



Міністерство освіти і науки України
Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

Лабораторна робота №4

Технологія розробки програмного забезпечення

Тема роботи: «ШАБЛОНИ «SINGLETON», «ITERATOR», «PROXY»,
«STATE», «STRATEGY»»

Виконала
студентка групи ІА-12:
Яковенко Дар'я

Перевірив:
вик. Колеснік В. М.

Київ 2023

Тема: ШАБЛОНИ «SINGLETON», «ITERATOR», «PROXY», «STATE», «STRATEGY»

Мета: Ознайомитися з шаблонами, реалізувати не менше 3-х класів відповідно до обраної теми, реалізувати один з розглянутих шаблонів за обраною темою.

Завдання:

28 JSON Tool (ENG) (strategy, command, observer, template method, flyweight)
Display JSON schema with syntax highlight. Validate JSON schema and display errors. Create user friendly table\list box\other for read and update JSON schema properties metadata (description, example, data type, format, etc.). Auto save\restore when edit, maybe history. Can check JSON value by schema (Put schema and JSON = valid\invalid, display errors). Export schema as markdown table. JSON to "flat" view.

Хід роботи

Оскільки з переліку шаблонів у моєму завданні зустрічається лише шаблон «STRATEGY», застосуємо його для певного класу. Оскільки цей шаблон не залежить від стану об'єкту, як наприклад шаблон «STATE», і головне – це досягти результату (яким шляхом – вирішує користувач). Застосуємо даний шаблон для експорту json.

Оскільки в майбутньому ми можемо розширювати функціональність і створювати нові види, в які будемо експортувати json – тому даний шаблон дуже спростить написання коду.

```

AppMenu.java x
74 private void createNew() {
75
76 }
77
78 private void showMetadata() {
79     MetadataEditor metadataEditor = new MetadataEditor(parent.getFrame());
80     metadataEditor.setVisible(true);
81 }
82
83 private void showHistory() {
84 }
85
86 private void chooseExportOption(String exportType) {
87     String json = parent.getAppBody().getJsonText();
88
89     if (json.isEmpty()) {
90         MessageDisplay.showError("JSON редактор порожній. Введіть JSON перед експортом.");
91         return;
92     }
93
94     ExportStrategy strategy = switch (exportType) {
95         case "table" -> new TableExport();
96         case "text" -> new TextExport();
97         default -> null;
98     };
99
100     if (strategy != null) {
101         strategy.exportJson(json);
102     }
103 }
104 }

```

```

ExportStrategy.java x
1 package export;
2
3 public interface ExportStrategy {
4     void exportJson(String json);
5 }

```

```

TableExport.java ×
4  import org.json.JSONObject;
5  import validation.MessageDisplay;
6  import javax.swing.*;
7  import java.io.File;
8  import java.io.IOException;
9  import java.util.Iterator;
10
11  public class TableExport implements ExportStrategy {
12  public void exportJson(String json) {
13      JFileChooser fileChooser = new JFileChooser();
14      int returnValue = fileChooser.showSaveDialog( parent: null);
15
16      if (returnValue == JFileChooser.APPROVE_OPTION) {
17          try {
18              String filePath = fileChooser.getSelectedFile().getAbsolutePath() + ".csv";
19              FileWriter writer = new FileWriter(filePath);
20
21              JSONObject jsonObject = new JSONObject(json);
22              Iterator<String> keys = jsonObject.keys();
23
24              while (keys.hasNext()) {
25                  String key = keys.next();
26                  writer.append(key).append("\t").append(jsonObject.get(key).toString()).append("\n\t");
27                  // if object have some new fields
28                  writer.append(key).append("\t").append(jsonObject.get(key).toString()).append("\n");
29              }
30              writer.close();
31              MessageDisplay.showInfo("Table exported successfully.");
32          } catch (IOException e) {
33              MessageDisplay.showError("Error exporting table: " + e.getMessage());
34          } catch (JSONException e) {
35              MessageDisplay.showError("Error exporting table: " + new RuntimeException(e));
36          }
37      }
38  }
39  }

```

```

1  package export;
2
3  import validation.MessageDisplay;
4  import javax.swing.*;
5  import java.io.PrintWriter;
6  import java.io.IOException;
7
8  public class TextExport implements ExportStrategy {
9      public void exportJson(String json) {
10         JFileChooser fileChooser = new JFileChooser();
11         int returnValue = fileChooser.showSaveDialog( parent: null);
12
13         if (returnValue == JFileChooser.APPROVE_OPTION) {
14             try {
15                 String filePath = fileChooser.getSelectedFile().getAbsolutePath() + ".txt";
16                 PrintWriter writer = new PrintWriter(filePath);
17
18                 json = json.replaceAll( regex: "\\s", replacement: "");
19
20                 writer.write(json);
21                 writer.close();
22                 MessageDisplay.showInfo("Text exported successfully.");
23             } catch (IOException e) {
24                 MessageDisplay.showError("Error exporting text: " + e.getMessage());
25             }
26         }
27     }
28 }

```

Висновки:

У даній лабораторній роботі я ознайомила з шаблонами, реалізувала класи відповідно до обраної теми, а також застосувала шаблон «STRATEGY» для експортування json.