

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ

УНИВЕРСИТЕТ»

КАФЕДРА «ИИТ»

Лабораторная работа №4

Выполнил:

Студент 4 курса

Группа АС-50

Годухин Р.А

Проверил:

Крощенко.А.А

Брест 2020

Вариант 2

Задание 1

Реализовать указанный класс, включив в него вспомогательный внутренний класс или классы. Реализовать 2-3 метода (на выбор).

Продемонстрировать использование реализованных классов.

2) Создать класс Payment (покупка) с внутренним классом, с помощью объектов которого можно сформировать покупку из нескольких товаров.

Код программы:

```
package com.company;

import java.util.ArrayList;
import java.util.Scanner;

class Payment{

    class Product{
        private int id;
        private String name;
        private double cost;
        private int count=0;

        public Product(int id, String name, double cost){
            this.id=id;
            this.name=name;
            this.cost=cost;
        }

        public void print(){
            System.out.println("id: "+id+", name: "+name+", cost: "+cost+"
p.");
        }

        public void printCount(){
            System.out.println("id: "+id+", name: "+name+", cost: "+cost+" p.
"+", count: "+count);
        }

        public double getSum(){
            return cost*count;
        }

        public Product clone(){
            Product product = new Product(id, name, cost);
            product.count = count;
            return product;
        }
    }

    private ArrayList<Product> products;
```

```

private ArrayList<Product> paymentProducts;

public Payment(){
    products = new ArrayList<Product>();

    products.add(new Product(0, "Молоко", 2.15));
    products.add(new Product(1, "Хлеб", 1.3));
    products.add(new Product(2, "Вода", 0.94));

    paymentProducts = new ArrayList<Product>();
}

public void printProducts(){
    for(int i=0; i<products.size(); i++)
        products.get(i).print();
}

public void printPaymentProducts(){
    double sum=0;

    for(int i=0; i<paymentProducts.size(); i++) {
        paymentProducts.get(i).printCount();
        sum+=paymentProducts.get(i).getSum();
    }
    String formatSum = String.format("%.2f", sum);
    System.out.println("Итого: "+formatSum+" p.");
}

public boolean addPaymentProduct(int id, int count){
    if(count<1)
        return false;

    for(int i=0; i<paymentProducts.size(); i++)
        if(paymentProducts.get(i).id == id)
        {
            paymentProducts.get(i).count += count;
            return true;
        }

    for(int i=0; i<products.size(); i++)
        if(products.get(i).id == id)
        {
            Product product = products.get(i).clone();
            product.count = count;
            paymentProducts.add(product);
            return true;
        }

    return false;
}

public boolean removePaymentProduct(int id){
    for(int i=0; i<paymentProducts.size(); i++)
        if(paymentProducts.get(i).id == id) {
            paymentProducts.remove(i);
            return true;
        }

    return false;
}

}

public class Main {

```

```

public static void main(String[] args) {
    Payment payment = new Payment();

    Scanner scan = new Scanner(System.in);
    for(boolean key=true; key; ){
        System.out.println("1 - посмотреть список доступных товаров");
        System.out.println("2 - посмотреть список товаров в моей
покупке");
        System.out.println("3 - добавить товар в мою покупку");
        System.out.println("4 - удалить товар из моей покупки");
        System.out.println("5 - выйти");

        int menuKey = Integer.parseInt(scan.nextLine());

        switch(menuKey) {
            case 1: payment.printProducts();
                System.out.println("Нажмите Enter:");break;
            case 2: payment.printPaymentProducts();
                System.out.println("Нажмите Enter:");break;
            case 3: System.out.println("Введите id товара:");
                int id = Integer.parseInt(scan.nextLine());
                System.out.println("Введите количество товара:");
                int count = Integer.parseInt(scan.nextLine());
                boolean result = payment.addPaymentProduct(id,
count);

                System.out.println("Нажмите Enter:");
                if(!result){
                    System.out.println("Произошла ошибка из-за
некорректных параметров");
                }
                break;
            case 4: System.out.println("Введите id товара:");
                int id2 = Integer.parseInt(scan.nextLine());
                boolean result2 = payment.removePaymentProduct(id2);
                if(!result2){
                    System.out.println("Произошла ошибка из-за
некорректных параметров");
                }
                System.out.println("Нажмите Enter:");break;
            case 5: key = false;
        }

        scan.nextLine();
        for(int i=0; i<10; i++)
            System.out.println();
    }
}

```

```
1 - посмотреть список доступных товаров
2 - посмотреть список товаров в моей покупке
3 - добавить товар в мою покупку
4 - удалить товар из моей покупки
5 - выйти

1
id: 0, name: Молоко, cost: 2.15 p.
id: 1, name: Хлеб, cost: 1.3 p.
id: 2, name: Вода, cost: 0.94 p.
```

1

```
3
Введите id товара:
2
Введите количество товара:
1
```

3

Нажмите Enter:

```
2
id: 2, name: Вода, cost: 0.94 p. , count: 1
Итого: 0,94 p.
```

2

Нажмите Enter:

```
4
Введите id товара:
2
```

4

Нажмите Enter:

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.

2) Создать класс Абзац, используя класс Строка.

Код программы:

```
package com.company;

class MyString{
    String text;
    boolean isUpperCase = false;

    public MyString(String text){
        this.text =text;
    }

    public void setIsUpperCase(boolean isUpperCase){
        this.isUpperCase = isUpperCase;
    }
}
```

```

    public String getText() {
        if (isUpperCase)
            return text.toUpperCase();
        else
            return text;
    }

    public void setText(String text) {
        this.text = text;
    }
}

class Paragraph {
    MyString myString;
    int indent = 0;

    public Paragraph(String text) {
        myString = new MyString(text);
    }

    public void setIndent(int indent) {
        this.indent = indent;
    }

    public void setText(String text) {
        myString.setText(text);
    }

    public void setIsUpperCase(boolean isUpperCase) {
        myString.setIsUpperCase(isUpperCase);
    }

    public void print() {
        for (int i = 0; i < indent; i++)
            System.out.print(" ");
        System.out.println(myString.getText());
    }
}

public class Main {

    public static void main(String[] args) {
        Paragraph paragraph = new Paragraph("Hello world!");
        paragraph.print();
        paragraph.setIndent(5);
        paragraph.setIsUpperCase(true);
        paragraph.print();
    }
}

```

Hello world!

HELLO WORLD!

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.

2) 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:

```
package com.company;

public class Main {

    public static void main(String[] args) {
        Account account1 = new Account(0);
        Account account2 = new Account(1);

        account1.setMoneyCount(30);
        account2.setMoneyCount(20.57);

        Chip chip = new Chip(0, "Semicond Inc");

        Card card1 = new Card(0, 200, chip);
        Card card2 = new Card(1, 30, chip);

        Client client1 = new Client(0, "Алексеев А.П", account1, card1);
        Client client2 = new Client(1, "Александров М.П.", account2, card2);

        String[] receivers = {"Столовая №2", "Интернет-магазин blizko", "Кафе\n"У дома\""};

        Order order1 = new Order(0, receivers[0], 12.5, false);
        Order order2 = new Order(1, receivers[2], 32, false);

        client1.setOrder(order1);
        client2.setOrder(order2);

        Admin admin = new Admin(0, "Главнов Г.Ю.");

        client1.show();
        System.out.println();
        client2.show();

        client2.payOrder(true);
        client1.paymentToOtherAccount(account2, 5, false);
        client1.blockCard();
        client1.cancelAccount();

        admin.blockCard(card2);
    }
}
```

```
        System.out.println();  
        System.out.println();  
        System.out.println("После действий:");  
        client1.show();  
        System.out.println();  
        client2.show();  
    }  
}
```

Клиент: 0; фιο: Алексеев А.П;
Счёт: 0; баланс: 30.0; аннулирован: false
Карта: 0; кредит: 0.0; макс.кредит: 200.0; заблокирована: false
Заказ: 0; получатель: Столовая №2; сумма: 12.5; оплачен: false

Клиент: 1; фιο: Александров М.П.;
Счёт: 1; баланс: 20.57; аннулирован: false
Карта: 1; кредит: 0.0; макс.кредит: 30.0; заблокирована: false
Заказ: 1; получатель: Кафе "У дома"; сумма: 32.0; оплачен: false

После действий:
Клиент: 0; фιο: Алексеев А.П;
Счёт: 0; баланс: 25.0; аннулирован: true
Карта: 0; кредит: 0.0; макс.кредит: 200.0; заблокирована: true
Заказ: 0; получатель: Столовая №2; сумма: 12.5; оплачен: false

Клиент: 1; фιο: Александров М.П.;
Счёт: 1; баланс: 25.57; аннулирован: false
Карта: 1; кредит: 32.0; макс.кредит: 30.0; заблокирована: true
Заказ: 1; получатель: Кафе "У дома"; сумма: 32.0; оплачен: true