МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
“БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ”
КАФЕДРА ИНТЕЛЛЕКТУАЛЬНЫХ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

Отчёт
по лабораторной работе №4

Выполнил:
студент группы ПО-9
Зеленков К. И.

Проверил:
Крощенко А. А.

Брест 2024

Вариант 6

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

Задание 1

Реализовать указанный класс, включив в него вспомогательный внутренний класс или классы.

Реализовать 2-3 метода (на выбор). Продемонстрировать использование реализованных классов.

6) Создать класс Catalog (каталог) с внутренним классом, с помощью объектов которого можно

хранить информацию об истории выдач книги читателям.

Код программы:

Main1.java:

```
import java.util.ArrayList;
import java.util.Date;
import java.util.List;

class Catalog {
    private List<Book> books;

    public Catalog() {
        books = new ArrayList<>();
    }

    public void addBook(Book book) {
        books.add(book);
    }

    public void displayBooks() {
        for (Book book : books) {
            System.out.println(book);
        }
    }

    public class IssueHistory {
        private Book book;
        private List<String> readers;
        private List<Date> dates;

        public IssueHistory(Book book) {
            this.book = book;
            readers = new ArrayList<>();
            dates = new ArrayList<>();
        }

        public void issueBook(String reader, Date date) {
            readers.add(reader);
            dates.add(date);
        }

        public void displayIssueHistory() {
            for (int i = 0; i < readers.size(); i++) {
                System.out.println("Читатель: " + readers.get(i) + ", Дата:
" + dates.get(i));
```

```
            }
        }
    }
}

class Book {
    private String title;
    private String author;
    private int pageCount;

    public Book(String title, String author, int pageCount) {
        this.title = title;
        this.author = author;
        this.pageCount = pageCount;
    }

    @Override
    public String toString() {
        return "Заголовок: " + title + ", Автор: " + author + ", Количество
страниц: " + pageCount;
    }
}

public class Main1 {
    public static void main(String[] args) {
        Catalog catalog = new Catalog();

        Book book1 = new Book("1", "Зеленков", 248);
        Book book2 = new Book("2", "Марзан", 192);

        catalog.addBook(book1);
        catalog.addBook(book2);

        Catalog.IssueHistory issueHistory1 = catalog.new
IssueHistory(book1);
        issueHistory1.issueBook("Дмитрий Кухарев", new Date());
        issueHistory1.issueBook("Никита Зейденс", new Date());

        Catalog.IssueHistory issueHistory2 = catalog.new
IssueHistory(book2);
        issueHistory2.issueBook("Ярослав Кучко", new Date());

        System.out.println("Книги:");
        catalog.displayBooks();

        System.out.println("\nИстория книги 1:");
        issueHistory1.displayIssueHistory();

        System.out.println("\nИстория книги 2:");
        issueHistory2.displayIssueHistory();
    }
}
```

Результат работы программы:

```
"C:\Program Files\Eclipse Adoptium\jdk-17.0.1.12-hotspot\bin\java.ex
Книги:
Заголовок: 1, Автор: Зеленков, Количество страниц: 248
Заголовок: 2, Автор: Марзан, Количество страниц: 192

История книги 1:
Читатель: Дмитрий Кухарев, Дата: Tue Apr 16 22:28:00 MSK 2024
Читатель: Никита Зейденс, Дата: Tue Apr 16 22:28:00 MSK 2024

История книги 2:
Читатель: Ярослав Кучко, Дата: Tue Apr 16 22:28:00 MSK 2024
```

Задание 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.

6) Создать класс Страница, используя класс Слово.

Код программы

Main2.java:

```
import java.util.ArrayList;

class Word {
    private String word;

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

    public String getWord() {
        return word;
    }

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

class Page {
    private ArrayList<Word> words;

    public Page(ArrayList<Word> words) {
        this.words = words;
    }

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

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

    public void displayWords() {
        System.out.println("Слова на странице:");
        for (Word word : words) {
            System.out.print(word.getWord() + " ");
        }
        System.out.println();
    }
    public int totalLength() {
        int totalLengthOfWords = 0;
        for (Word word : words) {
            totalLengthOfWords += word.length();
        }
        return totalLengthOfWords;
    }
}

public class Main2 {
    public static void main(String[] args) {
        Word word1 = new Word("Константин");
        Word word2 = new Word("Акимов");
```

```
        ArrayList<Word> pageWords = new ArrayList<>();
        pageWords.add(word1);
        pageWords.add(word2);
        Page page = new Page(pageWords);

        Word word3 = new Word("Игоревич");
        page.addWord(word3);

        page.displayWords();

        System.out.println("Общее количество слов на странице: " +
page.wordCount());
        System.out.println("Общая длина слов на странице: " +
page.totalLength());
    }
}
```

Результат работы программы:

```
Слова на странице:
Константин Акимов Игоревич
Общее количество слов на странице: 3
Общая длина слов на странице: 24
```

Задание 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.

6) 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

Main3.java:

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

class Arrears {
    private double amount;

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

    public double getAmount() {
        return amount;
    }

    public void setAmount(double amount) {
        this.amount = amount;
    }
}
```

```
class Service {
    private String name;
    private double price;

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

    public String getName() {
        return name;
    }

    public double getPrice() {
        return price;
    }
}

class Subscriber {
    public String phoneNumber;
    public List<Service> services;
    public Arrears arrears;
    public boolean isActive;

    public Subscriber(String phoneNumber) {
        this.phoneNumber = phoneNumber;
        this.services = new ArrayList<>();
        this.arrears = new Arrears(0);
        this.isActive = true;
    }

    public void requestPhoneNumberChange(Administrator administrator, String
newNumber) {
        administrator.changePhoneNumber(this, newNumber);
    }

    public void requestService(Administrator administrator, Service service)
{
        administrator.requestService(this, service);
    }

    public void cancelService(Administrator administrator, Service service)
{
        administrator.cancelService(this, service);
    }

    public void payArrears(double amount) {
        this.arrears.setAmount(this.arrears.getAmount() - amount);
        System.out.println("Абонент " + this.phoneNumber + " положил " +
amount + " на счет.");
        if(!this.isActive && !checkUnpaidArrears())
        {
            this.isActive = true;
            System.out.println
                    ("Абонент " + this.phoneNumber + " вновь подключен после
уплаты задолженности.");
        }
    }

    public void accountAmount()
    {
        if(checkUnpaidArrears())
        {
            System.out.println
                    ("У абонента " + this.phoneNumber + " имеется
```

```
задолженность суммой " + this.arrears.getAmount());
        }
        else
        {
            System.out.println
                    ("У абонента " + this.phoneNumber + " имеется остаток на
счете суммой " + (-this.arrears.getAmount()));
        }
    }

    public boolean checkUnpaidArrears() {
        return this.arrears.getAmount() > 0;
    }
}

class Administrator {
    public void changePhoneNumber(Subscriber subscriber, String newNumber) {
        System.out.println("Абонент с номером " + subscriber.phoneNumber + "
сменил номер телефона на " + newNumber);
        subscriber.phoneNumber = newNumber;
    }

    public void requestService(Subscriber subscriber, Service service) {
        subscriber.services.add(service);
        subscriber.arrears.setAmount(subscriber.arrears.getAmount() +
service.getPrice());
        System.out.println("Абонент " + subscriber.phoneNumber + "
подписался на услугу: " + service.getName());
    }

    public void cancelService(Subscriber subscriber, Service service) {
        if (subscriber.services.contains(service)) {
            subscriber.services.remove(service);
            subscriber.arrears.setAmount(subscriber.arrears.getAmount() -
service.getPrice());
            System.out.println("Абонент " + subscriber.phoneNumber + "
отписался от услуги: " + service.getName());
        }
    }

    public void temporarilyDisableSubscriber(Subscriber subscriber) {
        if (subscriber.checkUnpaidArrears()) {
            subscriber.isActive = false;
            System.out.println("Абонент " + subscriber.phoneNumber + "
отключен за неуплату.");
        }
    }
}

class TelephoneStation {
    private List<Subscriber> subscribers;

    public TelephoneStation() {
        this.subscribers = new ArrayList<>();
    }

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

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

public class Main3 {
```

```
    public static void main(String[] args) {
        TelephoneStation telephoneStation = new TelephoneStation();

        Service service1 = new Service("Интернет обслуживание", 34.0);
        Service service2 = new Service("Подписка на музыку", 89.0);
        Service service3 = new Service("Подписка на анекдоты", 15.0);
        Subscriber subscriber1 = new Subscriber("297228696");

        telephoneStation.addSubscriber(subscriber1);

        Administrator administrator = new Administrator();

        subscriber1.requestPhoneNumberChange(administrator, "336663432");

        subscriber1.payArrears(15);

        subscriber1.requestService(administrator, service1);
        subscriber1.requestService(administrator, service2);
        subscriber1.requestService(administrator, service3);

        subscriber1.accountAmount();

        subscriber1.cancelService(administrator, service2);

        administrator.temporarilyDisableSubscriber(subscriber1);

        subscriber1.payArrears(1000);

        subscriber1.accountAmount();


    }
}
```

Результат работы программы:

```
Абонент с номером 297228696 сменил номер телефона на 336663432
Абонент 336663432 положил 15.0 на счет.
Абонент 336663432 подписался на услугу: Интернет обслуживание
Абонент 336663432 подписался на услугу: Подписка на музыку
Абонент 336663432 подписался на услугу: Подписка на анекдоты
У абонента 336663432 имеется задолженность суммой 123.0
Абонент 336663432 отписался от услуги: Подписка на музыку
Абонент 336663432 отключен за неуплату.
Абонент 336663432 положил 1000.0 на счет.
Абонент 336663432 вновь подключен после уплаты задолженности.
У абонента 336663432 имеется остаток на счете суммой 966.0
```

Вывод: Приобрёл практические навыки в области объектно-ориентированного проектирования на языке Java.