

Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №6

По дисциплине «Современные платформы программирования»

Выполнила:

Студентка 3 курса

Группы ПО-3

Гаврилкович Е.В.

Проверил:

Крощенко А.А.

Брест 2020 г.

## Цель работы:

Приобрести навыки применения паттернов проектирования при решении практических задач с использованием языка Java

## Ход работы

### Задание 1:

Завод по производству смартфонов. Обеспечить создание нескольких различных моделей мобильных телефонов с заранее выбранными характеристиками.

### Текст программы:

```
import java.util.ArrayList;
public class task1
{
    public static void main(String[] args) {
        Apple f1 = (Apple)FactoryPhone.createPhone("Apple", 32, "ellow");
        FactoryPhone f2 = FactoryPhone.createPhone("Samsung", 64, "ellow");
        f1.sayCharacteristics();
    }
}
class FactoryPhone{
    public static FactoryPhone createPhone(String model, int mem, String
color){
        if (model.equals("Apple")){
            System.out.println("Apple phone crate!");
            return new Apple(mem, color);
        }
        if (model.equals("Samsung")){
            System.out.println("Samsung phone crate!");
            return new Samsung(mem, color);
        }
        else{
            System.out.println("Error");
        }
        return null;
    }
}
class Apple extends FactoryPhone {
    int mem;
    String color;
    public Apple(int mem, String color){
        this.mem=mem;
        this.color=color;
    }
    public void sayCharacteristics() {
```

```

        System.out.println("mem = "+this.mem+" color = "+ this.color);
    }
}
class Samsung extends FactoryPhone {
    int mem;
    String color;
    public Samsung(int mem, String color){
        this.mem=mem;
        this.color=color;
    }
    public void sayCharacteristics(int mem, String color) {
        System.out.println("mem = "+this.mem+" color = "+ this.color);
    }
}

```

### Рисунок с результатом работы программы:

```

/usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/java -javaagent:/home/kate/.local/share/umake/ide/idea/lib/idea_rt.jar=45023:/home
Name: Samsung
Diagonal: 5
RAM: 8
Camera: 16

Name: Sony
Diagonal: 6
RAM: 6
Camera: 12

```

### Задание 2:

Проект «Электронный градусник». В проекте должен быть реализован класс, который дает возможность пользоваться аналоговым градусником так же, как и электронным. В классе «Аналоговый градусник» хранится высота ртутного столба и границы измерений (верхняя и нижняя).

### Текст программы:

```

public class task2{
    public static void main(String[] args) {
        ElectronicThermometer t1 = new ElectronicThermometer();
        MercuryThermometer t2 =new MercuryThermometer(100, 0, 30);
        Thermometer.measure(t1, 1000);
        t1.getT();
        Thermometer.measure(t2, 25);
        t2.getT();
        Thermometer.measure(t2, 1000);
        t2.getT();
    }
}

```

```

}
class ElectronicThermometer{
    int T;
    ElectronicThermometer(){
        this.T = 0;
    }
    ElectronicThermometer(int t){
        this.T = t;
    }
    void measure(int t){
        this.T = t;
    }
    void getT(){
        System.out.println("T = "+ this.T);
    }
}
class MercuryThermometer{
    int minT;
    int maxT;
    int T;
    MercuryThermometer(int max, int min){
        this.minT = min;
        this.maxT = max;
    }
    MercuryThermometer(int max, int min, int t){
        this.minT = min;
        this.maxT = max;
        if (t <= this.maxT && t >= this.minT){
            this.T = t;
        }else {
            this.T = this.minT;
        }
    }
    void measure(int t){
        if (t <= this.maxT && t >= this.minT){
            this.T = t;
        }
    }
    void getT(){
        System.out.println("T = "+ this.T);
    }
}
class Thermometer{
    static public void measure(MercuryThermometer m, int t){
        m.measure(t);
    }
    static public void measure(ElectronicThermometer m, int t){
        m.measure(t);
    }
}

```

```
}  
}
```

### Рисунок с результатом работы программы:

```
T = 1000  
T = 25  
T = 25
```

### Задание 3:

Проект «Банкомат». Предусмотреть выполнение основных операций (ввод пин-кода, снятие суммы, завершение работы) и наличие различных режимов работы (ожидание, аутентификация, выполнение операции, блокировка – если нет денег). Атрибуты: общая сумма денег в банкомате, ID.

### Текст программы:

```
import java.util.ArrayList;  
public class task3{  
    public static void main(String[] args) {  
        ATM atm = new ATM(1000, 1, 4550);  
        atm.withdrawalMoney(33);  
        atm.authorization(4550);  
        atm.withdrawalMoney(33);  
        atm.end();  
    }  
}  
class Session{  
    int pass;  
    boolean authorization;  
    Session(int p){  
        this.pass= p;  
        this.authorization = false;  
    }  
    boolean authorization (int p){  
        if (p == this.pass){  
            this.authorization = true;  
            return true;  
        }  
        return false;  
    }  
    void end(){  
        this.authorization = false;  
        System.out.println("Сессия закрыта");  
    }  
}
```

```

        boolean getStatus(){

            return this.authorization;
        }
    }
}

class Mode{
    String modename;
    Mode(){
        this.modename = "waiting";
        System.out.println("Mode = "+this.modename);
    }
    void setWaitng(){
        this.modename = "waiting";
        System.out.println("Mode = "+this.modename);
    }
    void setAuthorization(){
        this.modename = "authorization";
        System.out.println("Mode = "+this.modename);
    }
    void setPerformance(){
        this.modename = "performance";
        System.out.println("Mode = "+this.modename);
    }
    void setBlocking(){
        this.modename = "blocking";
        System.out.println("Mode = "+this.modename);
    }
    String getMode(){
        return this.modename;
    }
}

class ATM{
    int allmoney;
    int id;
    Mode mode;
    Session ses;
    ATM(int am, int i, int pass){
        this.allmoney = am;
        this.id = i;
        this.mode = new Mode();
        this.ses = new Session(pass);
    }
    void authorization(int pass){
        if (this.mode.getMode().equals("waiting")){
            if (this.ses.authorization(pass)){
                this.mode.setAuthorization();
            }
        }
        else{
            System.out.println("Error");
        }
    }
}

```

```

    }
}
}
void withdrawalMoney(int outmaoney){
    if(this.mode.getMode().equals("authorization") &&
this.ses.getStatus()){
        this.mode.setPerformance();
        if(this.allmoney >= outmaoney){
            this.allmoney-=outmaoney;
            System.out.println("С н я т о = "+ outmaoney);
            this.mode.setAuthorization();
        }else{
            this.mode.setBlocking();
            this.ses.end();
            System.out.println("Н е х в а т к а   д е н е г");
        }
    }else{
        System.out.println("П о л ь з о в а т е л ь   н е
а в т о р е з и р о в а н");
    }
}
void end(){
    this.ses.end();
    this.mode.setWaitng();
}
}

```

**Рисунок с результатом работы программы:**

```

Mode = waiting
Пользователь не авторезирован
Mode = authorization
Mode = performance
Снято = 33
Mode = authorization
Сессия закрыта
Mode = waiting

```

**Выводы:**

Закрепили базовые знания языка программирования Java при решении практических задач.

