



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.04.01 Информатика и вычислительная техника

МАГИСТЕРСКАЯ ПРОГРАММА 09.04.01/12 Интеллектуальный анализ больших
данных в системах поддержки принятия решений.

О Т Ч Е Т

по лабораторной работе № 4

Вариант 6

Название: Внутренние классы, интерфейсы

Дисциплина: Языки программирования для работы с большими данными

Студент

ИУ6-23М

(Группа)

(Подпись, дата)

Г.Л. Кушнир

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

П.В. Степанов

(И.О. Фамилия)

Москва, 2023

Цель: Изучить работу внутренних классов и интерфейсов в java.

Вариант 1 задание 6: Создать класс Shop (магазин) с внутренним классом, с помощью объектов которого можно хранить информацию об отделах, товарах и услуг.

Код класса Shop:

```
public class Shop {

    private final ArrayList<Shop.ShopInfo> InfoService;

    public Shop(){
        InfoService = new ArrayList<>();
    }

    public void addInfo(String department, String product, String service){

        ShopInfo info = new ShopInfo();
        info.setDepartment(department);
        info.setProduct(product);
        info.setService(service);
        InfoService.add(info);

    }

    public ArrayList<Shop.ShopInfo> getInfoService() {
        return InfoService;
    }

    public static class ShopInfo{

        private String department;
        private String product;
        private String service;

        public void setDepartment(String department) {
            this.department = department;
        }

        public String getDepartment() {
            return department;
        }

        public void setProduct(String product) {
            this.product = product;
        }

        public String getProduct() {
            return product;
        }

        public void setService(String service) {
            this.service = service;
        }

        public String getService() {
            return service;
        }

    }

}
```

```
}  
}
```

Вариант 1 задание 7: Создать класс Справочная Служба Общественного Транспорта с внутренним классом, с помощью объектов которого можно хранить информацию о времени, линиях маршрутов и стоимости проезда.

Код класса PubTransportInfService:

```
public class PubTransportInfService {  
  
    private final ArrayList<RouteInfo> InfoService;  
  
    public PubTransportInfService(){  
        InfoService = new ArrayList<>();  
    }  
  
    public void addInfo(Calendar time, String route, double cost){  
  
        RouteInfo info = new RouteInfo();  
        info.setTime(time);  
        info.setRoute(route);  
        info.setCost(cost);  
        InfoService.add(info);  
  
    }  
  
    public ArrayList<RouteInfo> getInfoService() {  
        return InfoService;  
    }  
  
    public static class RouteInfo{  
  
        private Calendar time;  
        private String route;  
        private double cost;  
  
        public void setTime(Calendar time) {  
            this.time = time;  
        }  
  
        public Calendar getTime() {  
            return time;  
        }  
  
        public void setRoute(String route) {  
            this.route = route;  
        }  
  
        public String getRoute() {  
            return route;  
        }  
  
        public void setCost(double cost) {  
            this.cost = cost;  
        }  
  
        public double getCost() {
```

```

        return cost;
    }
}
}

```

Вариант 2 задание 6: Реализовать абстрактные классы или интерфейсы, а также наследование и полиморфизм для следующих классов: interface Корабль <- abstract class Военный Корабль <- class Авианосец.

Код интерфейса Ship:

```

public interface Ship {

    void setShipName();
    String getShipName();

}

```

Код абстрактного класса Warship:

```

public abstract class Warship implements Ship{

    protected String Type;
    abstract String getType();

}

```

Код класса Aerocarrier:

```

public class Aerocarrier extends Warship{

    private String shipName;

    public Aerocarrier(String shipName){
        this.Type = "Aero Carrier";
        this.shipName = shipName;
    }

    @Override
    public String getType(){
        return Type;
    }

    @Override
    public void setShipName(){
        System.out.print("Input name of the ship: ");
        Scanner in = new Scanner(System.in);
        shipName = in.nextLine();
    }

    @Override
    public String getShipName(){
        return shipName;
    }

}

```

Вариант 2 задание 7: Реализовать абстрактные классы или интерфейсы, а также наследование и полиморфизм для следующих классов: interface Врач <- class Хирург <- class Нейрохирург.

Код интерфейса Doctor:

```
public interface Doctor {  
  
    void treatPatient();  
  
}
```

Код класса Surgeon:

```
public class Surgeon implements Doctor{  
  
    protected String DoctorType = "Surgeon";  
    @Override  
    public void treatPatient() {  
        System.out.println(DoctorType);  
    }  
  
}
```

Код класса Neurosurgeon:

```
public class Neurosurgeon extends Surgeon{  
  
    public Neurosurgeon() {  
        DoctorType = "Neurosurgeon";  
    }  
  
}
```

Код основного класса Main:

```
public class Main {  
  
    public static void main(String[] args) {  
  
        // Вариант 1 задание 6  
  
        Shop shopDirectory = new Shop();  
  
        shopDirectory.addInfo("Pharmacy", "Drugs", "Buying");  
        shopDirectory.addInfo("Confectionery", "Sweets", "Buying");  
        shopDirectory.addInfo("Jeweller's", "Jewelry", "Selling");  
  
        for (Shop.ShopInfo info : shopDirectory.getInfoService()) {  
            System.out.println("Department: " + info.getDepartment());  
            System.out.println("Product: " + info.getProduct());  
            System.out.println("Service: " + info.getService());  
            System.out.println();  
        }  
  
        System.out.println("-----  
-----");  
  
        // Вариант 1 задание 7  
  
        PubTransportInfService transportDirectory = new  
        PubTransportInfService();  
  
        Calendar time = new GregorianCalendar();  
        time.set(Calendar.HOUR_OF_DAY, 12);  
        time.set(Calendar.MINUTE, 30);  
        transportDirectory.addInfo(time, "Route 1", 2.50);  
    }  
}
```

```

        time.set(Calendar.HOUR_OF_DAY, 15);
        time.set(Calendar.MINUTE, 45);
        transportDirectory.addInfo(time, "Route 2", 3.00);

        for (PubTransportInfService.RouteInfo info :
transportDirectory.getInfoService()) {
            System.out.println("Time: " +
info.getTime().get(Calendar.HOUR_OF_DAY) + ":" +
info.getTime().get(Calendar.MINUTE));
            System.out.println("Route: " + info.getRoute());
            System.out.println("Cost: " + info.getCost());
            System.out.println();
        }

        System.out.println("-----");

        // Вариант 2 задание 6

        Aerocarrier ship = new Aerocarrier("Nevada");
        System.out.println("Name of ship: " + ship.getShipName());
        System.out.println("Type of ship: " + ship.getType());

        System.out.println("-----");

        // Вариант 2 задание 7

        Neurosurgeon doctor = new Neurosurgeon();
        doctor.treatPatient();

    }
}

```

Работа программы представлена на рисунке 1.

```

Department: Pharmacy
Product: Drugs
Service: Buying

Department: Confectionery
Product: Sweets
Service: Buying

Department: Jeweller's
Product: Jewelry
Service: Selling

-----

Time: 15:45
Route: Route 1
Cost: 2.5

Time: 15:45
Route: Route 2
Cost: 3.0

-----

Name of ship: Nevada
Type of ship: Aero Carrier
-----

Neurosurgeon

```

Рисунок 1 – Работа программы

Вывод: Была изучена работа внутренних классов и интерфейсов в java.