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

**Институт информационных технологий (ИТ)**

**Кафедра инструментального и прикладного программного обеспечения (ИиППО)**

**Дисциплина «Программирование на языке Джава»**

**ОТЧЕТ**

**ПО ПРАКТИЧЕСКИМ ЗАНЯТИЯМ №21, №22**

Выполнил студент группы ИНБО-02-20 Чернов Я. А.

Принял Степанов П.В.

Практические работы выполнены «\_\_\_»\_\_\_\_\_\_\_2021г.

«\_\_\_\_\_\_\_\_\_\_\_\_\_\_» «\_\_\_»\_\_\_\_\_\_\_2021г.

Отметка о выполнении

**Москва 2021 г.**

**Практические занятие №21-22**

**Задание**

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:

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 -&gt; New и File -&gt; 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. В данной практической работе рекомендуется использовать следующие паттерны: Абстрактная фабрика и фабричный метод.

##Рекомендации

Создавать интерфейс в 3 задаче нет необходимости, но это и не запрещается.

**Код программы**

**Класc Main**

**package** ru.mirea.Practice21\_22;  
  
**import** java.util.Scanner;  
  
**public class** Main {  
 **public static void** main(String[] args) {  
 Frame frame = **new** Frame();  
 Scanner in = **new** Scanner(System.***in***);  
 String choice;  
 **do** {  
 frame.New();  
 frame.Open();  
 frame.Save();  
 frame.Exit();  
 System.***out***.println(**"Создать новый документ? (Да/Нет)"**);  
 choice = in.nextLine();  
 } **while**(choice.equals(**"Да"**));  
 }  
}

**Класc Frame**

**package** ru.mirea.Practice21\_22;  
  
**import** java.util.Scanner;  
  
**public class** Frame {  
 **private** IDocument **Document**;  
 **private** IDocumentCreate **DocumentCreate**;  
 **public void** New(){  
 Scanner in = **new** Scanner(System.***in***);  
 System.***out***.println(**"Text, Image, Music"**);  
 String nameDocument = in.nextLine();  
 **if** (nameDocument.equals(**"Text"**)){  
 **DocumentCreate** = **new** CreateTextDocument();  
 **Document** = **DocumentCreate**.CreateNew();  
 **Document**.New();  
 }  
 **if** (nameDocument.equals(**"Image"**)){  
 **DocumentCreate** = **new** CreateImageDocument();  
 **Document** = **DocumentCreate**.CreateNew();  
 **Document**.New();  
 }  
 **if** (nameDocument.equals(**"Music"**)){  
 **DocumentCreate** = **new** CreateMusicDocument();  
 **Document** = **DocumentCreate**.CreateNew();  
 **Document**.New();  
 }  
 }  
 **public void** Open(){  
 **Document**.Open();  
 }  
 **public void** Save(){  
 **Document**.Save();  
 }  
 **public void** Exit(){  
 **Document**.Exit();  
 }  
}

**Интерфейс IDocument**

**package** ru.mirea.Practice21\_22;  
  
**public interface** IDocument {  
 **void** New();  
 **void** Open();  
 **void** Save();  
 **void** Exit();  
}

**Класc TextDocument**

**package** ru.mirea.Practice21\_22;  
  
**public class** TextDocument **implements** IDocument {  
 **public void** New() {  
 System.***out***.println(**"Файл создан (Text)"**);  
 }  
 **public void** Open() {  
 System.***out***.println(**"Файл открыт (Text)"**);  
 }  
 **public void** Save() {  
 System.***out***.println(**"Файл сохранен (Text)"**);  
 }  
 **public void** Exit() {  
 System.***out***.println(**"Файл закрыт (Text)"**);  
 }  
}

**Класc** **ImageDocument**

**package** ru.mirea.Practice21\_22;  
  
**public class** ImageDocument **implements** IDocument {  
 **public void** New() {  
 System.***out***.println(**"Файл создан (Image)"**);  
 }  
 **public void** Open() {  
 System.***out***.println(**"Файл открыт (Image)"**);  
 }  
 **public void** Save() {  
 System.***out***.println(**"Файл сохранен (Image)"**);  
 }  
 **public void** Exit() {  
 System.***out***.println(**"Файл закрыт (Image)"**);  
 }  
}

**Класc** **MusicDocument**

**package** ru.mirea.Practice21\_22;  
  
**public class** MusicDocument **implements** IDocument {  
 **public void** New() {  
 System.***out***.println(**"Файл создан (Music)"**);  
 }  
 **public void** Open() {  
 System.***out***.println(**"Файл открыт (Music)"**);  
 }  
 **public void** Save() {  
 System.***out***.println(**"Файл сохранен (Music)"**);  
 }  
 **public void** Exit() {  
 System.***out***.println(**"Файл закрыт (Music)"**);  
 }  
}

**Интерфейс** **IDocumentCreate**

**package** ru.mirea.Practice21\_22;  
  
**public interface** IDocumentCreate {  
 IDocument CreateNew();  
 IDocument CreateOpen();  
}

**Класc** **CreateTextDocument**

**package** ru.mirea.Practice21\_22;  
  
**public class** CreateTextDocument **implements** IDocumentCreate{  
 **private** TextDocument **text**;  
 **public** IDocument CreateNew() {  
 **text** = **new** TextDocument();  
 **return text**;  
 }  
 **public** IDocument CreateOpen() {  
 **return text**;  
 }  
}

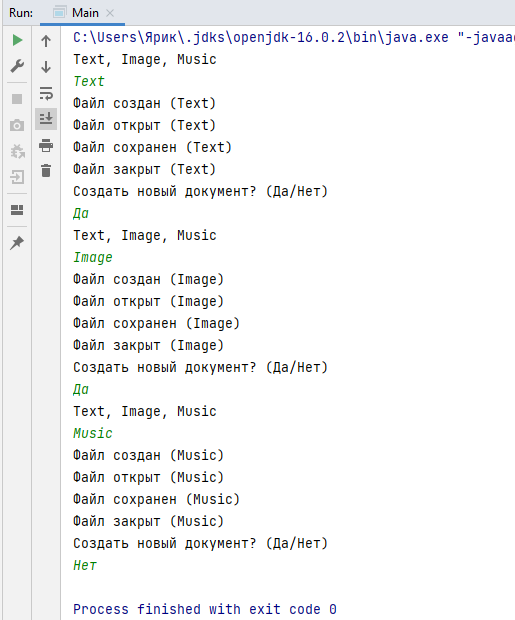
**Класc** **CreateImageDocument**

**package** ru.mirea.Practice21\_22;  
  
**public class** CreateImageDocument **implements** IDocumentCreate {  
 **private** ImageDocument **image**;  
 **public** IDocument CreateNew() {  
 **image** = **new** ImageDocument();  
 **return image**;  
 }  
 **public** IDocument CreateOpen() {  
 **return image**;  
 }  
}

**Класc CreateMusicDocument**

**package** ru.mirea.Practice21\_22;  
  
**public class** CreateMusicDocument **implements** IDocumentCreate {  
 **private** MusicDocument **music**;  
 **public** IDocument CreateNew() {  
 **music** = **new** MusicDocument();  
 **return music**;  
 }  
 **public** IDocument CreateOpen() {  
 **return music**;  
 }  
}

**Тестирование**

****

**Выводы**

По ходу выполнения данной практической работы мы изучили использование паттернов: абстрактная фабрика и фабричный метод.

Git: <https://github.com/YaroslavChernov0220/PracticeMIREA>