```java
//Author, Student-number: Seyit Uzun, 23394725; Koray Hacibayramoglu, 23409691

public class Color {

    private int rgb;

    // Constructor for the direct transfer of the int representation of the color
    public Color(int rgb) {
        this.rgb = rgb;
    }

    // Constructor for passing the color values red, green and blue
    public Color(int red, int green, int blue) {
        if (!isValidColorValue(red) || !isValidColorValue(green) || !isValidColorValue(blue)) {
            System.err.println("Invalid color values! Values are set to interval limits.");
            red = clampColorValue(red);
            green = clampColorValue(green);
            blue = clampColorValue(blue);
        }
        this.rgb = (red << 16) | (green << 8) | blue;
    }

    // Default constructor for the color black
    public Color() {
        this.rgb = 0;
    }

    // Get-methode for rgb
    public int getRgb() {
        return rgb;
    }

    // Get-method for the color channels
    public int getRed() {
        return (rgb >> 16) & 0xFF;
    }

    public int getGreen() {
        return (rgb >> 8) & 0xFF;
    }

    public int getBlue() {
        return rgb & 0xFF;
```

```
44      }
45
46      // Method for converting to hexadecimal representation
47      public String getHex() {
48          return String.format("#%06X", rgb);
49      }
50
51      // Constructor for the hexadecimal representation of the color
52      public Color(String hex) {
53          hex = hex.replace("#", "");
54          this.rgb = Integer.parseInt(hex, 16);
55      }
56
57      // Overwriting the toString() method
58      @Override
59      public String toString() {
60          return getHex();
61      }
62
63      // Method for calculating the complementary color
64      public Color complementaryColor() {
65          int red = 255 - getRed();
66          int green = 255 - getGreen();
67          int blue = 255 - getBlue();
68          return new Color(red, green, blue);
69      }
70
71      // Method for mixing colors
72      public Color mixColor(Color color) {
73          int red = (getRed() + color.getRed()) / 2;
74          int green = (getGreen() + color.getGreen()) / 2;
75          int blue = (getBlue() + color.getBlue()) / 2;
76          return new Color(red, green, blue);
77      }
78
79      // Auxiliary method for checking valid color values
80      private boolean isValidColorValue(int value) {
81          return value >= 0 && value <= 255;
82      }
83
84      // Auxiliary method for limiting color values to the interval [0, 255]
85      private int clampColorValue(int value) {
86          return Math.max(0, Math.min(value, 255));
```

```
87      }
88
89
90      // Static constants for frequently used colors
91      public static final Color BLACK = new Color(0, 0, 0);
92      public static final Color WHITE = new Color(255, 255, 255);
93      public static final Color GRAY = new Color(128, 128, 128);
94      public static final Color RED = new Color(255, 0, 0);
95      public static final Color GREEN = new Color(0, 255, 0);
96      public static final Color BLUE = new Color(0, 0, 255);
97
98      public static void main(String[] args) {
99          Color turquoise = new Color("#40E0D0"); // Turquoise
100         Color red = Color.RED;
101         Color lightGreen = new Color("#90EE90"); // Light Green
102         Color green = Color.GREEN;
103         Color darkGreen = new Color("#006400"); // Dark Green
104         Color peachPuff = new Color("#FFDAB9"); // PeachPuff
105         Color blueViolet = new Color("#8A2BE2");
106         Color whiteSmoke = new Color("#F5F5F5");
107         Color powderBlue = new Color("#B0E0E6");
108         Color gold = new Color("#FFD700");
109
110         new ColorVisualizer(turquoise);
111         new ColorVisualizer(red);
112         new ColorVisualizer(lightGreen);
113         new ColorVisualizer(green);
114         new ColorVisualizer(darkGreen);
115         new ColorVisualizer(peachPuff);
116         new ColorVisualizer(blueViolet);
117         new ColorVisualizer(whiteSmoke);
118         new ColorVisualizer(powderBlue);
119         new ColorVisualizer(gold);
120     }
121 }
```