

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ  
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ  
«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»  
ФАКУЛЬТЕТ ЭЛЕКТРОННО-ИНФОРМАЦИОННЫХ СИСТЕМ  
Кафедра интеллектуальных информационных технологий

Отчет по лабораторной работе №4

Выполнил

Булей Е.В.

студент группы АС50

Проверил

А. А. Крощенко,

ст. преп. кафедры ИИТ,

## Цель работы:

Научиться создавать и использовать классы в программах на языке программирования Java.

### **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:**

```
public class Payment {

    private int money;
    private ArrayList<Product> products; // = new ArrayList<Product>();

    public Payment(int money) {
        this.money = money;
        this.products = new ArrayList<>();
    }

    private void addInListProduct(Product product) {
        this.products.add(product);
    }

    public void listProducts() throws IOException {
        BufferedReader bufferedReader = new BufferedReader(new
        InputStreamReader(System.in));
        System.out.print("Enter counts products :");
        String num = bufferedReader.readLine();
        int number = Integer.parseInt(num);
        for(int i = 0; i<number;i++){
            String Name,Prise,Count;
            System.out.print("Enter name product :");
            String name = bufferedReader.readLine();
            System.out.print("Enter prise product :");
            Prise = bufferedReader.readLine();
            int prise = Integer.parseInt(Prise);
            System.out.print("Enter count product :");
            Count = bufferedReader.readLine();
            int count = Integer.parseInt(Count);
            Product product = new Product(name,prise,count);
            this.addInListProduct(product);
        }
    }

    public void buy() {
        for(Product product : this.products) {
            if (this.money > 0) {
```

```

        this.money -= product.priseProduct * product.countProduct;
        System.out.println("Buy " + product.nameProduct + ", count "
+ product.countProduct);
    } else {
        System.out.println("not money");
        System.out.println("Not buy " + product.nameProduct + ",
count " + product.countProduct);
    }
}

}

public int getMoney() {
    return money;
}

public void getBalance() {
    System.out.println("Balance : "+this.getMoney());
}

public class Product{
    private String nameProduct;
    private int priseProduct;
    private int countProduct;

    public Product(String nameProduct, int priseProduct, int
countProduct){
        this.nameProduct = nameProduct;
        this.priseProduct = priseProduct;
        this.countProduct = countProduct;
    }

}

}

```

```

public class Main {
    public static void main(String[] args) throws IOException {
        Payment payment = new Payment(1000);
        payment.listProducts();
        payment.buy();
        payment.getBalance();
    }
}

```

**Входные параметры:**

```

Enter counts products :3
Enter name product :chokalate
Enter prise product :180
Enter count product :2
Enter name product :bread
Enter prise product :70
Enter count product :1
Enter name product :milk
Enter prise product :80
Enter count product :2

```

### Вывод:

```
Buy chokalate, count 2
Buy bread, count 1
Buy milk, count 2
Balance : 410
```

## 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:

```
import java.util.ArrayList;

public class Paragraph {
    ArrayList<Line> paragraph;

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

    public void addLine() {
        Line line = new Line();
        paragraph.add(line);
    }

    public void print() {
        for (Line line : paragraph) {
            line.print();
        }
        System.out.println("");
    }

    public void addCountLine(int count) {
        for (int i = 0; i < count; i++) {
            this.addLine();
        }
    }
}
```

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

public class Line {
    private String line;
```

```

    public Line() {
        try {
            BufferedReader bufferedReader = new BufferedReader(new
InputStreamReader(System.in));
            line = bufferedReader.readLine();
            if(!(line.endsWith("!") || line.endsWith(".") ||
line.endsWith("?"))){
                StringBuilder stringBuilder = new StringBuilder(line);
                stringBuilder.insert(line.length(), ".");
                line = stringBuilder.toString();
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

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

    public boolean isEmpty(){
        return line.isEmpty();
    }

    public void print(){
        System.out.print(line);
    }
}

```

```

public class Main {
    public static void main(String[] args){
        Paragraph paragraph = new Paragraph();
        paragraph.addCountLine(5);
        paragraph.print();
    }
}

```

**Входные параметры:**

```

Надеюсь тут все хорошо)
И вы примите эту лабу.
Не хотелось бы еще мучаться с ней.
Пожалуйста?
И всего вам хорошего!

```

**Вывод:**

```

Надеюсь тут все хорошо).И вы примите эту лабу.Не хотелось бы еще мучаться с ней.Пожалуйста?И всего вам хорошего!

```

### 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:**

```
public class Klient {
    CreditCard creditCard;
    Adminictrator adminictrator = new Adminictrator();

    public void createCardAndAccount(String nameAccout,double money,String
nameCard) {
        this.creditCard = new CreditCard(new
BankAccount(nameAccout,money),nameCard);
    }

    public void buyProduct(Buy buy) {
        if (creditCard.isStatus() &&
!creditCard.bankAccount.getName().equals("Delete")) {
            creditCard.buyProduct(buy);
            if (creditCard.bankAccount.getMoney() < 0)
adminictrator.blockedCard(creditCard);
            return;
        }
        System.out.println("Card broken or dekete bank account");
    }

    public void blockedCard() {
        System.out.println("Card broken");
        creditCard.blockCard();
    }

    public void cancellation(){
        System.out.println("Delete accaunt");
        creditCard.bankAccount.cancellation();
    }

    public void giftMoney(BankAccount bankAccount, double money) {
        if (creditCard.isStatus() &&
!creditCard.bankAccount.getName().equals("Delete")) {
            creditCard.gifMoney(bankAccount, money);
            if (creditCard.bankAccount.getMoney() < 0)
adminictrator.blockedCard(creditCard);
            return;
        }
        System.out.println("Card broken or dekete bank account");
    }

    public CreditCard getCreditCard() {
        return creditCard;
    }

    public double money() {
```

```

        return creditCard.bankAccount.getMoney();
    }

    public BankAccount getBankAccount() {
        return creditCard.bankAccount;
    }
}

```

```

public class CreditCard{
    private String nameCard;
    private boolean status;
    BankAccount bankAccount;

    public CreditCard(BankAccount bankAccount, String nameCard) {
        this.bankAccount = bankAccount;
        this.nameCard = nameCard;
        this.status = true;
    }

    public void blockCard(){
        this.status = false;
    }

    public void gifMoney(BankAccount bankAccount, double money){
        if(bankAccount.isStatus()) {
            bankAccount.setMoney(bankAccount.getMoney() + money);
            this.bankAccount.setMoney(this.bankAccount.getMoney() - money);
            return;
        }
        System.out.println("Account delete");
    }

    public void buyProduct(Buy buy) {
        if(bankAccount.isStatus()) {
            System.out.println("Buy product "+buy.getNameProduct());
            this.bankAccount.setMoney(this.bankAccount.getMoney() -
buy.getPrice());
            return;
        }
        System.out.println("Account delete");
    }

    public boolean isStatus() {
        return status;
    }
}

```

```

public class Adminictrator {

    public void blockedCard(CreditCard creditCard){
        creditCard.blockCard();
        System.out.println("Card bloked by Adminictrator");
    }
}

```

```

public class BankAccount{
    private String name;

```

```

private double money;
private boolean status = true;

public BankAccount(String name, double money) {
    this.name = name;
    this.money = money;
}

public void cancellation(){
    this.money = 0;
    this.name = "Delete";
    this.status = false;
}

public String getName() {
    return name;
}

public double getMoney() {
    return money;
}

public void setMoney(double money) {
    this.money = money;
}

public boolean isStatus() {
    return status;
}
}

```

```

public class Buy {
    private String nameProduct;
    private double price;

    public Buy(String nameProduct, double price) {
        this.nameProduct = nameProduct;
        this.price = price;
    }

    public String getNameProduct() {
        return nameProduct;
    }

    public double getPrice() {
        return price;
    }
}

```

### Входные параметры:

```

public class Main {
    public static void main(String[] args){
        Buy buy = new Buy("Gold",500);
        Klient klient = new Klient();
        klient.createCardAndAccount("AAAAAAAAAA",800,"vix");
        Klient klient1 = new Klient();
        klient1.createCardAndAccount("BBBBBBBBB",1000,"zek");
        klient.buyProduct(buy);
        System.out.println("Money - "+klient.money());
        klient.giftMoney(klient1.getBankAccount(),400);
    }
}

```



```
System.out.println("Money klient - "+klient.money());  
System.out.println("Money klient1 - "+klient1.money());  
  
klient1.blockedCard();  
klient1.cancellation();  
}  
}
```

#### **Вывод:**

```
Buy product Gold  
Money - 300.0  
Card bloked by Adminictrator  
Money klient - -100.0  
Money klient1 - 1400.0  
Card broken  
Delete accaunt  
  
Process finished with exit code 0
```

**Вывод:** Приобрел практические навыки создания и использования классов в программах на языке программирования Java.