

```
1 import java.util.HashMap;
2 //Name: Till Bessermann, Matrikelnummer: 23303124
3 //Name: Hamza Jaouadi, Matrikelnummer: 23414609
4
5 public class Color {
6
7     //Variablendeklaration
8     private int rgb;
9     public static final Color GRAY = new Color(128, 128, 128);
10    public static final Color WHITE = new Color(255, 255, 255);
11    public static final Color BLACK = new Color(0, 0, 0);
12    public static final Color RED = new Color (255, 0,0);
13    public static final Color BLUE = new Color (0, 0, 255);
14    public static final Color GREEN = new Color (0, 255, 0);
15    public Color (int rgb) {
16
17        this.rgb = rgb;
18    }
19
20    //Konstruktor mit drei ?bergabeparametern
21    public Color (int red, int green, int blue) {
22
23        if (0 > red) {
24            System.out.println("Error! Value out of bounds!");
25            red = 0;
26        } else if (255 < red) {
27            System.out.println("Error! Value out of bounds!");
28            red = 255;
29        } else if (0 > green) {
30            System.out.println("Error! Value out of bounds!");
31            green = 0;
32        } else if (255 < green) {
33            System.out.println("Error! Value out of bounds!");
34            green = 255;
35        } else if (0 > blue) {
36            System.out.println("Error! Value out of bounds!");
37            blue = 0;
38        } else if (255 < blue) {
39            System.out.println("Error! Value out of bounds!");
40            blue = 255;
41        }
42
43        this.rgb = (red << 16) | (green << 8) | blue;
```

```
44     }
45
46     //Konstruktor ohne ?bergabeparameter
47     public Color () {
48
49         this.rgb = 0;
50     }
51
52     //Konstruktor mit String als ?bergabeparameter
53     public Color (String hex) {
54
55         String hexa = hex.substring(1);
56         this.rgb = Integer.parseInt(hexa, 16);
57         System.out.println(rgb);
58     }
59
60     public int getRgb() {
61
62         return rgb;
63     }
64
65     public int getRed() {
66
67         int red = rgb >> 16 ;
68         return red;
69     }
70
71     public int getGreen() {
72
73         int green = rgb >> 8 & 0xFF;
74         return green;
75     }
76
77     public int getBlue() {
78
79         int blue = rgb & 0xFF;
80         return blue;
81     }
82
83     public String getHex() {
84
85         return "#" + String.format("%06X", rgb);
86     }
```

//nur die unteren 8 bits werden beruecksichtigt, die oberen werden zu 0

```
87
88  @Override
89  public String toString() {
90
91      return getHex();
92  }
93
94  //Methode für die Komplementärfarbe
95  public Color complementaryColor () {
96
97      Color complementColor = new Color(255 - getRed(), 255 - getGreen(), 255 - getBlue());
98      return complementColor;
99  }
100
101  //Methode zum Farbenmischen
102  public Color mixColor (Color color) {
103
104      int rneu = (this.getRed() + color.getRed()) / 2;
105      int gneu = (this.getGreen() + color.getGreen()) / 2;
106      int bneu = (this.getBlue() + color.getBlue()) / 2;
107      Color mixedColor = new Color(rneu, gneu, bneu);
108      return mixedColor;
109  }
110
111  public static void main(String[] args) {
112
113      //Farbobjekte
114      Color deepSkyBlue = new Color("#00BFFF");
115      Color orangeRed = new Color("#FF4500");
116      Color turquoise = new Color("#40E0D0");
117      Color olive = new Color("#808000");
118      Color peachPuff = new Color("#FFDAB9");
119      Color orange = new Color("#FFA500");
120
121      //Datenausgabe der Farben
122      System.out.println("Red rate of DeepSkyBlue: " + deepSkyBlue.getRed());
123      System.out.println("Green rate of DeepSkyBlue: " + deepSkyBlue.getGreen());
124      System.out.println("Blue rate of DeepSkyBlue: " + deepSkyBlue.getBlue());
125      System.out.println("Hexadecimal of DeepSkyBlue: " + deepSkyBlue.getHex());
126      System.out.println("Hexadecimal of PeachPuff: " + peachPuff.getHex());
127
128      //Visualisierungsobjekte
129      ColorVisualizer deepSkyBlueVisual = new ColorVisualizer(deepSkyBlue);
```

```
130     ColorVisualizer orangeRedVisual = new ColorVisualizer(orangeRed);
131     ColorVisualizer turquoiseVisual = new ColorVisualizer(turquoise);
132
133     //Visualisierung der Komplementaerfarbe
134     ColorVisualizer deepSkyBlueComplement = new ColorVisualizer(deepSkyBlue.complementaryColor());
135     ColorVisualizer orangeRedComplement = new ColorVisualizer(orangeRed.complementaryColor());
136
137     //Visualisierung der Mischfarbe
138     ColorVisualizer deepSkyBlueMixed = new ColorVisualizer(deepSkyBlue.mixColor(WHITE));
139     ColorVisualizer oliveMixed = new ColorVisualizer(olive.mixColor(GRAY));
140 }
141 }
```