```java
//Author: Muhammad Ridwan Bagindo, 23363816
//Author: Pascal-Andre Stifter, 23117434

public class Color {
    private int rgb;     //Konstruktor fuer die int-Darstellung der Farbe

    public Color(int rgb) {      //Konstruktor mit den Farbkanaele
        this.rgb = rgb;
    }

    public Color(int red, int green, int blue) {
        if (!isValidColorValue(red) || !isValidColorValue(green) || !isValidColorValue(blue)) {
            System.err.println("Invalid values detected. Readjusting values to existing parameters [0, 255]!");

            red = correctColorValue(red);
            green = correctColorValue(green);
            blue = correctColorValue(blue);
        }

        this.rgb = (red << 16) | (green << 8) | blue;
    }

    private boolean isValidColorValue(int value) {   //ueberprueft ob der Farbwert im Intervall liegt
        return value >= 0 && value <= 255;
    }

    private int correctColorValue(int value) {   //korrigiert ein ungueltiges Farbintervall eine der Intervallgrenzen[0,255]
        return Math.min(Math.max(value, 0), 255);
    }

    public Color() {     //Standard-Konstruktor fuer die Farbe Schwarz
        this.rgb = 0;
    }

    public int getRgb() {    //oeffentliche Methode um rgb auszulesen
        return rgb;
    }

    //oeffentliche Getter fuer die Farbkanaele
    public int getRed() {
        return (rgb >> 16) & 0xFF;
    }

```

```
44      public int getGreen() {
45          return (rgb >> 8) & 0xFF;
46      }
47
48      public int getBlue() {
49          return rgb & 0xFF;
50      }
51
52      public String getHex() {     //oeffentliche Methode um die Hexadezimaldarstellung zu lesen
53          String hex = Integer.toHexString(rgb).toUpperCase();
54
55          while (hex.length() < 6) {
56              hex = "0" + hex;
57          }
58
59          return "#" + hex;
60      }
61
62      public Color(String hex) {  //Konstruktor uebernimmt die Hexadezimaldarstellung der Farbe als String mit #
63          if (hex.startsWith("#")) {
64              hex = hex.substring(1); // entfernt das Praefix
65          }
66
67          this.rgb = Integer.parseInt(hex, 16);    // speichert die Hexadezimalzahl im rgb
68      }
69
70      @Override    //ueberschreibung der toString()-Methode
71      public String toString() {
72          return getHex();
73      }
74
75      public Color complementaryColor() { //oeffentliche Methode um Komplementaerfarben zu berechnen
76          int Red = 255 - getRed();
77          int Green = 255 - getGreen();
78          int Blue = 255 - getBlue();
79
80          return new Color(Red, Green, Blue);
81      }
82
83      public Color mixColor(Color color) {     //oeffentliche Methode zum Mischen der Farben
84          int mixRed = (getRed() + color.getRed()) / 2;
85          int mixGreen = (getGreen() + color.getGreen()) / 2;
86          int mixBlue = (getBlue() + color.getBlue()) / 2;
```

```
87
88          return new Color(mixRed, mixGreen, mixBlue);
89      }
90
91      //Definition der haeufig verwendeten Farben
92      public static final Color BLACK = new Color(0, 0, 0);
93      public static final Color WHITE = new Color(255, 255, 255);
94      public static final Color GRAY = new Color(128, 128, 128);
95      public static final Color RED = new Color(255, 0, 0);
96      public static final Color GREEN = new Color(0, 255, 0);
97      public static final Color BLUE = new Color(0, 0, 255);
98
99      //Main zum Testen
100     public static void main(String[] args) {
101         //Test des RGB Konstruktors
102         Color color1 = new Color(8421504);
103         System.out.println("RGB value: " + color1.getRgb());
104
105         //Test des RGB Konstruktors zum Vergleichen mit den korrigierten
106         Color color2 = new Color(255, 0, 255);
107         System.out.println("Border RGB value: " + color2.getRgb());
108
109         //Test des Standard-Konstruktors
110         Color color3 = new Color();
111         System.out.println("Black RGB value: " + color3.getRgb());
112
113         //Test des Konstruktors mit ungueltigen RGB-Werten
114         Color color4 = new Color(300, -50, 1200);
115         System.out.println("Adjusted RBG value: " + color4.getRgb());
116
117         /*
118         //Test fuer den Visualizer
119         new ColorVisualizer(BLACK);
120         new ColorVisualizer(WHITE);
121         new ColorVisualizer(GRAY);
122         new ColorVisualizer(RED);
123         new ColorVisualizer(GREEN);
124         new ColorVisualizer(BLUE);
125          */
126
127         //weitere Farben zum Testen
128         Color aliceBlue = new Color("#F0F8FF");
129         Color chartreuse = new Color("#7FFF00");
```

```
130         Color cadetBlue = new Color("#5F9EA0");
131         Color hotPink = new Color("#FF69B4");
132         Color navajoWhite = new Color("#778899");
133         Color lightSlateGray = new Color("#FFDEAD");
134
135         //Tests mit zusaetzlichen Farben aus der Farbtabelle
136         System.out.println("-------------");
137         System.out.println("Alice Blue");
138         System.out.println("RGB value: " + aliceBlue.getRgb());
139         System.out.println("Individual RGB values: " + aliceBlue.getRed() + " " + aliceBlue.getGreen() + " " +
aliceBlue.getBlue());
140         System.out.println("Hex value: " + aliceBlue.getHex());
141
142         // Test der Komplementaerfarbe
143         Color compAliceBlue = aliceBlue.complementaryColor();
144         System.out.println("\nComplementary color");
145         System.out.println("RGB value: " + compAliceBlue.getRgb());
146         System.out.println("Individual RGB values: " + compAliceBlue.getRed() + " " + compAliceBlue.getGreen() + " " +
compAliceBlue.getBlue());
147         System.out.println("Hex value: " + compAliceBlue.getHex());
148
149         System.out.println("-------------");
150         System.out.println("Chartreuse");
151         System.out.println("RGB value: " + chartreuse.getRgb());
152         System.out.println("Individual RGB values: " + chartreuse.getRed() + " " + chartreuse.getGreen() + " " +
chartreuse.getBlue());
153         System.out.println("Hex value: " + chartreuse.getHex());
154
155         Color compChartreuse = chartreuse.complementaryColor();
156         System.out.println("\nComplementary color");
157         System.out.println("RGB value: " + compChartreuse.getRgb());
158         System.out.println("Individual RGB values: " + compChartreuse.getRed() + " " + compChartreuse.getGreen() + " " +
compChartreuse.getBlue());
159         System.out.println("Hex value: " + compChartreuse.getHex());
160
161         System.out.println("-------------");
162         System.out.println("Cadet Blue");
163         System.out.println("RGB value: " + cadetBlue.getRgb());
164         System.out.println("Individual RGB values: " + cadetBlue.getRed() + " " + cadetBlue.getGreen() + " " +
cadetBlue.getBlue());
165         System.out.println("Hex value: " + cadetBlue.getHex());
166
167         Color compCadetBlue = cadetBlue.complementaryColor();
```

```
168         System.out.println("\nComplementary color");
169         System.out.println("RGB value: " + compCadetBlue.getRgb());
170         System.out.println("Individual RGB values: " + compCadetBlue.getRed() + " " + compCadetBlue.getGreen() + " " +
compCadetBlue.getBlue());
171         System.out.println("Hex value: " + compCadetBlue.getHex());
172 
173         System.out.println("-------------");
174         System.out.println("Hot Pink");
175         System.out.println("RGB value: " + hotPink.getRgb());
176         System.out.println("Individual RGB values: " + hotPink.getRed() + " " + hotPink.getGreen() + " " + hotPink.getBlue());
177         System.out.println("Hex value: " + hotPink.getHex());
178 
179         Color compHotPink = hotPink.complementaryColor();
180         System.out.println("\nComplementary color");
181         System.out.println("RGB value: " + compHotPink.getRgb());
182         System.out.println("Individual RGB values: " + compHotPink.getRed() + " " + compHotPink.getGreen() + " " +
compHotPink.getBlue());
183         System.out.println("Hex value: " + compHotPink.getHex());
184 
185         System.out.println("-------------");
186         System.out.println("Navajo White");
187         System.out.println("RGB value: " + navajoWhite.getRgb());
188         System.out.println("Individual RGB values: " + navajoWhite.getRed() + " " + navajoWhite.getGreen() + " " +
navajoWhite.getBlue());
189         System.out.println("Hex value: " + navajoWhite.getHex());
190 
191         Color compNavajoWhite = navajoWhite.complementaryColor();
192         System.out.println("\nComplementary color");
193         System.out.println("RGB value: " + compNavajoWhite.getRgb());
194         System.out.println("Individual RGB values: " + compNavajoWhite.getRed() + " " + compNavajoWhite.getGreen() + " " +
compNavajoWhite.getBlue());
195         System.out.println("Hex value: " + compNavajoWhite.getHex());
196 
197         System.out.println("-------------");
198         System.out.println("Light Slate Gray");
199         System.out.println("RGB value: " + lightSlateGray.getRgb());
200         System.out.println("Individual RGB values: " + lightSlateGray.getRed() + " " + lightSlateGray.getGreen() + " " +
lightSlateGray.getBlue());
201         System.out.println("Hex value: " + lightSlateGray.getHex());
202 
203         Color compLightSlateGray = lightSlateGray.complementaryColor();
204         System.out.println("\nComplementary color");
205         System.out.println("RGB value: " + compLightSlateGray.getRgb());
```

```
206         System.out.println("Individual RGB values: " + compLightSlateGray.getRed() + " " + compLightSlateGray.getGreen() + " "
+ compLightSlateGray.getBlue());
207         System.out.println("Hex value: " + compLightSlateGray.getHex());
208 
209         //Test des Mischens von Farben
210         Color mixedColor = aliceBlue.mixColor(chartreuse);
211 
212         System.out.println("-------------");
213         System.out.println("Alice Blue + Chartreuse");
214         System.out.println("RGB value: " + mixedColor.getRgb());
215         System.out.println("Individual RBG values: " + mixedColor.getRed() + " " + mixedColor.getGreen() + " " +
mixedColor.getBlue());
216         System.out.println("Hex value: " + mixedColor.getHex());
217     }
218 
219 }
```