|  |
| --- |
|  |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение высшего образования **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** |

Отчет по выполнению практической работы 12

**Тема. ПАТТЕРНЫ ПРОЕКТИРОВАНИЯ. ПОРОЖДАЮЩИЕ ПАТТЕРНЫ: АБСТРАКТНАЯ ФАБРИКА,**

**ФАБРИЧНЫЙ МЕТОД**

Дисциплина Программирование на языке Джава

Выполнил

|  |  |
| --- | --- |
| студент | Болотов Михаил |
|  | Фамилия И.О. |
| группа | ИКБО-06-19 |
|  | Номер группы |

Москва 2020

**Содержание**

[​ Теория 2](#__RefHeading___Toc4128_811265792)

[​ Задание 2](#__RefHeading___Toc4136_811265792)

[​ Код 3](#__RefHeading___Toc35859_811265792)

[​ IDocument.java 3](#__RefHeading___Toc68396_811265792)

[​ ICreateDocument.java 3](#__RefHeading___Toc68398_811265792)

[​ TextCreateDocument.java 3](#__RefHeading___Toc68400_811265792)

[​ Redactor.java 4](#__RefHeading___Toc68402_811265792)

[​ TextDocument.java 6](#__RefHeading___Toc68404_811265792)

[​ Скриншот 7](#__RefHeading___Toc3748_811265792)

[​ Заключение 7](#__RefHeading___Toc3750_811265792)

[​ Библиографический список 7](#__RefHeading___Toc3752_811265792)

# Теория

Понятие паттерна.

Паттерны (или шаблоны) проектирования описывают типичные способы решения часто встречающихся проблем при проектировании программ.

# Задание

Вводная. Компании нужно написать редактор текста, редактор изображений и редактор музыки. В трёх приложениях будет очень много общего: главное окно, панель инструментов, команды меню будут весьма схожими. Чтобы не писать повторяющуюся основу трижды, вам поручили разработать основу (каркас) приложения, которую можно использовать во всех трёх редакторах. На совещании в компании была принята следующая архитектура:

Исходные данные. Есть некий базовый интерфейс IDocument, представляющий документ неопределённого рода. От него впоследствии будут унаследованы конкретные документы: TextDocument, ImageDocument, MusicDocument и т.п. Интерфейс IDocument перечисляет общие свойства и операции для всех документов.

* При нажатии пунктов меню File -> New и File -> Open требуется создать новый экземпляр документа (конкретного подкласса). Однако каркас не должен быть привязан ни к какому конкретному виду документов.
* Нужно создать фабричный интерфейс ICreateDocument. Этот интерфейс содержит два абстрактных фабричных метода: CreateNew и CreateOpen, оба возвращают экземпляр IDocument
* Каркас оперирует одним экземпляром IDocument и одним экземпляром ICreateDocument. Какие конкретные классы будут подставлены сюда, определяется во время запуска приложения.

Требуется:

1. создать перечисленные классы. Создать каркас приложения — окно редактора с меню File. В меню File реализовать пункты New, Open, Save, Exit.
2. продемонстрировать работу каркаса на примере текстового редактора. Потребуется создать конкретный унаследованный класс TextDocument и фабрику для него — CreateTextDocument.

# Код

### IDocument.java

package dev.ky3he4ik.pr.pr12;  
  
import java.io.File;  
import java.io.IOException;  
  
public interface IDocument {  
 void save(File file) throws IOException;  
 void setData(String data);  
 String getData();  
}

### ICreateDocument.java

package dev.ky3he4ik.pr.pr12;  
  
import java.io.File;  
  
public interface ICreateDocument {  
 IDocument CreateOpen(File file);  
  
 IDocument CreateNew();  
}

### TextCreateDocument.java

package dev.ky3he4ik.pr.pr12;  
  
import java.io.File;  
import java.io.IOException;  
import java.nio.file.Files;  
import java.util.List;  
  
public class TextCreateDocument implements ICreateDocument {  
 @Override  
 public IDocument CreateOpen(File file) {  
 try {  
 List<String> lines = Files.readAllLines(file.toPath());  
 StringBuilder sb = new StringBuilder();  
 for (String line : lines)  
 sb.append(line).append('\n');  
  
 return new TextDocument(sb.toString());  
 } catch (IOException e) {  
 return null;  
 }  
 }  
  
 @Override  
 public IDocument CreateNew() {  
 return new TextDocument();  
 }  
}

### Redactor.java

package dev.ky3he4ik.pr.pr12;  
  
  
import javax.swing.\*;  
import java.awt.\*;  
import java.awt.event.WindowEvent;  
import java.io.File;  
import java.io.IOException;  
import java.nio.file.AccessDeniedException;  
  
public class Redactor extends JFrame {  
 private ICreateDocument createDocument;  
 private IDocument document;  
 private JTextArea textArea;  
 private JFileChooser fileChooser;  
  
 public Redactor(ICreateDocument concreteCreateDocument) {  
 createDocument = concreteCreateDocument;  
 fileChooser = new JFileChooser(".");  
  
 setSize(1280, 720);  
 setLocation(320, 180);  
  
 setLayout(new CardLayout());  
  
 textArea = new JTextArea();  
 textArea.setEditable(false);  
 textArea.setEnabled(true);  
 add(textArea);  
  
 Menu menu = new Menu("File");  
  
 MenuItem menuItem = new MenuItem("New");  
 menuItem.addActionListener(e -> {  
 document = createDocument.CreateNew();  
 textArea.setEditable(true);  
 textArea.setText("");  
 });  
 menu.add(menuItem);  
  
 menuItem = new MenuItem("Open");  
 menuItem.addActionListener(e -> {  
 int resultCode = fileChooser.showOpenDialog(this);  
 if (resultCode == JFileChooser.APPROVE\_OPTION) {  
 File file = fileChooser.getSelectedFile();  
 IDocument tDocument = createDocument.CreateOpen(file);  
 if (tDocument == null)  
 JOptionPane.showMessageDialog(null, "Can't open this file!");  
 else {  
 document = tDocument;  
 textArea.setEditable(true);  
 textArea.setText(document.getData());  
 }  
 }  
 });  
 menu.add(menuItem);  
  
 menuItem = new MenuItem("Save");  
 menuItem.addActionListener(e -> {  
 if (document == null) {  
 JOptionPane.showMessageDialog(null, "Nothing to save!");  
 return;  
 }  
 int resultCode = fileChooser.showSaveDialog(this);  
 if (resultCode == JFileChooser.APPROVE\_OPTION) {  
 document.setData(textArea.getText());  
 File file = fileChooser.getSelectedFile();  
 try {  
 document.save(file);  
 } catch (AccessDeniedException ex) {  
 ex.printStackTrace();  
 JOptionPane.showMessageDialog(null, "Can't save to " + file.getPath() + ": not enough rights");  
 } catch (IOException ex) {  
 ex.printStackTrace();  
 JOptionPane.showMessageDialog(null, "Can't save to " + file.getPath());  
 }  
 }  
 });  
 menu.add(menuItem);  
  
 menuItem = new MenuItem("Exit");  
 menuItem.addActionListener(e -> dispatchEvent(new WindowEvent(this, WindowEvent.WINDOW\_CLOSING)));  
 menu.add(menuItem);  
  
 MenuBar menuBar = new MenuBar();  
 menuBar.add(menu);  
  
 setMenuBar(menuBar);  
  
 setVisible(true);  
 setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  
 }  
  
 public static void main(String[] args) {  
 new Redactor(new TextCreateDocument());  
 }  
}

### TextDocument.java

package dev.ky3he4ik.pr.pr12;  
  
import java.io.File;  
import java.io.IOException;  
import java.nio.file.Files;  
import java.nio.file.Paths;  
  
public class TextDocument implements IDocument {  
 private String data;  
  
 public TextDocument(String data) {  
 this.data = data;  
 }  
  
 public TextDocument() {  
 }  
  
 @Override  
 public String getData() {  
 return data;  
 }  
  
 @Override  
 public void setData(String data) {  
 this.data = data;  
 }  
  
 @Override  
 public void save(File file) throws IOException {  
 Files.writeString(file.toPath(), data);  
 }  
}

# Скриншот



# Заключение

В данной практической работе я научился применять порождающие паттерны при разработке программ на Java

# Библиографический список

1. Зорина Н.В. Курс лекций по Объектно-ориентированному программированию на Java, МИРЭА, Москва, 2016
2. Программирование на языке Java: работа со строками и массивами. Методические указания. [Электронный ресурс] : Учебно-методические пособия — Электрон. дан. — СПб. : ПГУПС, 2015. — 24 с.
3. Кожомбердиева, Г.И. Программирование на языке Java: создание графического интерфейса пользователя: учеб. пособие. [Электронный ресурс] : Учебные пособия / Г.И. Кожомбердиева, М.И. Гарина. — Электрон. дан. — СПб.: ПГУПС, 2012. — 67 с.
4. Вишневская, Т.И. Технология программирования. Часть 1. [Электронный ресурс] / Т.И. Вишневская, Т.Н. Романова. — Электрон. дан. — М. : МГТУ им. Н.Э. Баумана, 2007. — 59 с.