

Scala Programming

(25/09/2023)

Name: Niranjana J

USN: 22BTRAD027

Branch: CSE- AI& DE

Question:

Write a Scala program that creates an enum class Color with values for different colors. Use the enum class to represent an object's color

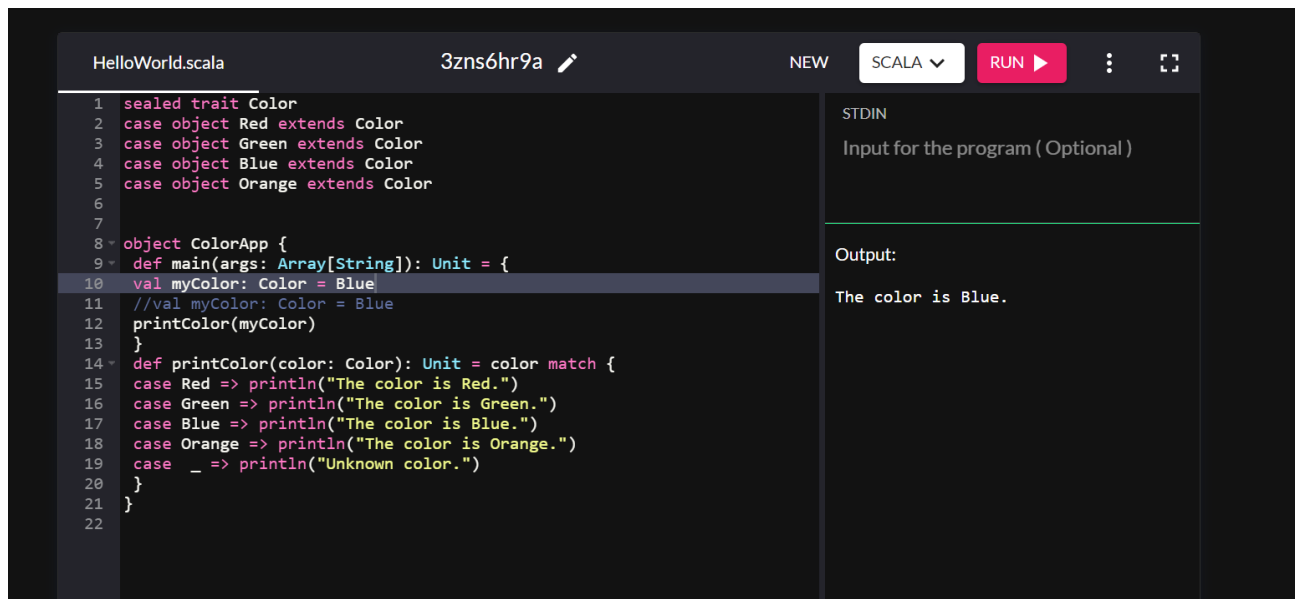
Code:

```
sealed trait Color
case object Red extends Color
case object Green extends Color
case object Blue extends Color
case object Orange extends Color

object ColorApp {
  def main(args: Array[String]): Unit = {
    val myColor: Color = Blue
    printColor(myColor)
  }

  def printColor(color: Color): Unit = color match {
    case Red => println("The color is Red.")
    case Green => println("The color is Green.")
    case Blue => println("The color is Blue.")
    case Orange => println("The color is Orange.")
    case _ => println("Unknown color.")
  }
}
```

Output:



The screenshot shows a Scala IDE with a file named 'HelloWorld.scala'. The code defines a sealed trait 'Color' with case objects 'Red', 'Green', 'Blue', and 'Orange'. An object 'ColorApp' contains a 'main' function that takes an array of strings and a 'printColor' function that uses a match statement to print the color name. In the 'main' function, an immutable variable 'myColor' is set to 'Blue' and passed to 'printColor'. The IDE interface includes a 'RUN' button and a 'STDOUT' panel on the right, which displays the output 'The color is Blue.'

```
1 sealed trait Color
2 case object Red extends Color
3 case object Green extends Color
4 case object Blue extends Color
5 case object Orange extends Color
6
7
8 object ColorApp {
9   def main(args: Array[String]): Unit = {
10    val myColor: Color = Blue
11    //val myColor: Color = Blue
12    printColor(myColor)
13  }
14  def printColor(color: Color): Unit = color match {
15    case Red => println("The color is Red.")
16    case Green => println("The color is Green.")
17    case Blue => println("The color is Blue.")
18    case Orange => println("The color is Orange.")
19    case _ => println("Unknown color.")
20  }
21 }
22
```

Output:

The color is Blue.

In this program, we are creating a sealed trait named 'Color' and creating case objects of different COLOR names including red, blue, green etc. In the object ColorApp, we are creating a immutable variable (val) of type Color and passes it to the function printcolor. In the function printcolor if the case given is Blue(in this example), the function gives an output "The color is blue".

In the next example, if we create a case object named purple, set the value of the immutable variable myColor to Purple, but don't define its case in the function printColor, the function will give an output of "Unknown color" because it will belong to the case _ .

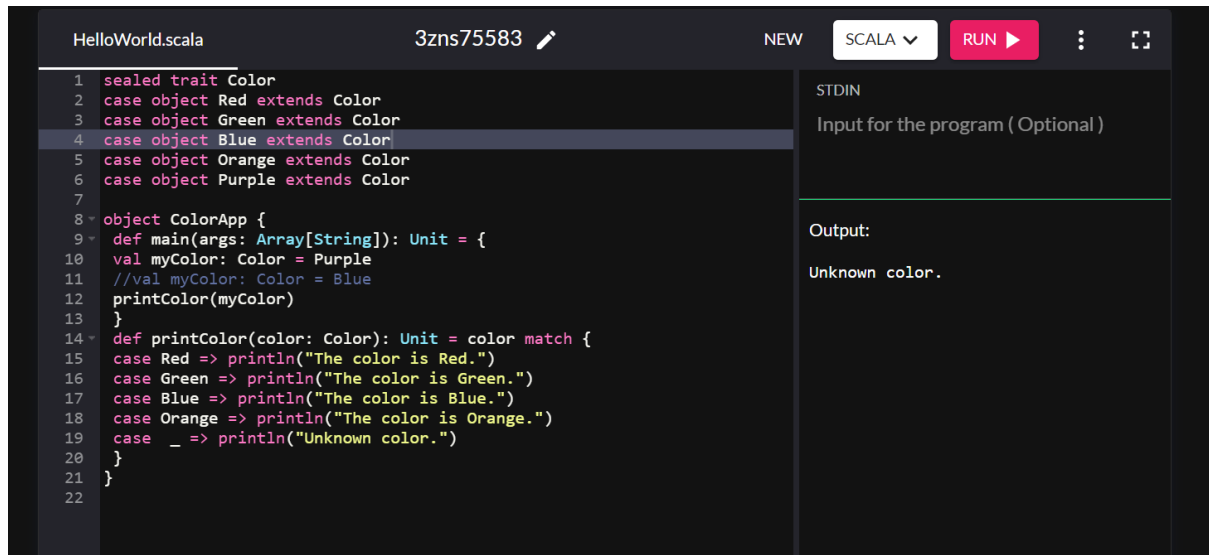
Code:

```
sealed trait Color
case object Red extends Color
case object Green extends Color
case object Blue extends Color
case object Orange extends Color
case object Purple extends Color

object ColorApp {
  def main(args: Array[String]): Unit = {
    val myColor: Color = Purple
    //val myColor: Color = Blue
    printColor(myColor)
  }
  def printColor(color: Color): Unit = color match {
    case Red => println("The color is Red.")
    case Green => println("The color is Green.")
```

```
case Blue => println("The color is Blue.")
case Orange => println("The color is Orange.")
case _ => println("Unknown color.")
}
}
```

Output:



The screenshot shows a Scala IDE interface with a file named 'HelloWorld.scala'. The code defines a sealed trait 'Color' with objects 'Red', 'Green', 'Blue', 'Orange', and 'Purple' extending it. A 'ColorApp' object contains a 'main' function that takes an array of strings, creates a 'myColor' variable (commented as 'Purple' but set to 'Blue'), and calls 'printColor'. The 'printColor' function uses a match expression to print the color name. The IDE shows the file is open in a new editor, and the 'RUN' button is visible. The output pane on the right shows 'Unknown color.'

```
1 sealed trait Color
2 case object Red extends Color
3 case object Green extends Color
4 case object Blue extends Color
5 case object Orange extends Color
6 case object Purple extends Color
7
8 object ColorApp {
9   def main(args: Array[String]): Unit = {
10    val myColor: Color = Purple
11    //val myColor: Color = Blue
12    printColor(myColor)
13  }
14  def printColor(color: Color): Unit = color match {
15    case Red => println("The color is Red.")
16    case Green => println("The color is Green.")
17    case Blue => println("The color is Blue.")
18    case Orange => println("The color is Orange.")
19    case _ => println("Unknown color.")
20  }
21 }
22
```

STDIN

Input for the program (Optional)

Output:

Unknown color.