

Flyweight

```
1 package edu.cpp.cs5800.flyweight;
2 import static edu.cpp.cs5800.flyweight.CharacterFormat.*;
3
4 public class Driver {
5     public static void main(String[] args) {
6         String docFileName = "document.txt";
7
8         Document document = createDocument();
9         System.out.print(document.getContent());
10
11         document.save(docFileName);
12         System.out.println(">> Successfully saved the document file: " + docFileName + "\n"
13     );
14
15         Document loadedDoc = loadDocument(docFileName);
16         System.out.println(loadedDoc.getContent());
17
18         editDocument(loadedDoc);
19         System.out.println(loadedDoc.getContent());
20     }
21
22     private static Document createDocument() {
23         System.out.println(">> Creating a new document");
24         Document document = new Document();
25         document.addCharacter('H', Font.ARIAL, Color.RED, 12);
26         document.addCharacter('e', Font.ARIAL, Color.BLACK, 12);
27         document.addCharacter('l', Font.CALIBRI, Color.BLUE, 14);
28         document.addCharacter('l', Font.CALIBRI, Color.BLUE, 14);
29         document.addCharacter('o', Font.VERDANA, Color.BLACK, 16);
30         document.addCharacter('W', Font.VERDANA, Color.RED, 16);
31         document.addCharacter('o', Font.ARIAL, Color.RED, 12);
32         document.addCharacter('r', Font.CALIBRI, Color.BLUE, 14);
33         document.addCharacter('l', Font.VERDANA, Color.BLACK, 16);
34         document.addCharacter('d', Font.ARIAL, Color.BLACK, 12);
35         document.addCharacter('C', Font.CALIBRI, Color.BLUE, 14);
36         document.addCharacter('S', Font.VERDANA, Color.BLACK, 16);
37         document.addCharacter('5', Font.ARIAL, Color.RED, 12);
38         document.addCharacter('8', Font.CALIBRI, Color.BLUE, 14);
39         document.addCharacter('0', Font.VERDANA, Color.RED, 16);
40         document.addCharacter('0', Font.ARIAL, Color.BLACK, 12);
41         return document;
42     }
43
44     private static Document loadDocument(String fileName) {
45         System.out.println(">> Loading the document from file: " + fileName);
46         Document document = new Document();
47         document.load(fileName);
48         return document;
49     }
50
51     private static void editDocument(Document document) {
52         System.out.println(">> Editing document");
```

Flyweight

```
53     document.edit(1, 'E', Font.CALIBRI, Color.BLUE, 16);
54     document.edit(2, 'L', Font.ARIAL, Color.BLUE, 12);
55     document.edit(3, 'L', Font.ARIAL, Color.BLACK, 14);
56     document.edit(4, 'O', Font.VERDANA, Color.RED, 12);
57 }
58 }
59
```

```
1 package edu.cpp.cs5800.flyweight;
2
3 import java.io.*;
4 import java.util.ArrayList;
5 import java.util.List;
6
7 public class Document {
8     private List<Character> characters = new ArrayList<>();
9
10    public boolean load(String path) {
11        characters.clear();
12        try {
13            BufferedReader reader = new BufferedReader(new FileReader(path));
14            String line;
15            while ((line = reader.readLine()) != null) {
16                Character ch = parseCharacter(line);
17                if (ch != null) {
18                    this.addCharacter(ch);
19                }
20            }
21        } catch (IOException e) {
22            return false;
23        }
24
25        return true;
26    }
27
28    public boolean addCharacter(char symbol,
29                               CharacterFormat.Font font,
30                               CharacterFormat.Color color,
31                               int size) {
32        Character ch = CharacterFactory.getCharacter(symbol, font, color, size);
33        return characters.add(ch);
34    }
35
36    public boolean addCharacter(Character character) {
37        return characters.add(character);
38    }
39
40    public Character edit(int index,
41                          char symbol,
42                          CharacterFormat.Font font,
43                          CharacterFormat.Color color,
44                          int size) {
45        Character ch = CharacterFactory.getCharacter(symbol, font, color, size);
46        return characters.set(index, ch);
47    }
48
49    public boolean save(String path) {
50        try (BufferedWriter writer = new BufferedWriter(new FileWriter(path))) {
51            for (Character character : characters) {
52                writer.write(character.getSymbol() + "," + character.getFormat()
53                    .getFont() + "," + character.getFormat()
```

Flyweight

```
54         .getColor() + "," + character.getFormat().getFontSize());
55         writer.newLine();
56     }
57     } catch (IOException e) {
58         return false;
59     }
60     return true;
61 }
62
63 public String getContent() {
64     StringBuilder content = new StringBuilder();
65     for (Character character : characters) {
66         content.append(character);
67         content.append('\n');
68     }
69     return content.toString();
70 }
71
72 private Character parseCharacter(String content) {
73     String[] parts = content.split(",");
74     if (parts.length == 4) {
75         char symbol = parts[0].charAt(0);
76         CharacterFormat.Font font = CharacterFormat.Font.valueOf(parts[1]);
77         CharacterFormat.Color color = CharacterFormat.Color.valueOf(parts[2]);
78         int size = Integer.parseInt(parts[3]);
79         return CharacterFactory.getCharacter(symbol, font, color, size);
80     } else {
81         return null;
82     }
83 }
84 }
85
```

Flyweight

```
1 package edu.cpp.cs5800.flyweight;
2
3 public class Character {
4     private final char symbol;
5     private final CharacterFormat format;
6
7     public Character(char symbol, CharacterFormat format) {
8         this.symbol = symbol;
9         this.format = format;
10    }
11
12    public char getSymbol() {
13        return symbol;
14    }
15    public CharacterFormat getFormat() {
16        return format;
17    }
18
19    @Override
20    public String toString() {
21        return symbol + "(" + format + ")";
22    }
23 }
24
```

Flyweight

```
1 package edu.cpp.cs5800.flyweight;
2
3 public class CharacterFormat {
4     public enum Color {
5         RED, BLACK, BLUE
6     }
7
8     public enum Font {
9         ARIAL, CALIBRI, VERDANA
10    }
11
12    private final Color color;
13    private final Font font;
14    private final int fontSize;
15
16    public CharacterFormat(Font font, Color color, int fontSize) {
17        this.color = color;
18        this.font = font;
19        this.fontSize = fontSize;
20    }
21
22    public Color getColor() {
23        return color;
24    }
25
26    public Font getFont() {
27        return font;
28    }
29
30    public int getFontSize() {
31        return fontSize;
32    }
33
34    @Override
35    public String toString() {
36        return font + "," + color + "," + fontSize;
37    }
38 }
39
```

Flyweight

```
1 package edu.cpp.cs5800.flyweight;
2
3 import java.util.HashMap;
4 import java.util.Map;
5
6 public class CharacterFactory {
7     private static Map<String, CharacterFormat> formats = new HashMap<>();
8
9     public static Character getCharacter(char symbol, CharacterFormat.Font font,
10     CharacterFormat.Color color, int fontSize) {
11         String formatType = font.toString() + color.toString() + fontSize;
12         CharacterFormat format = formats.get(formatType);
13         if (format == null) {
14             format = new CharacterFormat(font, color, fontSize);
15         }
16         return new Character(symbol, format);
17     }
18 }
19
```