

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

ОТЧЁТ
по лабораторной работе №5

Выполнил:

Студент 3 курса
группы ПО-9
Кучко Ярослав Валерьевич

Проверил:

Крощенко А. А.

Брест 2024

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

Вариант 10

Задание 1. Реализовать абстрактные классы или интерфейсы, а также наследование и полиморфизм для следующих классов:

interface Файл ← abstract class Директория ← class Диск.

Код программы.

```
public interface File {

    void open();
    void delete();

}

public abstract class Directory implements File {
    protected String path;

    public String getPath() {
        return path;
    }

    public void setPath(String path) {
        this.path = path;
    }

    @Override
    public void open() {
        System.out.println("Directory " + path + " opened.");
    }

    @Override
    public void delete() {
        System.out.println("Directory " + path + " deleted.");
    }
}

public class Disk extends Directory {

    @Override
    public void open() {
        System.out.println("Disk " + path + " opened.");
    }

    @Override
    public void delete() {
        return;
    }
}
```

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.

Создать суперкласс Учащийся и подклассы Школьник и Студент. Создать массив объектов суперкласса и заполнить этот массив объектами. Показать отдельно студентов и школьников.

Код программы.

```
public class Task2 {

    public static void main(String[] args) {
        List<Pupil> pupils = new ArrayList<>();

        pupils.add(new Schoolkid("Mark", 15, "Velikorita School"));
        pupils.add(new Student("Yaroslav", 19, "BrSTU"));
        pupils.add(new Schoolkid("Dima", 16, "Gymnasium #1"));
        pupils.add(new Student("Dasha", 20, "BrSTU"));
        pupils.add(new Student("Kostya", 21, "BrSTU"));

        pupils.stream().filter(obj -> obj instanceof
Student).forEach(System.out::println);
        pupils.stream().filter(obj -> obj instanceof
Schoolkid).forEach(System.out::println);
    }
}

public abstract class Pupil {
    protected String name;
    protected int age;
    protected String school;

    public abstract void goToSchool();
    public abstract void doHomeWork();

    public Pupil(String name, int age, String school) {
        this.name = name;
        this.age = age;
        this.school = school;
    }
}

public class Schoolkid extends Pupil {
    public Schoolkid(String name, int age, String school) {
        super(name, age, school);
    }

    @Override
    public void goToSchool() {
        System.out.println("Schoolkid " + this.name + " went to " +
this.school);
    }

    @Override
    public void doHomeWork() {
        System.out.println("Schoolkid " + this.name + " did homework.");
    }

    @Override
    public String toString() {
        return "Schoolkid{" +
            "name='" + name + '\'' +
            ", age='" + age + '\'' +
            ", school='" + school + '\'' +
            '}';
    }
}

public class Student extends Pupil {

    public Student(String name, int age, String school) {
```

```

        super(name, age, school);
    }

    @Override
    public void goToSchool() {
        System.out.println("Student " + this.name + " went to " +
this.school);
    }

    @Override
    public void doHomeWork() {
        System.out.println("Student " + this.name + " did homework.");
    }

    @Override
    public String toString() {
        return "Student{" +
            "name='" + name + '\'' +
            ", age='" + age + '\'' +
            ", school='" + school + '\'' +
            '}';
    }
}

```

Результат работы:

```

Student{name='Yaroslav', age='19', school='BrSTU'}
Student{name='Dasha', age='20', school='BrSTU'}
Student{name='Kostya', age='21', school='BrSTU'}
Schoolkid{name='Dima', age='16', school='Gymnasium #1'}
Schoolkid{name='Mark', age='15', school=' Velikorita School'}

```

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

Вывод: в ходе выполнения лабораторной работы приобретены практические навыки в области объектно-ориентированного проектирования.