public class Lecturer implements Speaker{
   @Override
   public void speak(){
      System.out.println("politician");
   }
}
```

//E04

//Shape class

```
package com.mycompany.e04;
public abstract class Shape {
   public abstract double calculateArea();
   public void display(){
      System.out.println(calculateArea());
   }

}
```

//Circle class

```
package com.mycompany.e04;
public class Circle extends Shape{
   private double radious=10;
   @Override
   public double calculateArea(){
      return Math.PI*radious*radious;
   }
}
```

//Rectangle class

```
package com.mycompany.e04;
public class Rectangle extends Shape{
   private double height=23, width=10;

   @Override
   public double calculateArea(){
      return height*width;
   }
}
```

//Main

```
package com.mycompany.e04;
public class E04 {

   public static void main(String[] args) {
      Shape s1=new Circle();
      Shape s2=new Rectangle();
      s1.calculateArea();
      s1.display();
      s2.calculateArea();
      s2.display();
   }
}
```