

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
“БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ”
ФАКУЛЬТЕТ ЭЛЕКТРОННО-ИНФОРМАЦИОННЫХ СИСТЕМ
КАФЕДРА ИНТЕЛЛЕКТУАЛЬНЫХ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

Отчет по лабораторной работе №5

Специальность ПО-9

Выполнил
А. С. Мисюк,
студент группы ПО-9
Проверил
А. А. Крощенко,
ст. преп. кафедры ИИТ,
«__» _____ 2024 г.

Брест 2024

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

Вариант № 2

Задание 1. Реализовать абстрактные классы или интерфейсы, а также наследование и полиморфизм для следующих классов: interface Abiturient ← abstract class Student ← class Student Of Faculty.

Код программы:

```
package com.labrab5.task1;

/**
 * Вариант №2
 * Реализовать абстрактные классы или интерфейсы, а также наследование и полиморфизм для
 * следующих классов:
 * interface Abiturient ← abstract class Student ← class Student Of Faculty.
 */
public class Main
{
    public static void main(String[] args)
    {
        StudentOfFaculty student1 = new StudentOfFaculty("Student", 18, "Faculty");

        Abiturient[] abiturients = {student1};
        Student[] students = {student1};

        for (Abiturient abiturient : abiturients)
        {
            abiturient.applyForAdmission();
        }

        for (Student student : students)
        {
            student.study();
            student.takeExams();
        }
    }
}

package com.labrab5.task1;

public interface Abiturient
{
    void applyForAdmission();
}

package com.labrab5.task1;

@SuppressWarnings({"FieldCanBeLocal", "FieldMayBeFinal"})
public abstract class Student implements Abiturient
{
    private String name;
    private int age;

    public Student(String name, int age)
    {
        this.name = name;
        this.age = age;
    }
}
```

```

    public void study()
    {
        System.out.println(name + " (" + age + ") is studying...");
    }

    public abstract void takeExams();
}

package com.labrab5.task1;

@SuppressWarnings("FieldMayBeFinal")
public class StudentOfFaculty extends Student
{
    private String faculty;

    public StudentOfFaculty(String name, int age, String faculty)
    {
        super(name, age);
        this.faculty = faculty;
    }

    @Override
    public void applyForAdmission()
    {
        System.out.println("Applying for admission to " + faculty + " faculty");
    }

    @Override
    public void takeExams()
    {
        System.out.println("Taking exams for " + faculty + " faculty");
    }
}

```

Пример

```

>java.exe com.labrab5.task1.Main
Applying for admission to Faculty faculty
Student (18) is studying...
Taking exams for Faculty faculty

```

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:

```

package com.labrab5.task2;

/**
 * Вариант №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.
 * <p>
 * Создать суперкласс Учащийся и подклассы Школьник и Студент. Создать массив объектов
 * суперкласса и заполнить этот массив объектами. Показать отдельно студентов и школьников.
 */
public class Main
{
    public static void main(String[] args)
    {
        Student[] students = new Student[3];
        students[0] = new SchoolStudent("John", 15, 10);
        students[1] = new CollegeStudent("Emma", 20, "Computer Science");
        students[2] = new SchoolStudent("Michael", 14, 9);

        System.out.println("Students study:");
        for (Student student : students)
        {
            student.study();
        }

        System.out.println("\nSchool Students:");
        for (Student student : students)
        {
            if (student instanceof SchoolStudent schoolStudent)
            {
                System.out.println("Name: " + schoolStudent.getName()
                    + ", Age: " + schoolStudent.getAge()
                    + ", Grade: " + schoolStudent.getGrade()
                );
            }
        }

        System.out.println("\nCollege Students:");
        for (Student student : students)
        {
            if (student instanceof CollegeStudent collegeStudent)
            {
                System.out.println("Name: " + collegeStudent.getName()
                    + ", Age: " + collegeStudent.getAge()
                    + ", Major: " + collegeStudent.getMajor()
                );
            }
        }
    }
}

package com.labrab5.task2;

@SuppressWarnings("FieldMayBeFinal")
public abstract class Student
{
    private String name;
    private int age;

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

```

```

        this.age = age;
    }

    public abstract void study();

    public String getName()
    {
        return name;
    }

    public int getAge()
    {
        return age;
    }
}

package com.labrab5.task2;

@SuppressWarnings("FieldMayBeFinal")
public class SchoolStudent extends Student
{
    private int grade;

    public SchoolStudent(String name, int age, int grade)
    {
        super(name, age);
        this.grade = grade;
    }

    @Override
    public void study()
    {
        System.out.println("School student " + getName() + " (" + getAge() + ") is studying");
    }

    public int getGrade()
    {
        return grade;
    }
}

package com.labrab5.task2;

@SuppressWarnings("FieldMayBeFinal")
public class CollegeStudent extends Student
{
    private String major;

    public CollegeStudent(String name, int age, String major)
    {
        super(name, age);
        this.major = major;
    }

    @Override
    public void study()
    {
        System.out.println("College student " + getName() + " (" + getAge() + ") is studying");
    }

    public String getMajor()
    {
        return major;
    }
}

```

Пример

```
>java.exe com.labrab5.task2.Main
Students study:
School student John (15) is studying
College student Emma (20) is studying
School student Michael (14) is studying

School Students:
Name: John, Age: 15, Grade: 10
Name: Michael, Age: 14, Grade: 9

College Students:
Name: Emma, Age: 20, Major: Computer Science
```

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

Р. С. Ввиду отсутствия суперклассов, был выделен суперкласс Human из Client, а Administrator расширен суперклассом Human.

Код программы (измененный):

```
package com.labrab5.task3;

/**
 * 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.
 * <p>
 * Система Платежи.
 * Клиент имеет Счет в банке и Кредитную Карту (КК).
 * Клиент может оплатить Заказ, сделать платеж на другой Счет,
 * заблокировать КК и аннулировать Счет.
 * Администратор может заблокировать КК за превышение кредита.
 */
public class Task3
{
    public static void exec()
    {
        Client client = new Client("Иванов", 30, "ул. Пушкина, д.10");

        Account account = new Account("1234567890", 1000.0);
        client.setAccount(account);

        CreditCard creditCard = new CreditCard("9876543210", account, 5000.0);
        client.setCreditCard(creditCard);

        Order order = new Order("Z123", 500.0);

        client.payOrder(order);
        client.payOrder(order);

        Account otherAccount = new Account("0987654321", 2000.0);
        client.makePayment(otherAccount, 300.0);
        client.makePayment(otherAccount, 300.0);
    }
}
```

```

        Administrator administrator = new Administrator("Сергей", 28, "ул. Пушкина, д.16");
        administrator.blockCreditCard(creditCard);

        client.payOrder(order);

        client.blockCreditCard();

        client.cancelAccount();
        client.payOrder(order);
    }
}

package com.labrab5.task3;

public class Human
{
    protected String name;
    protected int age;
    protected String address;

    public Human(String name, int age, String address)
    {
        this.name = name;
        this.age = age;
        this.address = address;
    }
}

package com.labrab5.task3;

public class Administrator extends Human
{
    public Administrator(String name, int age, String address)
    {
        super(name, age, address);
    }

    public void blockCreditCard(CreditCard card)
    {
        card.setBlocked(true);
        System.out.println("Блокировка кредитной карты " + card.getCardNumber() + " администратором "
+ name);
    }
}

package com.labrab5.task3;

/**
 * Клиент имеет Счет в Банке и Кредитную Карту (КК).
 * Клиент может оплатить Заказ,
 * сделать платеж на другой Счет,
 * заблокировать КК
 * и аннулировать Счет.
 */
@SuppressWarnings({"FieldMayBeFinal"})
public class Client extends Human
{
    private Account account;
    private CreditCard creditCard;

    public Client(String name, int age, String address)
    {
        super(name, age, address);
    }
    //.....
}

```

Пример

```
>java.exe com.labrab5.task3.Main
Оплата заказа Z123 клиентом Иванов
Заказ №Z123 уже оплачен
Выполнение платежа на счет 0987654321 в размере 300.0
Нехватает баланса
Блокировка кредитной карты 9876543210 администратором Сергей
Карточка клиента Иванов заблокирована!
Блокировка кредитной карты клиента Иванов
Аннулирование счета клиента Иванов
У клиента Иванов нет счета
```

Рисунки с результатами работы программы

```
C:\Users\misij\Documents\SPP\labrab5\out\production\labrab5 com.labrab5.task1.Main
Applying for admission to Faculty faculty
Student (18) is studying...
Taking exams for Faculty faculty
```

```
C:\Users\misij\Documents\SPP\labrab5\out\production\labrab5 com.labrab5.task2.Main
Students study:
School student John (15) is studying
College student Emma (20) is studying
School student Michael (14) is studying

School Students:
Name: John, Age: 15, Grade: 10
Name: Michael, Age: 14, Grade: 9

College Students:
Name: Emma, Age: 20, Major: Computer Science
```

```
C:\Users\misij\Documents\SPP\labrab5\out\production\labrab5 com.labrab5.task3.Main
Оплата заказа Z123 клиентом Иванов
Заказ №Z123 уже оплачен
Выполнение платежа на счет 0987654321 в размере 300.0
Нехватает баланса
Блокировка кредитной карты 9876543210 администратором Сергей
Карточка клиента Иванов заблокирована!
Блокировка кредитной карты клиента Иванов
Аннулирование счета клиента Иванов
У клиента Иванов нет счета
```


Вывод: исследовал создание классов и объектно-ориентированное программирование на языке программирования Java, приобрел практические навыки в области объектно-ориентированного проектирования в использовании суперклассов и интерфейсов.