**Лабораторна робота №3**

**Тема: «**Парадигма наслідування в Java**»**

**Мета: «**Опанувати реалізацію наслідування та перевантаження і перевизначення методів в Java**»**

**Теоретична робота**

Одним з ключових аспектів об'єктно-орієнтованого програмування є успадкування. За допомогою спадкування можна розширити функціонал вже наявних класів за рахунок додавання нового функціоналу або зміни старого в класах нащадках. При цьому батьківський клас називається ще суперкласом. Доступ до полів та методів суперкласу можна організувати в класі нащадку за допомогою службового слова super.

Похідний клас має доступ до всіх методів і полях базового класу (навіть якщо базовий клас знаходиться в іншому пакеті) крім тих, які позначені модифікатором private. Модифікатора final забороняє наслідування для елементу класу, перед яким він встановлений.

**Наприклад, є такий клас Person, що описує окрему людину:**

public class Person {

private String name;

private String surname;

public String getName() { return name; }

public String getSurname() { return surname; }

public Person(String name, String surname){

this.name=name;

this.surname=surname;

}

public void displayInfo(){

System.out.println("Ім’я: " + name + " Фамілія: " + surname);

}

}

Потім ми вирішили розширити наявну систему класів, додавши в неї клас, що описує співробітника підприємства - клас Employee. Так як цей клас реалізує той же функціонал, що і клас Person, так як співробітник - це також і людина, то було б раціонально зробити клас Employee похідним (або спадкоємцем) від класу Person, який, в свою чергу, називається базовим класом або батьком :

class Employee extends Person{

}

Щоб оголосити один клас спадкоємцем від іншого, треба використовувати після імені класу-спадкоємця ключове слово extends, після якого йде ім'я базового класу. Для класу Employee базовим є Person, і тому клас Employee успадковує всі ті ж поля і методи, які є в класі Person.

У класі Employee можуть бути визначені свої методи і поля, а також конструктор. Здатність до зміни функціональності, успадкованої від базового класу, званим, поліморфізмом і є одіним з ключових аспектів об'єктно-орієнтованого програмування поряд зі спадкуванням та інкапсуляцією.

**Наприклад, перевизначимо метод displayInfo () класу Person в класі Employee:**

class Employee extends Person{

      private String company;

      public Employee(String name, String surname, String company) {

         super(name, surname);

         this.company=company;

    }

    public void displayInfo(){

          super.displayInfo();

       System.out.println("Компанія: " + company);

    }

}

Клас Employee визначає додаткове поле для зберігання компанії, в якій працює співробітник. Крім того, воно також встановлюється в конструкторі.

Так як поля name і surname в базовому класі Person оголошені з модифікатором private, то ми не можемо до них безпосередньо звернутися з класу Employee. Однак в даному випадку нам це не потрібно. Щоб їх встановити, ми звертаємося до конструктору базового класу за допомогою ключового слова super, в дужках після якого йде перерахування переданих аргументів.

За допомогою ключового слова super ми можемо звернутися до будь-якого члену базового класу - методу або полю, якщо вони не визначені з модифікатором private.

Використовуючи звернення до методом базового класу, можна було б перевизначити метод displayInfo () наступним чином:

public void displayInfo(){

         System.out.println("Имя: " + super.getName() + " Фамилия: "

        + super.getSurname() + " Компания: " + company);

}

При цьому нам необов'язково перевизначати всі методи базового класу. Наприклад, в даному випадку ми не перевизначаємо методи getName () і getSurname (). Тому для цих методів клас-спадкоємець буде використовувати реалізацію з базового класу. І в основній програмі ми можемо ці методи використовувати:

public static void main(String[] args) {

     Employee empl = new Employee("Tom", "Simpson", "Oracle");

     empl.displayInfo();

     String firstName = empl.getName();

 System.out.println(firstName);

}

**Завдання до виконання:**

1. Реалізувати в головному методі класу ввод значень змінних (клас Scanner).
2. Реалізувати операцію конкатенації та вивід на екран рядка для змінних (name, surname, company).
3. Реалізувати програму згідно індивідуального завдання.

У всіх завданнях реалізувати висновок на екран, методи отримання значень полів і методи установки значень полів, а також необхідні конструктори (якщо це не зазначено в завданні явно). Перетворення в рядок реалізувати у вигляді функції перетворення string.

Для демонстрації роботи з об'єктами нового типу у всіх завданнях потрібно написати головну функцію. У програмі повинні бути присутніми різні способи створення об'єктів. Програма повинна демонструвати використання всіх методів.

**Варіанти завдань**

1. Створити базовий клас Саr (машина), що характеризується торговою маркою (рядок), числом циліндрів, потужністю. Визначити методи перепризначення і зміни потужності. Створити похідний клас Lorry (вантажівка), що характеризується також вантажопідйомністю кузова. Визначити функції перепризначення марки і зміни вантажопідйомності.

2. Створити клас Pair (пара чисел); визначити методи зміни полів і порівняння пар: пара p1 більше пари р2, якщо (pl.first> р2.first) або (pl.first = р2.first) і (pl.second> р2.second). Визначити клас-спадкоємець Fraction з полями: ціла частина числа і дрібна частина числа. Визначити повний набір методів порівняння.

3. Створити клас Liquid (рідина), який має поля назви і щільності. Визначити методи перепризначення і зміни щільності. Створити похідний клас Alcohol (спирт), який має міцності. Визначити методи перепризначення і зміни міцності.

4. Створити клас Pair (пара чисел); визначити методи зміни полів і обчислення добутку чисел. Визначити похідний клас Rectangle (прямокутник) з полями-сторонами. Визначити методи обчислення периметра і площі прямокутника.

5. Створити клас Man (чоловік), з полями: ім'я, вік, стать і вагу. Визначити методи перепризначення імені, настання певного віку і зміни ваги. Створити похідний клас Student, що має поле року навчання. Визначити методи перепризначення і збільшення року навчання.

6. Створити клас Triad (трійка чисел); визначити методи зміни полів і обчислення суми чисел. Визначити похідний клас Triangle з полями-сторонами. Визначити методи обчислення кутів і площі трикутника.

7. Створити клас Triangle з полями-сторонами. Визначити методи зміни сторін, обчислення кутів, обчислення периметра. Створити похідний клас Equilateral (рівносторонній), що має поле площі. Визначити метод обчислення площі.

8. Створити клас Triangle з полями-сторонами. Визначити методи зміни сторін, обчислення кутів, обчислення периметра. Створити похідний клас RightAngled (прямокутний), який має поле площі. Перевизначити або перевантажити методи для обчислення площ даних геометричних фігур.

9. Створити клас Pair (пара чисел); визначити методи зміни полів і обчислення добутку чисел. Визначити похідний клас RightAngled з полями-катетами. Визначити методи обчислення гіпотенузи і площі трикутника.

10. Створити клас Triad (трійка чисел); визначити метод порівняння тріад (див. завдання 2). Визначити похідний клас Date з полями: рік, місяць і день. Визначити повний набір методів порівняння дат.

11. Створити клас Triad (трійка чисел); визначити метод порівняння тріад – трійок чисел (див. завдання 2). Визначити похідний клас Time з полями: година, хвилина і секунда. Визначити повний набір методів порівняння моментів часу.

12. Реалізувати клас-оболонку Number для числового типу float. Реалізувати методи складання і ділення. Створити похідний клас Real, в якому реалізувати метод зведення в довільну ступінь, і метод для обчислення логарифма числа.

13. Створити клас Triad (трійка чисел); визначити методи збільшення полів на 1. Визначити похідний клас Date з полями: рік, місяць і день. Перевизначити методи збільшення полів на 1 і визначити метод збільшення дати на n днів.

14. Реалізувати клас-оболонку Number для числового типу double. Реалізувати методи множення і віднімання. Створити похідний клас Real, в якому реалізувати метод, що обчислює корінь довільного ступеня, і метод для обчислення числа даного ступеню.

15. Створити клас Triad (трійка чисел); визначити методи збільшення полів на 1. Визначити клас-спадкоємець Time з полями: година, хвилина, секунда. Перевизначити методи збільшення полів на 1 і визначити методи збільшення на n секунд і хвилин.

16. Створити базовий клас Pair (пара цілих чисел) з операціями перевірки на рівність і перемноження полів. Реалізувати операцію віднімання пар по формулі (а, b) - (с, d) = (а - b, с - d). Створити похідний клас Rational; визначити нові операції додавання (а, b) + (с, d) = (a\*d + b\*c, b\*d) і ділення (a, b) / (c, d) = (a\*d, b\*c); перевизначити операцію віднімання (а, b) - (с, d) = (a\*d – b\*c, b\*d).

17. Створити клас Pair (пара чисел); визначити метод перемноження полів і операцію складання пар (а, b) + (с, d) = (а + b, с + d). Визначити похідний клас Complex з полями: дійсна і уявна частини числа. Визначити методи множення (а, b) х (с, d) = (а\*с – b\*d, a\*d + b\*c) і віднімання (а, b) - (с, d) = (а - b, с - d).

18. Створити клас Pair (пара цілих чисел); визначити методи зміни полів і операцію складання пар (а, b) + (с, d) = (а + b, з + d). Визначити клас-спадкоємець Long з полями: старша частина числа і молодша частина числа. Перевизначити операцію складання і визначити методи множення і віднімання.

19. Створити базовий клас Triad (трійка чисел) з операціями додавання з числом, множення на число, перевірки на рівність. Створити похідний клас vector3D, що задається трійкою координат. Повинні бути реалізовані: операція додавання векторів, скалярний добуток векторів.

20. Створити клас Pair (пара цілих чисел); визначити метод множення на число і операцію додавання пар чисел (а, b) + (с, d) = (а + b, з + d). Визначити клас-спадкоємець Money з нулями: гривні і копійки. Перевизначити операцію складання і визначити методи вирахування і розподілу грошових сум.