|  |
| --- |
|  |
| 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 ПРАКТИЧЕСКОМУ ЗАНЯТИЮ №10**

Выполнил студент группы ИНБО-02-20 Баринов.И.В.

Принял Степанов П.В.

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

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

Отметка о выполнении

**Москва – 2021 г.**

## **Задание**

Научиться применять порождающие паттерны при разработке программ на Java

## **Ход Работы**

В ходе выполнения работы были получены следующие исходные коды:

package Practise10;  
  
public class ConcreteFactory implements ComplexAbstractFactory {  
  
 public Complex createComplex() {  
 return new Complex(0, 0);  
 }  
  
 public Complex createComplex(int real, int image) {  
 return new Complex(real, image);  
 }  
}

----

package Practise10;  
  
import java.util.Scanner;  
public class Test {  
 private static final Scanner *IN* = new Scanner(System.*in*);  
 private static final ConcreteFactory *FACTORY* = new ConcreteFactory();  
  
 public static void main(String[] args) {  
 System.*out*.print("Введите вещественную часть: ");  
 int real = *IN*.nextInt();  
  
 System.*out*.print("Введите мнимую часть: ");  
 int image = *IN*.nextInt();  
  
 Complex ComplexNumber11 = *FACTORY*.createComplex();  
 Complex ComplexNumber12 = *FACTORY*.createComplex(real, image);  
  
 System.*out*.println("ComplexNumber11 = " + ComplexNumber11);  
 System.*out*.println("ComplexNumber12 = " + ComplexNumber12);  
 }  
}

----

package Practise10.Exercise2;  
  
public interface AbstractChairFactory {  
  
 VictorianChair createVictorianChair(int age);  
  
 MagicalChair createMagicalChair();  
  
 FunctionalChair createFunctionalChair();  
}

----

package Practise10.Exercise2;  
  
public interface Chair {  
 String toString();  
}

----

package Practise10.Exercise2;  
  
public class ChairFactory implements AbstractChairFactory {  
  
 public VictorianChair createVictorianChair(int age) {  
 return new VictorianChair(10);  
 }  
  
  
 public MagicalChair createMagicalChair() {  
 return new MagicalChair();  
 }  
  
  
 public FunctionalChair createFunctionalChair() {  
 return new FunctionalChair();  
 }  
}

----

package Practise10.Exercise2;  
  
import java.util.Scanner;  
  
public class Client {  
  
 public Chair chair;  
  
 public void sit() throws Exception {  
 var scan = new Scanner(System.*in*);  
 var fact = new ChairFactory();  
  
 System.*out*.println("Какой стул создать?");  
 System.*out*.println(" 1) Викторианский стул");  
 System.*out*.println(" 2) Магический стул");  
 System.*out*.println(" 3) Функциональный стул");  
  
 int k = scan.nextInt();  
  
 System.*out*.println();  
  
 switch (k) {  
 case 1:  
 System.*out*.print("Введите возраст стула: ");  
 chair = fact.createVictorianChair(scan.nextInt());  
 System.*out*.println(chair);  
 break;  
  
 case 2:  
 chair = fact.createMagicalChair();  
 ((MagicalChair) chair).doMagic();  
 break;  
  
 case 3:  
 chair = fact.createFunctionalChair();  
 ((FunctionalChair) chair).doFunction();  
 break;  
 }  
  
 }  
}

----

package Practise10.Exercise2;  
  
public class FunctionalChair implements Chair {  
  
 public void doFunction() throws Exception {  
 System.*out*.println("Функциональный стул очень удобен\*");  
 throw new Exception("[ERROR]: Функциональный стул сломался");  
 }  
  
 @Override  
 public String toString() {  
 return "FunctionalChair очень удобен";  
 }  
}

----

package Practise10.Exercise2;  
  
public class MagicalChair implements Chair {  
  
 public void doMagic(){  
 System.*out*.println("Магия: \*происходит\*");  
 }  
  
 @Override  
 public String toString() {  
 return """  
 MagicalChair {  
 Вы телепортированы в подземелье  
 }""";  
 }  
}

----

package Practise10.Exercise2;  
  
public class Test {  
 public static void main(String[] args) {  
 Client cl = new Client();  
 try {  
 cl.sit();  
 } catch (Exception exc) {  
 System.*out*.println(exc.getMessage());  
 }  
 }  
}

----

package Practise10.Exercise2;  
  
public class VictorianChair implements Chair {  
 private final int age;  
  
 public VictorianChair(int age) {  
 this.age = age;  
 }  
  
 public int getAge() {  
 return this.age;  
 }  
  
 @Override  
 public String toString() {  
 return (  
 "VictorianChair {\n"  
 + " age: " + this.age + "\n"  
 + "}"  
 );  
 }  
}

## **Вывод**

Получены знания по работе AbstractFactory

GitHub:

https://github.com/BarinovIvan/RuPractises/tree/master/src/Practise10