

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ  
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ  
БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
КАФЕДРА ИИТ

**Лабораторная работа №4**

По дисциплине: «Современные платформы программирования»

**Выполнил:**

Студент 3 курса

группы ПО-8:

Макаревич Е.С.

**Проверил:**

Крощенко А.А.

**Цель работы:** приобрести практические навыки в области объектно-ориентированного проектирования.

### Вариант 17

**Задание 1.** Создать класс CD (mp3-диск) с внутренним классом, с помощью объектов которого можно хранить информацию о каталогах, подкаталогах и записях.

**Код программы:**

**CD.java**

```
package task01;

import java.util.ArrayList;

public class CD {
    private ArrayList<Directory> directories;

    public CD() {
        directories = new ArrayList<>();
    }

    public void addDirectory(String name) {
        directories.add(new Directory(name));
    }

    public void addRecord(String directoryName, String recordName) {
        for (Directory directory : directories) {
            if (directory.getName().equals(directoryName)) {
                directory.addRecord(recordName);
                return;
            }
        }
        System.out.println("Directory not found: " + directoryName);
    }

    public void showContents() {
        for (Directory directory : directories) {
            System.out.println("Directory: " + directory.getName());
            directory.showRecords();
        }
    }

    private class Directory {
        private String name;
        private ArrayList<String> records;

        public Directory(String name) {
            this.name = name;
            records = new ArrayList<>();
        }

        public String getName() {
            return name;
        }

        public void addRecord(String recordName) {
            records.add(recordName);
        }
    }
}
```

```

    }

    public void showRecords() {
        for (String record : records) {
            System.out.println("    Record: " + record);
        }
    }
}

public static void main(String[] args) {
    CD cd = new CD();
    cd.addDirectory("Rock");
    cd.addRecord("Rock", "Stairway to Heaven");
    cd.addRecord("Rock", "Bohemian Rhapsody");

    cd.addDirectory("Pop");
    cd.addRecord("Pop", "Shape of You");
    cd.addRecord("Pop", "Billie Jean");

    cd.showContents();
}
}

```

### Результат программы:

```

Directory: Rock
    Record: Stairway to Heaven
    Record: Bohemian Rhapsody
Directory: Pop
    Record: Shape of You
    Record: Billie Jean

```

**Задание 2. Создать класс Абзац, используя класс Слово.**

### Код программы:

#### Paragraph.java

```

package task02;

import java.util.ArrayList;

public class Paragraph {
    private ArrayList<Word> words;

    public Paragraph() {
        words = new ArrayList<>();
    }

    public void addWord(Word word) {
        words.add(word);
    }

    public int wordCount() {
        return words.size();
    }

    public int length() {
        int totalLength = 0;

```

```

        for (Word word : words) {
            totalLength += word.length();
        }
        return totalLength;
    }
    public ArrayList<Word> getWords() {
        return words;
    }
}

```

## Word.java

```

package task02;

public class Word {
    private String word;

    public Word(String word) {
        this.word = word;
    }

    public String getWord() {
        return word;
    }

    public int length() {
        return word.length();
    }

    public boolean isPalindrome() {
        String reversedWord = new StringBuilder(word).reverse().toString();
        return word.equalsIgnoreCase(reversedWord);
    }
}

```

## Main.java

```

package task02;

public class Main {
    public static void main(String[] args) {
        Word word1 = new Word("Hello");
        Word word2 = new Word("World");

        Paragraph paragraph = new Paragraph();

        paragraph.addWord(word1);
        paragraph.addWord(word2);

        System.out.println("Added words:");
        for (Word word : paragraph.getWords()) {
            System.out.println(word.getWord());
        }

        System.out.println("The number of words in a paragraph: " +
paragraph.wordCount());
        System.out.println("Paragraph length: " + paragraph.length());
    }
}

```

## Результат программы:

```
Added words:
Hello
World
The number of words in a paragraph: 2
Paragraph length: 10
```

**Задание 3. Система Телефонная станция. Абонент оплачивает Счет за разговоры и Услуги, может попросить Администратора сменить номер и отказаться от услуг. Администратор изменяет номер, Услуги и временно отключает Абонента за неуплату.**

## Код программы:

### Administrator.java

```
package task03;

public class Administrator {
    private String name;

    public Administrator(String name) {
        this.name = name;
    }

    public void changeSubscriberNumber(Subscriber subscriber, String newNumber)
    {
        subscriber.changePhoneNumber(newNumber);
        System.out.println("Номер абонента успешно изменен на " + newNumber);
    }

    public void temporarilyDisableSubscriber(Subscriber subscriber) {
        // Логика временного отключения абонента за неуплату
        System.out.println("Абонент " + subscriber.getPhoneNumber() + " временно отключен за неуплату");
    }

    public void manageServices(Subscriber subscriber, Service service, boolean add) {
        if (add) {
            subscriber.addService(service);
            System.out.println("Абоненту " + subscriber.getPhoneNumber() + " успешно подключена услуга: " + service.getName());
        } else {
            subscriber.removeService(service);
            System.out.println("Абонент отказался от услуги: " + service.getName());
        }
    }

    @Override
    public String toString() {
        return "Администратор " + name;
    }
}
```

## Bill.java

```
package task03;

public class Bill {
    private double amount;

    public Bill(double amount) {
        this.amount = amount;
    }

    public double getAmount() {
        return amount;
    }

    public void payBill(double amount) {
        // Логика оплаты счета
        System.out.println("Счет в размере " + amount + " оплачен");
    }
}
```

## Main.java

```
package task03;

public class Main {
    public static void main(String[] args) {
        TelephoneStation station = new TelephoneStation();

        Subscriber subscriber1 = new Subscriber("123456789");
        Subscriber subscriber2 = new Subscriber("987654321");

        Service service1 = new Service("Internet", 25.0);
        Service service2 = new Service("Call Forwarding", 10.0);

        Administrator admin = new Administrator("John Doe");

        station.addSubscriber(subscriber1);
        station.addSubscriber(subscriber2);
        station.addService(service1);
        station.addService(service2);
        station.addAdministrator(admin);

        station.manageServices(admin, subscriber1, service1, true);
        station.manageServices(admin, subscriber2, service2, true);

        station.changeSubscriberNumber(admin, subscriber1, "987654321");

        station.temporarilyDisableSubscriber(admin, subscriber1);

        System.out.println("Телефонная станция:");
        System.out.println("Список абонентов: " + station.getSubscribers());
        System.out.println("Список услуг: " + station.getServices());
        System.out.println("Список администраторов: " +
station.getAdministrators());
        System.out.println();
    }
}
```

## Service.java

```

package task03;

public class Service {
    private String name;
    private double cost;

    public Service(String name, double cost) {
        this.name = name;
        this.cost = cost;
    }

    public String getName() {
        return name;
    }

    public double getCost() {
        return cost;
    }

    @Override
    public String toString() {
        return name + " (Cost: $" + cost + ")";
    }
}

```

## Subscriber.java

```

package task03;

public class Subscriber extends User {
    public Subscriber(String phoneNumber) {
        super(phoneNumber);
    }

    public void makeCall(String number) {
        System.out.println("Вызываем номер " + number);
    }

    public void requestServiceCancellation(Service service) {
        removeService(service);
        System.out.println("Абонент успешно отказался от услуги: " +
service.getName());
    }

    @Override
    public String toString() {
        return "Абонент с номером " + getPhoneNumber();
    }
}

```

## TelephoneStation.java

```

package task03;

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

public class TelephoneStation {
    private List<Subscriber> subscribers;
    private List<Service> services;
    private List<Administrator> administrators;
}

```

```

private List<Bill> bills;

public TelephoneStation() {
    subscribers = new ArrayList<>();
    services = new ArrayList<>();
    administrators = new ArrayList<>();
    bills = new ArrayList<>();
}

public void addSubscriber(Subscriber subscriber) {
    subscribers.add(subscriber);
}

public void removeSubscriber(Subscriber subscriber) {
    subscribers.remove(subscriber);
}

public void addService(Service service) {
    services.add(service);
}

public void addAdministrator(Administrator administrator) {
    administrators.add(administrator);
}

public List<Subscriber> getSubscribers() {
    return subscribers;
}

public List<Service> getServices() {
    return services;
}

public List<Administrator> getAdministrators() {
    return administrators;
}

    public void changeSubscriberNumber(Administrator admin, Subscriber
subscriber, String newNumber) {
        if (administrators.contains(admin) &&
subscribers.contains(subscriber)) {
            admin.changeSubscriberNumber(subscriber, newNumber);
        } else {
            System.out.println("Ошибка: Администратор или абонент не
найжены");
        }
    }

    public void temporarilyDisableSubscriber(Administrator admin, Subscriber
subscriber) {
        if (administrators.contains(admin) &&
subscribers.contains(subscriber)) {
            admin.temporarilyDisableSubscriber(subscriber);
        } else {
            System.out.println("Ошибка: Администратор или абонент не
найжены");
        }
    }

    public void manageServices(Administrator admin, Subscriber subscriber,
Service service, boolean add) {
        if (administrators.contains(admin) &&
subscribers.contains(subscriber)) {
            admin.manageServices(subscriber, service, add);

```



```
        } else {  
            System.out.println("Ошибка: Администратор или абонент не  
найжены");  
        }  
    }  
}
```

## User.java

```
package task03;  
  
import java.util.ArrayList;  
import java.util.List;  
  
public abstract class User {  
    protected String phoneNumber;  
    protected List<Service> services;  
  
    public User(String phoneNumber) {  
        this.phoneNumber = phoneNumber;  
        this.services = new ArrayList<>();  
    }  
  
    public String getPhoneNumber() {  
        return phoneNumber;  
    }  
  
    public void changePhoneNumber(String newNumber) {  
        this.phoneNumber = newNumber;  
    }  
  
    public void addService(Service service) {  
        services.add(service);  
    }  
  
    public void removeService(Service service) {  
        services.remove(service);  
    }  
  
    public List<Service> getServices() {  
        return services;  
    }  
}
```

## Результат программы:

```
Абоненту 123456789 успешно подключена услуга: Internet  
Абоненту 987654321 успешно подключена услуга: Call Forwarding  
Номер абонента успешно изменен на 987654321  
Абонент 987654321 временно отключен за неуплату  
Телефонная станция:  
Список абонентов: [Абонент с номером 987654321, Абонент с номером 987654321]  
Список услуг: [Internet (Cost: $25.0), Call Forwarding (Cost: $10.0)]  
Список администраторов: [Администратор John Doe]
```

**Вывод:** Приобрели практические навыки в области объектно-ориентированного проектирования.