

Министерство образования Республики Беларусь
Учреждение образования
“Брестский государственный технический университет”
Кафедра ИИТ

Отчёт
По лабораторной работе №4
По дисциплине СПП

Выполнил

Студент группы ПО-3
3-го курса
Куликович И. Т.

Проверил

Крощенко А. А.

Лабораторная работа №4

ВАРИАНТ 13

Задание 1. Реализовать указанный класс, включив в него вспомогательный внутренний класс или классы. Реализовать 2-3 метода (на выбор). Продемонстрировать использование реализованных классов.

Создать класс Зачетная Книжка с внутренним классом, с помощью объектов которого можно хранить информацию о сессиях, зачетах, экзаменах.

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. Создать класс Строка, используя классы Слово, Символ.

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.

Система Платежи. Клиент имеет Счет в банке и Кредитную Карту (КК). Клиент может оплатить Заказ, сделать платеж на другой Счет, заблокировать КК и аннулировать Счет. Администратор может заблокировать КК за превышение кредита.

Код программы

live.ilyusha.spp4.task1.GradeBook

```
package live.ilyusha.spp4.task1;
```

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

```
class GradeBook {
```

```
    private static class Exam {  
        String subject;  
        String lecturer;  
        int grade;
```

```
        public Exam(String subject, String lecturer, int grade) {  
            this.subject = subject;  
            this.lecturer = lecturer;  
            this.grade = grade;  
        }
```

```
        @Override  
        public String toString() {  
            return String.format("<Exam subject=\"%s\" teacher=\"%s\"  
mark=%d>", subject, lecturer, grade);  
        }
```

```
    }
```

```
    private ArrayList<Exam> exams = new ArrayList<>();  
    private String student;
```

```
    public GradeBook(String student) {  
        this.student = student;  
    }
```

```
    public void add(String subject, String lecturer, int grade) {
```

```

        exams.add(new Exam(subject, lecturer, grade));
    }

    public void log() {
        System.out.println(String.join("\n",
exams.stream().map(Exam::toString).toArray(String[]::new)));
    }
}

```

live.ilyusha.spp4.task1.Main

```

package live.ilyusha.spp4.task1;

public class Main {

    public static void main(String[] args) {
        GradeBook book = new GradeBook("Kulinkovich I. T.");
        book.add("Math", "Gladky", 4);
        book.add("YaP", "Oskar", 10);
        book.add("OSISP", "Kroschenko", 10);
        book.log();
    }
}

```

live.ilyusha.spp4.task2.CustomString

```

package live.ilyusha.spp4.task2;

import java.util.ArrayList;
import java.util.stream.Collectors;

class CustomString {

    private final ArrayList<Word> words = new ArrayList<>();

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

    @Override
    public String toString() {
        return
words.stream().map(Word::toString).collect(Collectors.joining(" "));
    }
}

```

live.ilyusha.spp4.task2.Symbol

```

package live.ilyusha.spp4.task2;

class Symbol {

    private char inner;

    public Symbol(char value) {
        this.inner = value;
    }
}

```

```

        public char getValue() {
            return inner;
        }
    }
}

live.ilyusha.spp4.task2.Word

package live.ilyusha.spp4.task2;

import java.util.ArrayList;
import java.util.stream.Collectors;

class Word {

    private ArrayList<Symbol> chars = new ArrayList<>();

    void add(Symbol symbol) {
        chars.add(symbol);
    }

    @Override
    public String toString() {
        return chars.stream().map(x ->
String.valueOf(x.getValue())).collect(Collectors.joining(""));
    }
}

```

live.ilyusha.spp4.task2.Main

```

package live.ilyusha.spp4.task2;

public class Main {

    public static void main(String[] args) {
        Symbol a = new Symbol('a');
        Symbol b = new Symbol('b');
        Symbol c = new Symbol('c');

        Word w1 = new Word();
        w1.add(a);
        w1.add(c);
        w1.add(b);

        Word w2 = new Word();
        w2.add(b);
        w2.add(c);

        Word w3 = new Word();
        w3.add(a);
        w3.add(b);
        w3.add(b);

        CustomString s = new CustomString();
        s.add(w1);
        s.add(w2);
        s.add(w3);
    }
}

```

```

        System.out.println(s.toString());
    }
}

```

live.ilyusha.spp4.task3.Administrator

```

package live.ilyusha.spp4.task3;

public class Administrator {

    private String name;
    private String company;

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

    public void blockCard(CreditCard card) {
        card.setBlocked(true);
    }

    public void unblockCard(CreditCard card) {
        card.setBlocked(false);
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getCompany() {
        return company;
    }

    public void setCompany(String company) {
        this.company = company;
    }
}

```

live.ilyusha.spp4.task3.BankAccount

```

package live.ilyusha.spp4.task3;

class BankAccount {

    private String number;
    private String currency;
    private double amount = 0;

    public BankAccount(String number, String currency) {
        this.number = number;
        this.currency = currency;
    }
}

```

```

/* codegen */

public String getNumber() {
    return number;
}

public void setNumber(String number) {
    this.number = number;
}

public String getCurrency() {
    return currency;
}

public void setCurrency(String currency) {
    this.currency = currency;
}

public double getAmount() {
    return amount;
}

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

```

live.ilyusha.spp4.task3.Client

```

package live.ilyusha.spp4.task3;

import java.util.ArrayList;

class Client {

    static class Order {

        private String id;
        private String location;
        private double value;

        public Order(String id, String location, double value) {
            this.id = id;
            this.location = location;
            this.value = value;
        }

        public String toString() {
            return String.format(
                "<Order id=\"%s\" location=\"%s\" value=%f>",
                id, location, value
            );
        }

        /* codegen */

        public String getId() {
            return id;
        }
    }
}

```

```

    public void setId(String id) {
        this.id = id;
    }

    public String getLocation() {
        return location;
    }

    public void setLocation(String location) {
        this.location = location;
    }

    public double getValue() {
        return value;
    }

    public void setValue(double value) {
        this.value = value;
    }
}

private BankAccount bankAccount;
private CreditCard creditCard;
private String name;

private ArrayList<Order> paymentHistory = new ArrayList<>();

public Client(BankAccount bankAccount, CreditCard creditCard, String
name) {
    this.bankAccount = bankAccount;
    this.creditCard = creditCard;
    this.name = name;
}

public void transferTo(BankAccount otherAccount, double amount) {
    if (bankAccount.getAmount() < amount) {
        throw new IllegalStateException("Not enough money");
    }
    if (creditCard.isBlocked()) {
        throw new IllegalStateException("Credit card is blocked");
    }
    bankAccount.setAmount(bankAccount.getAmount() - amount);
    otherAccount.setAmount(bankAccount.getAmount() + amount);
}

public void payForOrder(Order order) {
    if (bankAccount.getAmount() < order.getValue()) {
        throw new IllegalStateException("Not enough money");
    }
    if (creditCard.isBlocked()) {
        throw new IllegalStateException("Credit card is blocked");
    }
    bankAccount.setAmount(bankAccount.getAmount() - order.getValue());
    paymentHistory.add(order);
}

public void blockCard() {
    creditCard.setBlocked(true);
}

```

```

    }

    public void nullifyAccount() {
        bankAccount.setAmount(0);
    }

    /* codegen */

    public BankAccount getBankAccount() {
        return bankAccount;
    }

    public void setBankAccount(BankAccount bankAccount) {
        this.bankAccount = bankAccount;
    }

    public CreditCard getCreditCard() {
        return creditCard;
    }

    public void setCreditCard(CreditCard creditCard) {
        this.creditCard = creditCard;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public ArrayList<Order> getPaymentHistory() {
        return paymentHistory;
    }

    public void setPaymentHistory(ArrayList<Order> paymentHistory) {
        this.paymentHistory = paymentHistory;
    }
}

```

live.ilyusha.spp4.task3.CreditCard

```
package live.ilyusha.spp4.task3;
```

```
class CreditCard {
```

```

    private String issuer;
    private boolean blocked;
    private int cvv;
    private String number;
    private int expiryDateMonth;
    private int expiryDateYear;

```

```

    public CreditCard(String issuer, boolean isBlocked, int cvv, String
number, int expiryDateMonth, int expiryDateYear) {
        this.issuer = issuer;
        this.blocked = isBlocked;
        this.cvv = cvv;
    }

```



```

        this.number = number;
        this.expiryDateMonth = expiryDateMonth;
        this.expiryDateYear = expiryDateYear;
    }

    /* codegen */

    public boolean isBlocked() {
        return blocked;
    }

    public void setBlocked(boolean blocked) {
        this.blocked = blocked;
    }

    public int getCvv() {
        return cvv;
    }

    public void setCvv(int cvv) {
        this.cvv = cvv;
    }

    public String getNumber() {
        return number;
    }

    public void setNumber(String number) {
        this.number = number;
    }

    public int getExpiryDateMonth() {
        return expiryDateMonth;
    }

    public void setExpiryDateMonth(int expiryDateMonth) {
        this.expiryDateMonth = expiryDateMonth;
    }

    public int getExpiryDateYear() {
        return expiryDateYear;
    }

    public void setExpiryDateYear(int expiryDateYear) {
        this.expiryDateYear = expiryDateYear;
    }

    public String getIssuer() {
        return issuer;
    }

    public void setIssuer(String issuer) {
        this.issuer = issuer;
    }
}

```

live.ilyusha.spp4.task3.Main

package live.ilyusha.spp4.task3;

```

import java.util.stream.Collectors;

public class Main {

    public static void main(String[] args) {
        Administrator admin = new Administrator("Person Name", "MTBank");
        BankAccount account = new BankAccount("40817810099910004312", "USD");
        CreditCard card = new CreditCard("Alfa Bank", false, 765, "0184 6138
4302 7429", 12, 23);
        Client client = new Client(account, card, "I. Kulinkovich");

        client.getBankAccount().setAmount(5000);

        client.payForOrder(new Client.Order("4139", "Coffee", 30));
        client.payForOrder(new Client.Order("4141", "Phone shop", 100));

        admin.blockCard(client.getCreditCard());

        System.out.println(client.getPaymentHistory().stream()
            .map(Client.Order::toString).collect(Collectors.joining("\n")));

        try {
            client.payForOrder(new Client.Order("4140", "Sushi", 45));
        } catch (IllegalStateException e) {
            System.out.println(e.toString());
        }
    }
}

```

Спецификация ввода

>java Main

Пример

>java Main

Спецификация вывода

Для задачи 1:

<экзамен 1>

<экзамен 2>

...

<экзамен N>

Для задачи 2:

<слово>

Для задачи 3:

<история операций>

<ошибка вызванная неуспешной операцией>

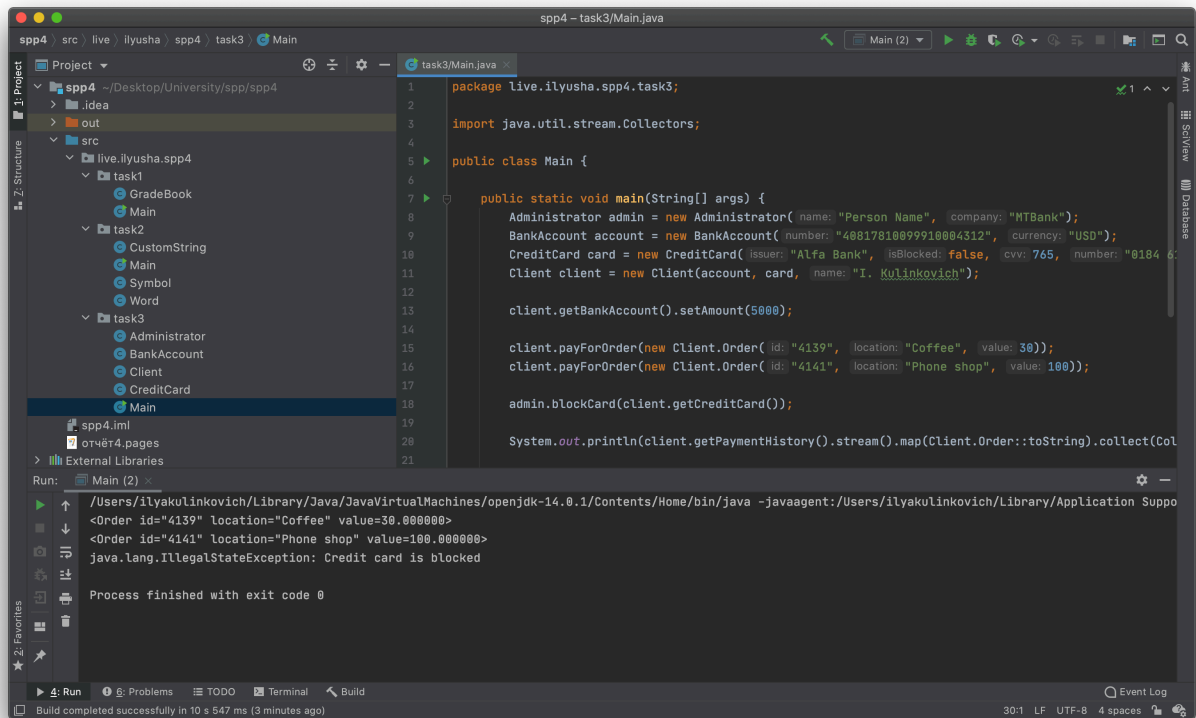
Пример

<Order id="4139" location="Coffee" value=30.000000>

<Order id="4141" location="Phone shop" value=100.000000>

java.lang.IllegalStateException: Credit card is blocked

Рисунки с результатами работы программы



The screenshot displays an IDE window titled 'spp4 - task3/Main.java'. The left sidebar shows a project structure with folders 'src', 'out', and 'task3'. The 'task3' folder contains files 'Main', 'Administrator', 'BankAccount', 'Client', and 'CreditCard'. The main editor shows the following Java code:

```
package live.ilyusha.spp4.task3;

import java.util.stream.Collectors;

public class Main {

    public static void main(String[] args) {
        Administrator admin = new Administrator( name: "Person Name", company: "MTBank");
        BankAccount account = new BankAccount( number: "40817810899910004312", currency: "USD");
        CreditCard card = new CreditCard( issuer: "Alfa Bank", isBlocked: false, cvv: 765, number: "0184 6...
        Client client = new Client(account, card, name: "I. Kulinkovich");

        client.getBankAccount().setAmount(5000);

        client.payForOrder(new Client.Order( id: "4139", location: "Coffee", value: 30));
        client.payForOrder(new Client.Order( id: "4141", location: "Phone shop", value: 100));

        admin.blockCard(client.getCreditCard());

        System.out.println(client.getPaymentHistory().stream().map(Client.Order::toString).collect(Collectors.toList()));
    }
}
```

The bottom panel shows the execution results for 'Main (2)'. The output is:

```
/Users/ilyakulinkovich/Library/Java/JavaVirtualMachines/openjdk-14.0.1/Contents/Home/bin/java -javaagent:/Users/ilyakulinkovich/Library/Application Support/
<Order id="4139" location="Coffee" value=30.000000>
<Order id="4141" location="Phone shop" value=100.000000>
java.lang.IllegalStateException: Credit card is blocked
Process finished with exit code 0
```

The status bar at the bottom indicates 'Build completed successfully in 10 s 547 ms (3 minutes ago)'.

Вывод

В данной лабораторной работе я приобрел практические навыки в области объектно-ориентированного проектирования.