

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

ФАКУЛЬТЕТ ЭЛЕКТРОННО-ИНФОРМАЦИОННЫХ СИСТЕМ

Кафедра интеллектуальных информационных технологий

Отчет по лабораторной работе №5

Специальность ПОЗ

Выполнила Р.И.
Гаврилюк, студентка
группы ПОЗ

Проверил А.А.
Крощенко,
ст. преп. кафедры ИИТ,
«—» ————— 2020 г.

Брест 2020

Вариант 6

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

Задание 1.

Реализовать абстрактные классы или интерфейсы, а также наследование и полиморфизм для следующих классов:

6) interface Mobile ← abstract class Samsung Mobile ← class Model.

Код программы

Task_1.java

```
class Task1{
    public static void main(String[] args) {
        Model model = new Model("a50", 400, "abc123");
        model.print();
    }
}
```

IMobile.java

```
interface IMobile{
    String getName();
    int getPrice();

    void print();
}
```

SamsungMobile.java

```
abstract class SamsungMobile implements IMobile {

    private String mName;
    private int mPrice;

    public SamsungMobile(String name, int price){
        this.mName = name;
        this.mPrice = price;
    }

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

    public void setPrice(int price) {
        this.mPrice = price;
    }

    @Override
    public String getName(){
        return this.mName;
    }
}
```

```

@Override
public int getPrice(){
    return this.mPrice;
}
}

```

Model.java

```

class Model extends SamsungMobile {

    private String mBatchNumber;

    public Model(String name, int price, String batchNumber){
        super(name, price);
        this.mBatchNumber = batchNumber;
    }

    public void setBranchNumber(String batchNumber){
        this.mBatchNumber = batchNumber;
    }

    public String getBachNumber(){
        return this.mBatchNumber;
    }

    @Override
    public void print(){
        System.out.println("Samsung Mobile " + this.getName() +
            " with batch number " + this.mBatchNumber +
            " costs " + this.getPrice());
    }
}

```

Рисунки с результатами работы программы

```

Samsung Mobile a50 with batch number abc123 costs 400

```

Задание 2.

В следующих заданиях требуется создать суперкласс (абстрактный класс, интерфейс) и определить общие методы для данного класса. Создать подклассы, в которых добавить специфические свойства и методы. Часть методов переопределить. Создать массив объектов суперкласса и заполнить объектами подклассов. Объекты подклассов идентифицировать конструктором по имени или идентификационному номеру. Использовать объекты подклассов для моделирования реальных ситуаций и объектов.

6) Создать суперкласс Домашнее животное и подклассы Собака, Кошка, Попугай. С помощью конструктора установить имя каждого животного и его характеристики.

Код программы

Task_2.java

```
import java.util.ArrayList;
class Task_2{
    public static void main(String[] args) {
        ArrayList<Pet> pets = new ArrayList<>();
        pets.add(new Cat("Matroskin", "fine"));
        pets.add(new Dog("Sharik", "good"));
        pets.add(new Parrot("Kesha", "excellent"));
        for(Pet pet : pets){
            pet.callPet();
            System.out.println();
        }
    }
}
```

Pet.java

```
abstract class Pet{
    private String mName;
    private String mMood;
    private String mVoice;

    public Pet(String name, String mood, String voice){
        this.mName = name;
        this.mMood = mood;
        this.mVoice = voice;
    }

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

    public void setMood(String mood){
        this.mMood = mood;
    }

    public void setVoice(String voice){
        this.mVoice = voice;
    }

    public String getName(){
        return this.mName;
    }

    public String getMood(){
        return this.mMood;
    }

    public String getVoice(){
        return this.mVoice;
    }

    public abstract void callPet();
}
```

Cat.java

```
class Cat extends Pet{
    private String mDocuments;

    public Cat(String name, String mood){
        super(name, mood, "We have the funds.\nWe are not smart enough.");
        this.mDocuments = "Whiskers, paws and tail";
    }

    public void setDocuments(String documents) {
        this.mDocuments = documents;
    }

    public String getDocuments() {
        return this.mDocuments;
    }

    @Override
    public void callPet(){
        System.out.println(this.getVoice() + "\nHi, I'm cat " +
            this.getName() + ". I'm " + this.getMood() +
            "\n" + this.mDocuments + " here are my documents");
    }
}
```

Dog.java

```
class Dog extends Pet{
    private String mMainOnTheTable;

    public Dog(String name, String mood){
        super(name, mood, "Sharik. I'm a simple dog, not a purebred one.");
        this.mMainOnTheTable = "Bonelet";
    }

    public void setMainOnTheTable(String mainOnTheTable) {
        this.mMainOnTheTable = mainOnTheTable;
    }

    public String getMainOnTheTable() {
        return this.mMainOnTheTable;
    }

    @Override
    public void callPet(){
        System.out.println(this.getVoice() + "\nHi, I'm dog " +
            this.getName() + ". I'm " + this.getMood() +
            "\n-What is main on the table?\n-" + this.mMainOnTheTable);
    }
}
```

Parrot.java

```
class Parrot extends Pet{
    private String mQuote;

    public Parrot(String name, String mood){
        super(name, mood, "Freedom for parrots!\nFre-e-dom for pa-rrots!");
        this.mQuote = "I can do it! I'll prove it!.. I'll prove it!.. \nThey'll find out about me. people will talk about me...";
    }

    public void setQuote(String quote) {
        this.mQuote = quote;
    }

    public String getQuote() {
        return this.mQuote;
    }

    @Override
    public void callPet(){
        System.out.println(this.getVoice() + "\nHi, I'm prodigal parrot " +
            this.getName() + ". I'm " + this.getMood() +
            "\n" + this.mQuote);
    }
}
```

Рисунки с результатами работы программы

```
We have the funds.
We are not smart enough.
Hi, I'm cat Matroskin. I'm fine
Whiskers, paws and tail here are my documents

Sharik. I'm a simple dog, not a purebred one.
Hi, I'm dog Sharik. I'm good
-What is main on the table?
-Bonelet

Freedom for parrots!
Fre-e-dom for pa-rrots!
Hi, I'm prodigal parrot Kesha. I'm excelent
I can do it! I'll prove it!.. I'll prove it!..
They'll find out about me. people will talk about me...
```

Задание 3.

В задании 3 ЛР №4, где возможно, заменить объявления суперклассов объявлениями абстрактных классов или интерфейсов.

Код программы

```
Abstract class Service{ ... }
class ServiceMMS extends Service{ ... }
class ServiceSMS extends Service{ ... }
class ServiceIntenet extends Service{ ... }
```

```
abstract class User{ ... }  
class Abonent extends User{ ... }  
class Administrator extends User{ ... }
```

Вывод: приобрела практические навыки в области объектно-ориентированного проектирования.