МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
КАФЕДРА ИИТ

Лабораторная работа №5

По дисциплине: «Современные платформы программирования»

Выполнил:
Студент 3 курса
группы ПО-8:
Макаревич Е.С.
Проверил:
Крощенко А.А.

Брест 2024

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

Вариант 17

Задание 1. Реализовать абстрактные классы или интерфейсы, а также наследование и полиморфизм для следующих классов:
interface Учебное Заведение ← class Колледж ← class Университет.

Код программы:

College.java

```
package task01;

public class College implements EducationalInstitution {
    private String name;
    private String location;

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

    @Override
    public String getName() {
        return name;
    }

    @Override
    public void setLocation(String location) {
        this.location = location;
    }

    @Override
    public String getLocation() {
        return location;
    }
}
```

EducationalInstitution.java

```
package task01;

public interface EducationalInstitution {
    void setName(String name);
    String getName();
    void setLocation(String location);
    String getLocation();
}
```

University.java

```
package task01;

public class University extends College {
```

```
    private String accreditation;

    public void setAccreditation(String accreditation) {
        this.accreditation = accreditation;
    }
    public String getAccreditation() {
        return accreditation;
    }
}
```

Main.java

```
package task01;

  public class Main {
      public static void main(String[] args) {
          // Создание объекта колледжа
          College college = new College();
          college.setName("Community College");
          college.setLocation("City Center");

          // Вывод информации о колледже
          System.out.println("College Name: " + college.getName());
          System.out.println("College Location: " + college.getLocation());

          // Создание объекта университета
          University university = new University();
          university.setName("State University");
          university.setLocation("Suburban Area");
          university.setAccreditation("Regional Accreditation");

          // Вывод информации о университете
          System.out.println("\nUniversity Name: " + university.getName());
          System.out.println("University Location: " +
  university.getLocation());
          System.out.println("University Accreditation: " +
  university.getAccreditation());
      }
  }
```

Результат программы:

```
College Name: Community College
College Location: City Center

University Name: State University
University Location: Suburban Area
University Accreditation: Regional Accreditation
```

Задание 2. Создать суперкласс Транспортное средство и подклассы Автомобиль, Велосипед, Повозка. Подсчитать время и стоимость перевозки пассажиров и грузов каждым транспортным средством.

Код программы:

Bicycle.java

```
package task02;

public class Bicycle extends TransportVehicle {
    private double averageSpeed;

    public Bicycle(String name, int passengersCapacity, int cargoCapacity,
double averageSpeed) {
        super(name, passengersCapacity, cargoCapacity);
        this.averageSpeed = averageSpeed;
    }

    @Override
    public double calculateTime(double distance) {
        return distance / averageSpeed;
    }

    @Override
    public double calculateCost(double distance, double cargoWeight) {
        // Велосипед не требует дополнительных расходов на топливо или
обслуживание
        return 0;
    }
}
```

Car.java

```
package task02;

public class Car extends TransportVehicle {
    private double speed;
    private double costPerKm;

    public Car(String name, int passengersCapacity, int cargoCapacity, double
speed, double costPerKm) {
        super(name, passengersCapacity, cargoCapacity);
        this.speed = speed;
        this.costPerKm = costPerKm;
    }

    @Override
    public double calculateTime(double distance) {
        return distance / speed;
    }

    @Override
    public double calculateCost(double distance, double cargoWeight) {
        return distance * costPerKm + cargoWeight * 0.1;
    }
}
```

Carriage.java

```
package task02;

public class Carriage extends TransportVehicle {
    private int horsePower;

    public Carriage(String name, int passengersCapacity, int cargoCapacity,
```

```
int horsePower) {
        super(name, passengersCapacity, cargoCapacity);
        this.horsePower = horsePower;
    }

    @Override
    public double calculateTime(double distance) {
        return distance / (horsePower * 0.1);
    }

    @Override
    public double calculateCost(double distance, double cargoWeight) {
        return 0;
    }
}
```

TransportVehicle.java

```
package task02;

public abstract class TransportVehicle {
    protected String name;
    protected int passengersCapacity;
    protected int cargoCapacity;

    public TransportVehicle(String name, int passengersCapacity, int
cargoCapacity) {
        this.name = name;
        this.passengersCapacity = passengersCapacity;
        this.cargoCapacity = cargoCapacity;
    }

    public abstract double calculateTime(double distance);

    public abstract double calculateCost(double distance, double
cargoWeight);
}
```

Main.java

```
package task02;

public class Main {
    public static void main(String[] args) {
        TransportVehicle[] vehicles = new TransportVehicle[3];

        vehicles[0] = new Car("Toyota Camry", 5, 500, 60, 5); // название,
вместимость, грузоподъемность, скорость, стоимость за км
        vehicles[1] = new Bicycle("Mountain Bike", 1, 50, 20); // название,
вместимость, грузоподъемность, средняя скорость
        vehicles[2] = new Carriage("Horse Carriage", 2, 200, 10); //
название, вместимость, грузоподъемность, мощность лошади

        double distance = 100; // расстояние в км
        double cargoWeight = 50; // вес груза в кг

        for (TransportVehicle vehicle : vehicles) {
            double time = vehicle.calculateTime(distance);
            double cost = vehicle.calculateCost(distance, cargoWeight);

            System.out.println(vehicle.name);
```

```
            System.out.println("Time: " + time + " hours");
            System.out.println("Cost: $" + cost);
            System.out.println();
        }
    }
}
```

Результат программы:

```
Toyota Camry
Time: 1.6666666666666667 hours
Cost: $505.0

Mountain Bike
Time: 5.0 hours
Cost: $0.0

Horse Carriage
Time: 100.0 hours
Cost: $0.0
```

Задание 3. В задании 3 ЛР No4, где возможно, заменить объявления суперклассов объявлениями абстрактных классов или интерфейсов.

Поменяется лишь класс User.java на абстрактный.

Работа программы: идентична заданию 2

User.java

```
package task03;

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

  public abstract class User {
      protected String phoneNumber;
      protected List<Service> services;

      public User(String phoneNumber) {
          this.phoneNumber = phoneNumber;
          this.services = new ArrayList<>();
      }

      public String getPhoneNumber() {
          return phoneNumber;
      }

      public void changePhoneNumber(String newNumber) {
```

```
        this.phoneNumber = newNumber;
    }

    public void addService(Service service) {
        services.add(service);
    }

    public void removeService(Service service) {
        services.remove(service);
    }
}
```

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