```
1
     #ifndef COLOR H
2
     #define COLOR H
3
 4
    #include <iostream>
    #include <string>
 5
 6
7
    class Color {
8
         public:
9
10
             enum class Code {
11
                WHITE,
12
                 BLUE,
13
                 GREEN,
14
                 RED,
15
                 BLACK
16
             };
17
             /**
18
19
              ^{\star} @brief Create a color with a color by code (enum) or with nothing
20
              * @param code default value WHITE
              */
21
22
             Color(Code code = Code::WHITE);
23
24
25
              * @brief Get the color
26
              * @return The code of the color (enum)
27
28
             Code getColorCode() const;
29
30
31
              * @brief Change the color
              * @param code
32
              * @return The color
33
              * /
34
35
             Color& setColor(Code code);
36
             /**
37
38
              * @brief Get the name of the color
39
              * @return The name of the color
40
              */
41
             std::string getName() const;
42
43
             /**
              * @brief Display a color
44
45
              * @param stream
              * @return The srteam
46
              */
47
             std::ostream& display(std::ostream& stream = std::cout) const;
48
49
50
         private:
51
            Code mCode;
52
   };
53
54
     * @brief Convert the code color in a string
55
56
     * @param code
57
     * @return The color in string
58
59
    std::string toString(const Color::Code& code);
60
61
     * @brief Overload of the output sream to display a color
62
     * @param stream
63
     * @param color
64
     * @return The stream
65
66
67
    std::ostream& operator<<(std::ostream& stream, const Color& color);</pre>
68
69
    #endif // COLOR_H
70
```