```java
package interfaceassignment;

public interface Recolorable {
  void recolor(ShapeColor sc);
}


package interfaceassignment;

public interface Resizable {
  void resize(double percentage);
}


package interfaceassignment;

public interface Shape {
      double area();

}

package interfaceassignment;

public abstract class TwoD implements Recolorable,Shape{
      private int a;
      private ShapeColor sc;
      private int b;
      private int c;
      private int d;
      TwoD(){
            System.out.println("default Condtructor");
      }
  TwoD(ShapeColor sc, int a){
        this.sc=sc;
      this.a=a;
  }
  TwoD(ShapeColor sc,int a,int b){

            /*
             * this.sc=sc; this.a=a;
             */
           this(sc,a);//controller jumps to line number 12 and executes Constructor
           this.b=b;

  }
 TwoD(ShapeColor sc,int a,int b,int c){
            /*
             * this.sc=sc; this.a=a; this.b=b;
             */
          this(sc,a,b);//controller will go to line number 16
        this.c=c;

  }
 TwoD(ShapeColor sc,int a,int b,int c,int d){
```

```
                /*
                 * this.sc=sc; this.a=a; this.b=b; this.c=c;
                 */
               this(sc,a,b,c);
                this.d=d;

 }
  TwoD(TwoD td){
         sc=td.sc;
         a=td.a;
         b=td.b;
         c=td.c;
         d=td.d;
  }
  int getA() {
       return 0;
  }
  int getB() {
              return 0;
         }
  int getC() {
              return 0;
         }
  int getD() {
              return 0;
         }
   ShapeColor getShapeColor() {
       return null;
    }
   void set(ShapeColor sc,int a,int b,int c,int d) {

    }
   public void recolor(ShapeColor sc) {
         // System.out.println(ShapeColor.white);
   }
@Override
public String toString() {
       return "TwoD [a=" + a + ", sc=" + sc + ", b=" + b + ", c=" + c + ", d=" + d +
"]";
}




}
package interfaceassignment;

public class Circle extends TwoD {


         private ShapeColor sc;
         private int radius;

         Circle(){
         } Circle(ShapeColor sc,int radius){
                this.sc=sc;
                this.radius=radius;
                }
         Circle(Circle c){
               sc=c.sc;
               radius=c.radius;

         }

         @Override public double area() {
                return 3.142*radius*radius;
                }

         int getRadius() {
                return radius;
         }
          void set(ShapeColor sc,int radius) {
               this.sc=sc;
               this.radius=radius;
         }
       @Override
       public String toString() {
              return "Circle [sc=" + sc + ", radius=" + radius + "]";
       }

}
package interfaceassignment;

public class Rectangle extends TwoD {
    private ShapeColor sc;
       private int length;
       private int width;
       Rectangle(){

    }
    Rectangle(ShapeColor sc,int length,int width){
       this.sc=sc;
       this.length=length;
       this.width=width;
    }
    Rectangle(Rectangle r){
       sc=r.sc;
       length=r.length;
       width=r.width;

    }
       @Override
       public double area() {
              return length*width;
       }

       int getLength() {
              return length;
```

```
        }
        int getWidth() {
                return width;

        }
        void set(ShapeColor sc,int length,int width) {
                this.sc=sc;
                this.length=length;
                this.width=width;

        }

        @Override
        public String toString() {
                return "Rectangle [sc=" + sc + ", length=" + length + ", width=" + width
+ "]";
        }



}


package interfaceassignment;

public class Triangle extends TwoD {
          private ShapeColor sc;
          private int a;
          private int b;
          private int c;
        Triangle(){
        System.out.println("Default Constructor");
     }
     Triangle(ShapeColor sc,int a,int b,int c){

                  this.sc=sc;
                  this.a=a;
                  this.b=b;
                  this.c=c;
     }
     Triangle(Triangle t){
        sc=t.sc;
        a=t.a;
        b=t.b;
        c=t.c;
     }
        @Override
        public double area() {
                return 1/2*a*b;
        }

        int getA() {
                return a;
        }
        int getB() {
                return b;
        }
        int getC() {
                return c;
        }
        void set(ShapeColor sc,int a,int b,int c) {
                this.sc=sc;
                this.a=a;
                this.b=b;
                this.c=c;

        }
        @Override
        public String toString() {
                return "Triangle [sc=" + sc + ", a=" + a + ", b=" + b + ", c=" + c +
"]";
        }




}


package interfaceassignment;

public class Trapezoid extends TwoD {
    private ShapeColor sc;
        private int a;
        private int b;
        private int c;
        private int d;
        Trapezoid(){
        System.out.println("Defaul Constructor");
    }
    Trapezoid(ShapeColor sc, int a,int b,int c,int d){
        this.sc=sc;
        this.a=a;
        this.b=b;
        this.c=c;
        this.d=d;
    }
    int getA() {
                return a;

    }
    int getB() {
                return b;

    }
    int getC() {
                return c;
    }
    int getD() {
            return d;

    }
    int getHeight() {
            return 0;

    }
      @Override
      public double area() {
            return 1/2*(a+b)*c;
      }
      void set(ShapeColor sc,int a,int b,int c,int d) {
            this.sc=sc;
      this.a=a;
      this.b=b;
      this.c=c;
      this.d=d;

      }

      @Override
      public String toString() {
            return "Trapezoid [sc=" + sc + ", a=" + a + ", b=" + b + ", c=" + c + ",
d=" + d + "]";
      }



}

package interfaceassignment;

public enum ShapeColor {
      blue,yellow,red,green,white;

}

package interfaceassignment;

public class TestShapeColor {

      public static void main(String[] args) {
      ShapeColor sc;
      sc=ShapeColor.red;
      System.out.println(sc);
      sc=ShapeColor.blue;
      System.out.println(sc);
      sc=ShapeColor.yellow;
      System.out.println(sc);
      sc=ShapeColor.green;
      System.out.println(sc);
      sc=ShapeColor.white;
      System.out.println(sc);
```

```
        }

}


package interfaceassignment;

public class TestTwoD {
        static Shape getInstance(String shape){
                if(shape.equalsIgnoreCase("Circle")) {
                        return new Circle();
                }else if(shape.equalsIgnoreCase("Rectangle") ) {
                        return new Rectangle(ShapeColor.blue,10,20);
                }else if(shape.equalsIgnoreCase("Trianle") ) {
                        return new Triangle();
                }else if(shape.equalsIgnoreCase("TrapeZoid") ) {
                        return new Trapezoid();

                }else {
                        return null;
                }
        }
        public static void main(String[] args) {
                Shape shape=getInstance("Rectangle");
                System.out.println(shape.area());


        }

}
```