

**Министерство образования Республики Беларусь
Учреждение образования
«Брестский государственный технический университет»
Кафедра ИИТ**

**Лабораторная работа №4
По дисциплине: «ССП»
Вариант 10**

Выполнил:
Студент 3 курса
Группы ПО-8
Дымша А.Г.
Проверил:
Крощенко А.А

Брест 2024

Лабораторная работа №4

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

Задание 1: Реализовать указанный класс, включив в него вспомогательный внутренний класс или классы. Реализовать 2-3 метода (на выбор). Продемонстрировать использование реализованных классов. 1) Создать класс Notepad (записная книжка) с внутренним классом или классами, с помощью объектов которого могут храниться несколько записей на одну дату.

Код программы:

```
import java.time.LocalDate;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

class Notepad {
    private Map<LocalDate, List<Entry>> entries;

    public Notepad() {
        entries = new HashMap<>();
    }

    public void addEntry(LocalDate date, String text) {
        List<Entry> entriesForDate = entries.computeIfAbsent(date, k -> new
ArrayList<>());
        entriesForDate.add(new Entry(text));
    }

    public List<Entry> getEntries(LocalDate date) {
        return entries.getDefault(date, new ArrayList<>());
    }

    public static void main(String[] args) {
        Notepad notepad = new Notepad();

        LocalDate today = LocalDate.now();
        notepad.addEntry(today, "Купить продукты");
        notepad.addEntry(today, "Позвонить другу");

        LocalDate tomorrow = today.plusDays(1);
        notepad.addEntry(tomorrow, "Встретиться с коллегой");

        System.out.println("Записи на сегодня:");
        for (Entry entry : notepad.getEntries(today)) {
            System.out.println(entry.getText());
        }
    }
}
```

```

        System.out.println("\nЗаписи на завтра:");
        for (Entry entry : notepad.getEntries(tomorrow)) {
            System.out.println(entry.getText());
        }
    }

    private static class Entry {
        private final String text;

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

        public String getText() {
            return text;
        }
    }
}

```

Результат работы программы:

```

Записи на сегодня:
Купить продукты
Позвонить другу

Записи на завтра:
Встретиться с коллегой

```

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.

10) Создать класс Планета, используя класс Материк.

Код программы:

```

import java.util.ArrayList;
import java.util.List;

class Planet {
    private String name;
    private List<Continent> continents;

    public Planet(String name) {
        this.name = name;
        this.continents = new ArrayList<>();
    }
}

```

```

public void addContinent(Continent continent) {
    continents.add(continent);
}

public void removeContinent(Continent continent) {
    continents.remove(continent);
}

public List<Continent> getContinents() {
    return continents;
}

public String getName() {
    return name;
}

public static void main(String[] args) {
    Planet earth = new Planet("Земля");

    Continent asia = new Continent("Азия", 44579000);
    Continent africa = new Continent("Африка", 30370000);
    Continent northAmerica = new Continent("Северная Америка", 24709000);

    earth.addContinent(asia);
    earth.addContinent(africa);
    earth.addContinent(northAmerica);

    System.out.println("Планета: " + earth.getName());
    System.out.println("Материки:");

    for (Continent continent : earth.getContinents()) {
        System.out.println("- " + continent.getName() + ", площадь: " +
continent.getArea() + " кв. км");
    }
}

class Continent {
    private String name;
    private double area;

    public Continent(String name, double area) {
        this.name = name;
        this.area = area;
    }

    public String getName() {

```

```

        return name;
    }

    public double getArea() {
        return area;
    }
}

```

Результаты работы программы:

```

Планета: Земля
Материки:
- Азия, площадь: 4.4579E7 кв. км
- Африка, площадь: 3.037E7 кв. км
- Северная Америка, площадь: 2.4709E7 кв. км

```

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. 10) Система Городской транспорт. На Маршрут назначаются Автобус или Троллейбус. Транспортные средства должны двигаться с определенным для каждого Маршрута интервалом. При поломке на Маршрут должен выходить резервный транспорт или увеличиваться интервал движения.

```

Код программы: import java.util.ArrayList;
import java.util.List;

// Интерфейс Транспортное Средство
interface Vehicle {
    void move();
}

// Класс Автобус (реализация интерфейса Vehicle)
class Bus implements Vehicle {
    @Override
    public void move() {
        System.out.println("Автобус движется");
    }
}

// Класс Троллейбус (реализация интерфейса Vehicle)
class Trolleybus implements Vehicle {
    @Override
    public void move() {
        System.out.println("Троллейбус движется");
    }
}

```

```

// Абстрактный класс Маршрут
abstract class Route {
    protected List<Vehicle> vehicles;
    protected int interval;
    protected List<Vehicle> reserveVehicles;

    public Route(int interval) {
        this.interval = interval;
        this.vehicles = new ArrayList<>();
        this.reserveVehicles = new ArrayList<>();
    }

    public void addVehicle(Vehicle vehicle) {
        vehicles.add(vehicle);
    }

    public void addReserveVehicle(Vehicle vehicle) {
        reserveVehicles.add(vehicle);
    }

    public void startRoute() {
        System.out.println("Маршрут начал движение. Интервал: " + interval + "
минут.");
        for (Vehicle vehicle : vehicles) {
            vehicle.move();
        }
    }

    public void breakdownHandler() {
        System.out.println("Поломка транспортного средства на маршруте!");
        if (!reserveVehicles.isEmpty()) {
            Vehicle reserveVehicle = reserveVehicles.remove(0);
            vehicles.add(reserveVehicle);
            System.out.println("Резервное транспортное средство вышло на
маршрут.");
        } else {
            System.out.println("Резервных транспортных средств нет. Увеличиваем
интервал движения.");
            interval *= 2;
        }
    }
}

// Конкретный класс Автобусный Маршрут (агрегация с классом Автобус)
class BusRoute extends Route {
    public BusRoute(int interval) {
        super(interval);
    }
}

```

```
}

// Конкретный класс Троллейбусный Маршрут (агрегация с классом Троллейбус)
class TrolleybusRoute extends Route {
    public TrolleybusRoute(int interval) {
        super(interval);
    }
}

class CityTransportSystem {
    public static void main(String[] args) {
        BusRoute busRoute = new BusRoute(10);
        busRoute.addVehicle(new Bus());
        busRoute.addVehicle(new Bus());
        busRoute.addReserveVehicle(new Bus());

        TrolleybusRoute trolleybusRoute = new TrolleybusRoute(15);
        trolleybusRoute.addVehicle(new Trolleybus());
        trolleybusRoute.addVehicle(new Trolleybus());
        trolleybusRoute.addReserveVehicle(new Trolleybus());

        busRoute.startRoute();
        busRoute.breakdownHandler();

        trolleybusRoute.startRoute();
        trolleybusRoute.breakdownHandler();
        trolleybusRoute.breakdownHandler();
    }
}
```

Результаты работы программы:

```
Маршрут начал движение. Интервал: 10 минут.  
Автобус движется  
Автобус движется  
Поломка транспортного средства на маршруте!  
Резервное транспортное средство вышло на маршрут.  
Маршрут начал движение. Интервал: 15 минут.  
Троллейбус движется  
Троллейбус движется  
Поломка транспортного средства на маршруте!  
Резервное транспортное средство вышло на маршрут.  
Поломка транспортного средства на маршруте!  
Резервных транспортных средств нет. Увеличиваем интервал движения.
```

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