МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ЮРИДИЧНА АКАДЕМІЯ»

Протокол лабораторної роботи №5

з дисципліни об'єктно-орієнтоване програмування

на тему: «Поліморфізм (пізнє зв'язування). Висхідне перетворення»

Виконала студентка групи

ІПЗ-212

Ярош Є.В.

Прийняв

Рудніченко М.Д.

Одеса, 2022

ЗМІСТ

[ВСТУП](#_30j0zll) 3

[ХІД РОБОТИ](#_8fy8djbnwuy9) 4

[ВИСНОВОК](#_2et92p0) 9

# ВСТУП

Мета роботи - ознайомитися з механізмом поліморфізму в ООП, розібратися зі статичним та динамічним зв'язуванням, навчитися використовувати висхідне перетворення.

# ХІД РОБОТИ

1. Створіть клас Person, який повинен містити такі поля та гетери/сетери для цих полів:

Прізвище; Ім'я; Вік.

Створіть метод printInfo(), який повертав би рядок наступного формату: Людина <прізвище> <ім'я>, вік: <вік>

2. Створіть клас Student, який має успадковуватись від класу Person. Додайте додаткові поля і гетери\сеттери для цих полів:

Група; Номер студентського квитка.

Перевизначте метод printInfo(), який має виводити рядок наступного формату:

Студент групи <група> <прізвище> <ім'я>, вік: <вік>. Номер студентського квитка: <номер>

3. Створіть клас Lecturer, який має успадковуватись від класу Person. Додайте додаткові поля і гетери\сетери для цих полів:

Кафедра; Зарплатня.

Перевизначте метод printInfo(), який має виводити рядок наступного формату:

Викладач кафедри <кафедра> <прізвище> <ім'я>, вік: <вік>. Зарплата: <зарплата>

4. Використовуючи висхідне перетворення, створіть у класі Main кілька об'єктів класів Student і Lecturer, після чого створіть масив, який міг би включати об'єкти класів Person, Student, Lecturer. Заповніть масив об'єктами цих класів.

5. Використовуючи цикл, зверніться до елементів масиву та виведіть у консоль, за допомогою методу printInfo(), інформацію від кожного об'єкта.

Код програми:

import java.util.Scanner;

class Person {

private String surname;

private String name;

private int age;

public void setSurname(String surname) {

this.surname = surname;

};

public void setName(String name) {

this.name = name;

};

public void setAge(int age) {

this.age = age;

};

public String getSurname() {

return surname;

};

public String getName() {

return name;

};

public int getAge() {

return age;

};

public void printInfo() {

System.*out*.println("Person " + getSurname() + " " + getName() + ", age: " + getAge());

};

};

class Student extends Person {

private int group;

private int cardNumber;

public void setGroup(int group) {

this.group = group;

};

public void setCardNumber(int cardNumber) {

this.cardNumber = cardNumber;

};

public int getGroup() {

return group;

};

public int getCardNumber() {

return cardNumber;

};

@Override

public void printInfo() {

System.*out*.println("Student of " + getGroup() + " group " + getSurname() + " " + getName() +

", age: " + getAge() + ". \nStudent's card number: " + getCardNumber());

};

};

class Lecturer extends Person {

private String chair;

private int salary;

public void setChair(String chair) {

this.chair = chair;

};

public void setSalary(int salary) {

this.salary = salary;

};

public float getSalary() {

return salary;

};

public String getChair() {

return chair;

};

@Override

public void printInfo() {

System.*out*.println("Lecturer of " + getChair() + " chair " + getSurname() + " " + getName() +

", age: " + getAge() + ". \nSalary: " + getSalary());

};

};

public class Main {

public static void main(String [] args) {

System.*out*.println("Enter info about person:");

Scanner in = new Scanner(System.*in*);

Person[] people = new Person[9];

System.*out*.println("Students:");

for (int i = 0; i < 3; i++) {

Person student = new Student();

System.*out*.println("Enter surname: ");

student.setSurname(in.next());

System.*out*.println("Enter name: ");

student.setName(in.next());

System.*out*.println("Enter age: ");

student.setAge(in.nextInt());

System.*out*.println("Enter group: ");

Student student1 = (Student) student;

student1.setGroup(in.nextInt());

System.*out*.println("Enter Student's card number: ");

student1.setCardNumber(in.nextInt());

people[i] = student;

};

System.*out*.println("Lecturers:");

for (int i = 3; i < 6; i++) {

Person lecturer = new Lecturer();

System.*out*.println("Enter surname: ");

lecturer.setSurname(in.next());

System.*out*.println("Enter name: ");

lecturer.setName(in.next());

System.*out*.println("Enter age: ");

lecturer.setAge(in.nextInt());

System.*out*.println("Enter chair: ");

Lecturer lecturer1 = (Lecturer) lecturer;

lecturer1.setChair((in.next() + " " + in.next()));

System.*out*.println("Enter salary: ");

lecturer1.setSalary(in.nextInt());

people[i] = lecturer;

};

System.*out*.println("Persons:");

for (int i = 6; i < 9; i++) {

Person person = new Person();

System.*out*.println("Enter surname: ");

person.setSurname(in.next());

System.*out*.println("Enter name: ");

person.setName(in.next());

System.*out*.println("Enter age: ");

person.setAge(in.nextInt());

people[i] = person;

};

in.close();

for (int i = 0; i < 9;i++) {

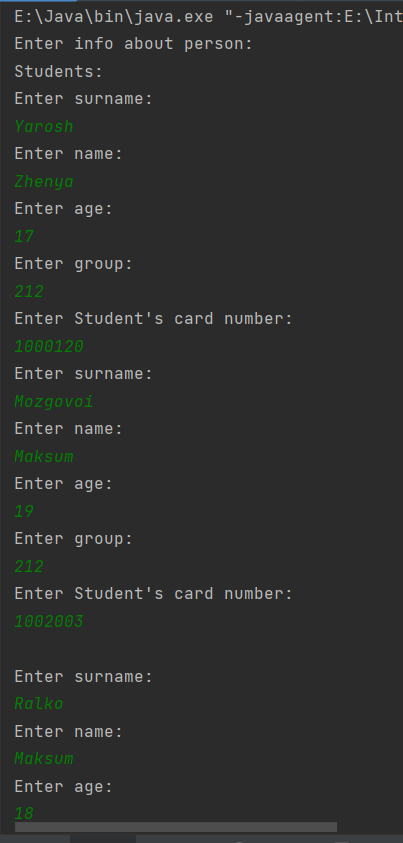
people[i].printInfo();

};

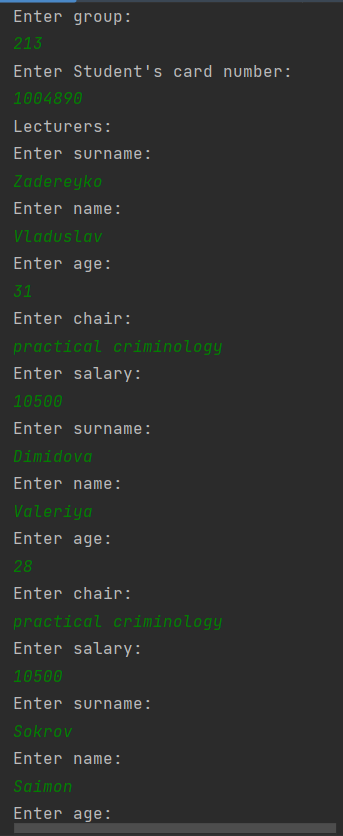
};

}

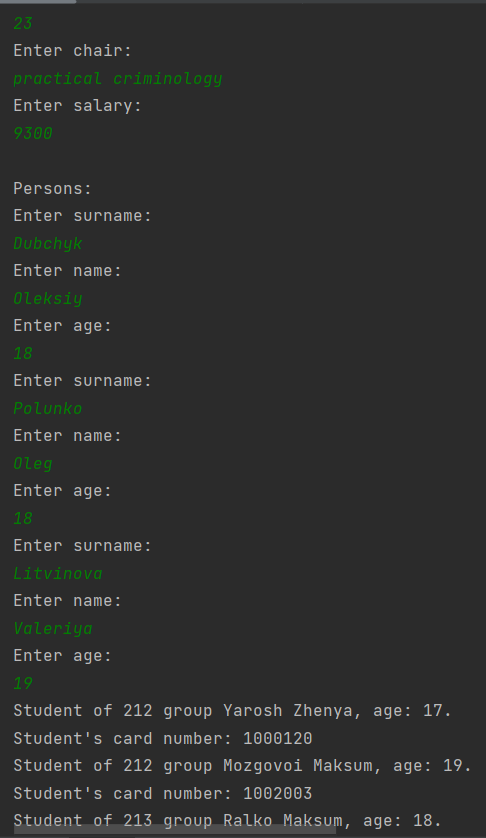
Результат роботи програми наведено на рис.1.-рис.4.



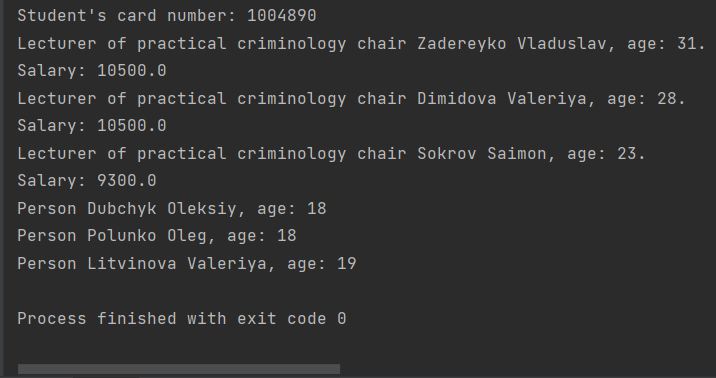
Риcунок 1 – Результат роботи програми.



Риcунок 2 – Результат роботи програми.



Риcунок 3 – Результат роботи програми.



Риcунок 4 – Результат роботи програми.

# ВИСНОВОК

Протягом виконання лабораторної роботи, я ознайомилася з механізмом поліморфізму в ООП, розібралася зі статичним та динамічним зв'язуванням, навчилася використовувати висхідне перетворення.