

**Министерство образования Республики Беларусь
Учреждение Образования
«Брестский Государственный Технический Университет»
Кафедра ИИТ**

**Лабораторная работа №4
По дисциплине СПП за 5 семестр
Тема: «Java»**

Выполнил:
Студент 3-го курса
Группы ПО-5
Крощук В.В.
Проверил:
Крощенко А.А.

Брест 2021

Лабораторная работа №4

Цель работы:

приобрести практические навыки в области объектно-ориентированного проектирования

Вариант: 8

Задания и выполненные решения:

- 1. Создать класс CD (тр3-диск) с внутренним классом, с помощью объектов которого можно хранить информацию о каталогах, подкаталогах и записях.*

Код:

Main.java

```
package com.company;
public class Main {

    public static void main(String[] args) {
        CD cd = new CD();

        cd.addCatalog("first").setCatalog("first_1").setCatalog("first_2").setCatalog("first_3");

        cd.addCatalog("second").setCatalog("second_1").setCatalog("second_2");

        cd.addCatalog("third").setCatalog("third_1").setCatalog("third_2").setCatalog("third_3").setCatalog("third_4");
        cd.print();
    }
}
```

CD.java

```
package com.company;
import java.util.ArrayList;
import java.util.List;
import java.util.Objects;

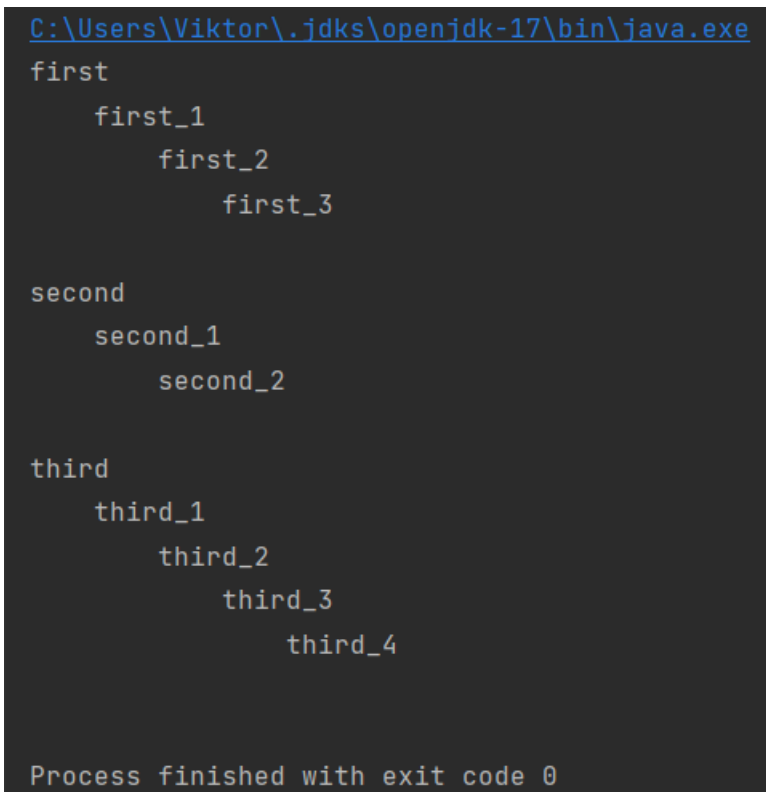
public class CD {
    private List<Catalog> catalogList;
    public CD() {
        catalogList = new ArrayList<>();
    }
    public Catalog addCatalog(String name) {
        Catalog catalog = new Catalog(name);
        catalog.setSpace("\t");
        catalogList.add(catalog);
        return catalog;
    }
}
```

```

    }
    public void print() {
        catalogList.forEach(System.out::println);
    }
    class Catalog {
        private String name;
        private Catalog catalog;
        private String space;
        public Catalog(String name) {
            this.name = name;
        }
        public void setSpace(String space) {
            this.space = space;
        }
        public Catalog setCatalog(String name) {
            Catalog catalog = new Catalog(name);
            catalog.setSpace(this.space+"\t");
            this.catalog = catalog;
            return catalog;
        }
        @Override
        public String toString() {
            return name + "\n" + space + (Objects.nonNull(catalog) ? catalog
: "");
        }
    }
}

```

Результат:



```

C:\Users\Viktor\.jdk\openjdk-17\bin\java.exe
first
    first_1
        first_2
            first_3

second
    second_1
        second_2

third
    third_1
        third_2
            third_3
                third_4

Process finished with exit code 0

```

Выполнение происходит согласно поставленной задачи. Работает корректно!

2. **Реализовать агрегирование. При создании класса агрегируемый класс объявляется как атрибут (локальная переменная, параметр метода). Включить в каждый класс 2-3 метода на выбор. Продемонстрировать использование разработанных классов.**

Создать класс Текст, используя класс Абзац.

Код:

Main.java

```
package com.company;

public class Main {

    public static void main(String[] args) {
        Paragraph paragraph1 = new Paragraph();
        paragraph1.addString("First string in first paragraph.");
        paragraph1.addString("Second string in first \nparagraph.");
        paragraph1.addString("Third string in first paragraph.");
        Paragraph paragraph2 = new Paragraph();
        paragraph2.addString("First string in second paragraph.");
        paragraph2.addString("Second string in second \nparagraph.");
        paragraph2.addString("Third string in second paragraph.");
        Paragraph paragraph3 = new Paragraph();
        paragraph3.addString("First string in third paragraph.");
        paragraph3.addString("Second string in third \nparagraph.");
        paragraph3.addString("Third string in third paragraph.");
        paragraph3.deleteString(2);
        Text text = new Text();
        text.addParagraph(paragraph1);
        text.addParagraph(paragraph2);
        text.addParagraph(paragraph3);
        text.printText();
        text.deleteParagraph(1);
        System.out.println("After removing second paragraph:");
        text.printText();
    }
}
```

Paragraph.java

```
package com.company;

import java.util.ArrayList;
import java.util.List;

public class Paragraph {
    private List<String> listStrings;
    public Paragraph() {
```

```

        this.listStrings = new ArrayList<>();
    }
    public List<String> getListStrings() {
        return listStrings;
    }
    public void addString(String string) {
        listStrings.add(string);
    }
    public void deleteString(int stringNumber) {
        listStrings.remove(stringNumber);
    }
    public void printParagraph() {
        listStrings.forEach(string -> System.out.print(string.concat(" ")));
    }
}

```

Text.java

```

package com.company;
import java.util.ArrayList;
import java.util.List;

public class Text {
    private List<Paragraph> listParagraph;
    public Text() {
        listParagraph = new ArrayList<>();
    }
    public List<Paragraph> getListParagraph() {
        return listParagraph;
    }
    public void addParagraph(Paragraph paragraph) {
        listParagraph.add(paragraph);
    }
    public void deleteParagraph(int paragraphId) {
        listParagraph.remove(paragraphId);
    }
    public void printText() {
        listParagraph.forEach(paragraph -> {paragraph.printParagraph();
            System.out.println("\n");});
    }
}

```

Результат:

```
C:\Users\Viktor\.jdk\openjdk-17\bin\java.exe "-javaagent:C:\Users\Viktor\.jdk\openjdk-17\bin\javaagent.jar"
First string in first paragraph. Second string in first paragraph. Third string in first paragraph.

First string in second paragraph. Second string in second paragraph. Third string in second paragraph.

First string in third paragraph. Second string in third paragraph.

After removing second paragraph:
First string in first paragraph. Second string in first paragraph. Third string in first paragraph.

First string in third paragraph. Second string in third paragraph.

Process finished with exit code 0
```

Исходя из полученных результатов, все работает корректно, согласно поставленному заданию.

3. *Построить модель программной системы с применением отношений (обобщения, агрегации, ассоциации, реализации) между классами. Задать атрибуты и методы классов. Реализовать (если необходимо) дополнительные классы. Продемонстрировать работу разработанной системы.*

Система **Интернет-магазин**. **Администратор** добавляет информацию о **Товаре**. **Клиент** делает и оплачивает **Заказ на Товары**. **Администратор** регистрирует **Продажу** и может занести неплательщиков в «черный список».

Код:

Main.java

```
package com.company;

public class Main {

    public static void main(String[] args) {
        OnlineStore onlineStore = new OnlineStore();
        Administrator administrator = new Administrator(onlineStore);
        administrator.addProduct("Диван");
        administrator.addProduct("Ноутбук");
        administrator.addProduct("Стул");
        administrator.addProduct("Лампа");
    }
}
```

```

        administrator.addProduct("Телефон");
        administrator.addProduct("Стол");
        Client client1 = new Client(1, onlineStore);
        System.out.println("Все товары: ");
        client1.printProducts();
        client1.addOrder(2).pay();
        client1.addOrder(3);
        client1.addOrder(1).pay();
        System.out.println("\nЗаказы 1 клиента: ");
        client1.printOrders();
        Client client2 = new Client(2, onlineStore);
        client2.addOrder(5).pay();
        client2.addOrder(1).pay();
        System.out.println("\nЗаказы 2 клиента: ");
        client2.printOrders();
        administrator.addToBlackList();
        System.out.println("\nПользователь пытается добавить товар: ");
        client1.addOrder(5).pay();
        client2.addOrder(6).pay();
        System.out.println("\nЗаказы 1 клиента: ");
        client1.printOrders();
        System.out.println("\nЗаказы 2 клиента: ");
        client2.printOrders();
    }
}

```

Administrator.java

```

package com.company;
public class Administrator {
    private OnlineStore onlineStore;
    public Administrator(OnlineStore onlineStore) {
        this.onlineStore = onlineStore;
    }
    public void addProduct(String productName) {
        onlineStore.addProduct(productName);
    }
    public void addToBlackList() {
        onlineStore.addToBlackList();
    }
}

```

Client.java

```

package com.company;
public class Client {
    private int clientId;
    private OnlineStore onlineStore;
    public Client(int clientId, OnlineStore onlineStore) {
        this.clientId = clientId;
        this.onlineStore = onlineStore;
    }
    public void printProducts() {

```

```

        onlineStore.printProducts();
    }
    public void printOrders() {
        onlineStore.printUserOrders(clientId);
    }
    public Order addOrder(int productId) {
        return onlineStore.addOrder(clientId, productId);
    }
}

```

OnlineStore.java

```

package com.company;
import java.util.ArrayList;
import java.util.List;
import java.util.stream.Collectors;

public class OnlineStore {
    private List<Product> productList;
    private List<Order> orderList;
    private List<Integer> blackList;
    public OnlineStore() {
        this.productList = new ArrayList<>();
        this.orderList = new ArrayList<>();
        this.blackList = new ArrayList<>();
    }
    private boolean isInBlackList(int userId) {
        return blackList.contains(userId);
    }
    public void printProducts() {
        productList.forEach(System.out::println);
    }
    public void printUserOrders(int userId) {
        orderList.stream().filter(order -> order.getUserId() ==
            userId).forEach(System.out::println);
    }
    public void addToBlackList() {
        blackList = orderList.stream().filter(order ->
!order.isPaid()).map(Order::getUserId).collect(Collectors.toList());
    }
    public void addProduct(String productName) {
        productList.add(new Product(productList.size() + 1, productName));
    }
    public Order addOrder(int userId, int productId) {
        Order order = new Order(userId, productList.get(--productId));
        if(isInBlackList(userId)) {
            System.out.println("Простите но вы в черном списке");
        } else{
            orderList.add(order);
        }
        return order;
    }
}

```


Order.java

```
package com.company;
public class Order {
    private int userId;
    private Product product;
    private boolean isPaid;

    public Order(int userId, Product product) {
        this.userId = userId;
        this.product = product;
        this.isPaid = false;
    }

    public void pay() {
        this.isPaid = true;
    }

    public int getUserId() {
        return userId;
    }

    public boolean isPaid() {
        return isPaid;
    }

    @Override
    public String toString() {
        return product +
            ", Куплено=" + isPaid;
    }
}
```

Product.java

```
package com.company;
public class Product {
    private int productId;
    private String productName;
    public Product() {
    }
    public Product(int productId, String productName) {
        this.productId = productId;
        this.productName = productName;
    }
    public int getProductId() {
        return productId;
    }
    public void setProductId(int productId) {
        this.productId = productId;
    }
    public String getProductName() {
        return productName;
    }
}
```

```

    public void setProductName(String productName) {
        this.productName = productName;
    }
    @Override
    public String toString() {
        return "Товар №" + productId +
            ": '" + productName + '\'';
    }
}

```

Результат:

```

C:\Users\Viktor\.jdk\openjdk-17\bin\java
Все товары:
Товар №1: 'Диван'
Товар №2: 'Ноутбук'
Товар №3: 'Стул'
Товар №4: 'Лампа'
Товар №5: 'Телефон'
Товар №6: 'Стол'

Заказы 1 клиента:
Товар №2: 'Ноутбук', Куплено=true
Товар №3: 'Стул', Куплено=false
Товар №1: 'Диван', Куплено=true

Заказы 2 клиента:
Товар №5: 'Телефон', Куплено=true
Товар №1: 'Диван', Куплено=true

Пользователь пытается добавить товар:
Простите но вы в черном списке

Заказы 1 клиента:
Товар №2: 'Ноутбук', Куплено=true
Товар №3: 'Стул', Куплено=false
Товар №1: 'Диван', Куплено=true

Заказы 2 клиента:
Товар №5: 'Телефон', Куплено=true
Товар №1: 'Диван', Куплено=true
Товар №6: 'Стол', Куплено=true

Process finished with exit code 0
|

```

Исходя из полученных результатов, все работает корректно, согласно поставленному заданию.

Вывод: приобрел практические навыки в области объектно-ориентированного проектирования.