Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Інститут комп’ютерних наук та інформаційних технологій

Кафедра автоматизованих систем управління



**Звіт**

до виконаної лабораторної роботи № 2

з дисципліни

“Прикладне програмування”

на тему:

***«Робота з класами»***

Виконав

студент групи *ОІ-11 сп*

*Вальчевський П. В.*

Прийняв:

*Пелех Ю. М.*

Львів – 2023

## Лабораторна робота № 2

*Тема роботи:* «Робота з класами».

*Мета роботи:* ***…***

***Порядок виконання роботи***

1. **Послідовність роботи**
2. Створити проект, що складається з двох класів: основного (Main) та класу для представлення об’єкта відповідно специфікації, що наведена у варіанті індивідуального завдання. Кожний клас повинен бути розміщений у окремому пакеті. У створеному класі визначити приватні поля для зберігання указаних даних, конструктори для створення об’єктів та відкриті методи setValue(), getValue(), toString() для доступу до полів об’єкту.
3. В основному класі програми визначити методи, що створюють масив об'єктів. Задати критерії вибору даних та вивести ці дані на консоль ( використати метод toString() ). Для кожного критерію створити окремий метод.
4. Виконати програму, та пересвідчитись, що дані зберігаються та коректно виводяться на екран відповідно до вказаних критеріїв.
5. **Індивідуальне завдання**

Номер завдання вибирався за порядковим номером у журналі групи. Мій варіант № 3.

1. House: id, Номер квартири, Площа, Поверх, Кількість кімнат, Вулиця.

Скласти масив об'єктів. Вивести:

* 1. список квартир, які мають задане число кімнат;
  2. список квартир, які мають задане число кімнат та розташовані на поверсі, який знаходиться в заданому проміжку;
  3. список квартир, які мають площу, що перевищує задану.

1. **Текст програми**

Посилання на репозиторій в GitHub: <https://github.com/ManFromLviv/nulp_prikladne_programuvannia_lr2.git>

*Файл House.java (клас специфікації) у пакеті lr2\_house*

/\*

\* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license

\* Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template

\*/

package lr2\_house;

/\*\*

\*

\* @author Valchevskyi

\*/

public class House {

private static int counter = -1; // Counter objects for identification.

private final int id;

private int number;

private double square;

private int level;

private int countRooms;

private String street;

public House(int number, double square, int level, int countRooms, String street) {

counter++;

this.id = counter;

this.number = number;

this.square = square;

this.level = level;

this.countRooms = countRooms;

this.street = street;

}

// Setters and getters.

public void setNumber(int number) {

this.number = number;

}

public void setSquare(double square) {

this.square = square;

}

public void setLevel(int level) {

this.level = level;

}

public void setCountRooms(int countRooms) {

this.countRooms = countRooms;

}

public void setStreet(String street) {

this.street = street;

}

public int getId() {

return id;

}

public int getNumber() {

return number;

}

public double getSquare() {

return square;

}

public int getLevel() {

return level;

}

public int getCountRooms() {

return countRooms;

}

public String getStreet() {

return new String(street);

}

@Override

public String toString() {

StringBuilder stringBuilder = new StringBuilder("Data house:\n");

stringBuilder.append("\tID: ").append(id).append("\n");

stringBuilder.append("\tNumber: ").append(number).append("\n");

stringBuilder.append("\tSquare: ").append(square).append("\n");

stringBuilder.append("\tLevel: ").append(level).append("\n");

stringBuilder.append("\tCount rooms: ").append(countRooms).append("\n");

stringBuilder.append("\tStreet: ").append(street);

return stringBuilder.toString();

}

}

*Файл Main.java (клас основний клас) у пакеті lr2*

/\*

\* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license

\* Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this template

\*/

package lr2;

import java.util.Random;

import lr2\_house.House;

/\*\*

\*

\* @author Valchevskyi

\*/

public class Main {

/\*\*

\* @param args the command line arguments

\*/

public static void main(String[] args) {

Main houses = new Main(5);

System.out.println(houses + "\n");

houses.taskA(2);

System.out.println();

houses.taskB(1, 4, 8);

System.out.println();

houses.taskC(50);

System.out.println();

}

// ===========================================================================

// ===========================================================================

// ===========================================================================

String[] streets = {"Shevchenka", "Chornovola", "Zernova", "Ivasiuka", "Franka"}; // Array streets (for generation array houses).

String line = "--------------------------------------------------------"; // Line for output information.

House[] houses;

private double roundDouble(double number) { // For round number to x,xx.

return Math.round(number \* 100) / 100.0;

}

public Main(int countHouses) {

houses = new House[countHouses];

Random random = new Random();

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

houses[i] = new House(i + 1, roundDouble(random.nextDouble(99) + 1), random.nextInt(8) + 1, random.nextInt(3) + 1, streets[random.nextInt(streets.length)]);

}

}

@Override

public String toString() {

StringBuilder stringBuilder = new StringBuilder("All houses:\n").append(line).append("\n");

for (House house : houses) {

stringBuilder.append(house).append("\n").append(line).append("\n");

}

stringBuilder.append("Count houses: ").append(houses.length).append("\n").append(line).append("\n");

return stringBuilder.toString();

}

public void taskA(int countRooms) {

int counter = 0;

System.out.println("Task A: List houses with count rooms: " + countRooms + "\n" + line);

for (House house : houses) {

if (house.getCountRooms() == countRooms) {

counter++;

System.out.println(house + "\n" + line);

}

}

outputCountHousesInTask(counter);

System.out.println(line + "\n");

}

public void taskB(int countRooms, int levelMin, int levelMax) {

int counter = 0;

System.out.println("Task B: List houses with count rooms: " + countRooms + " and level range: [" + levelMin + "; " + levelMax + "]\n" + line);

for (House house : houses) {

if (house.getCountRooms() == countRooms && house.getLevel() >= levelMin && house.getLevel() <= levelMax) {

counter++;

System.out.println(house + "\n" + line);

}

}

outputCountHousesInTask(counter);

System.out.println(line + "\n");

}

public void taskC(double square) {

int counter = 0;

System.out.println("Task C: List houses where square more: " + square + "\n" + line);

for (House house : houses) {

if (house.getSquare() > square) {

counter++;

System.out.println(house + "\n" + line);

}

}

outputCountHousesInTask(counter);

System.out.println(line + "\n");

}

private void outputCountHousesInTask(int counter) {

if (counter == 0) {

System.out.println("Not houses in this condition!");

} else {

System.out.println("Count houses: " + counter);

}

}

}

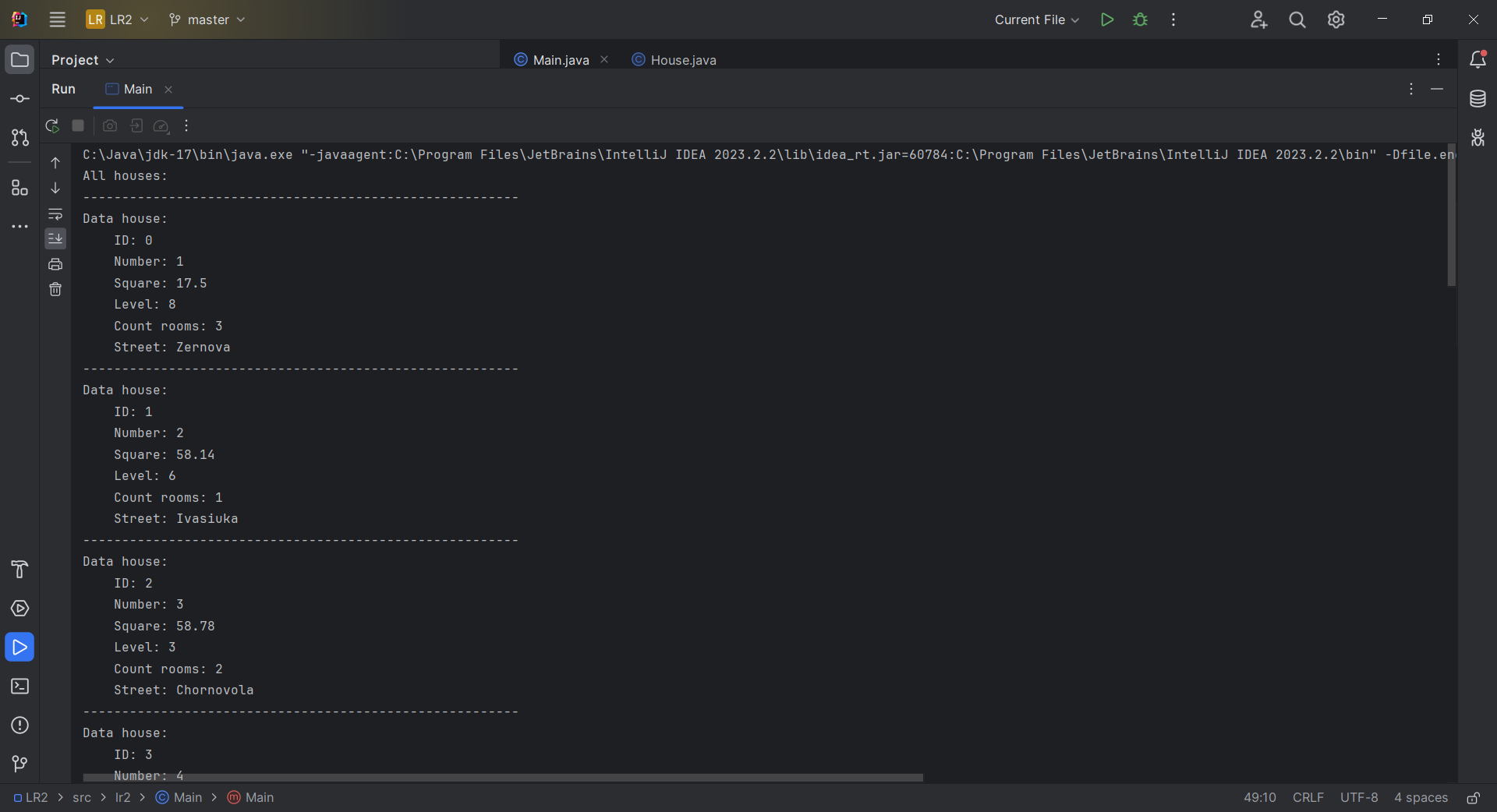
1. **Контрольний приклад та результати роботи програми**
2. 

Рисунок Результат програми виконання

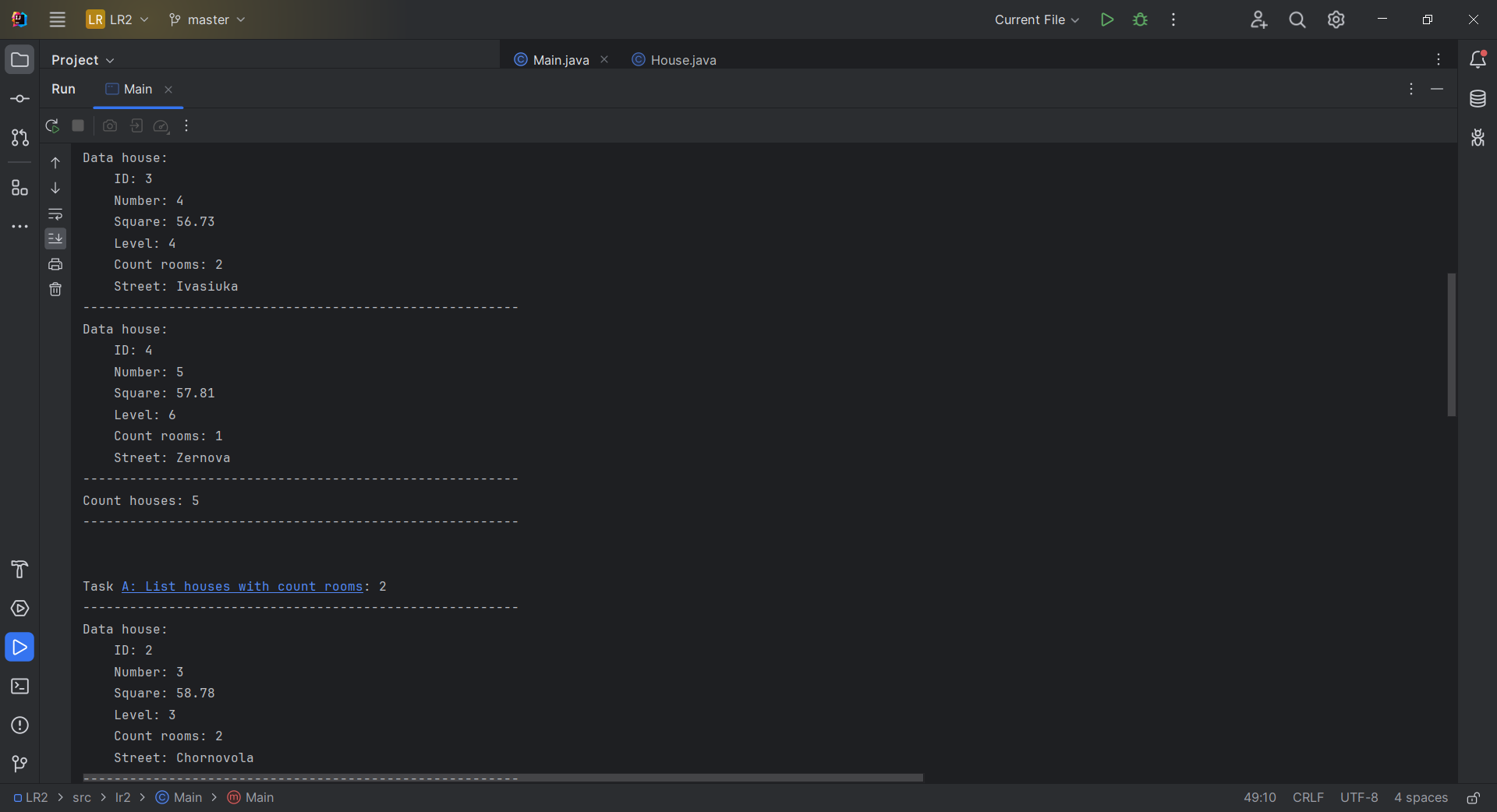


Рисунок 2 Результат програми виконання (продовження)

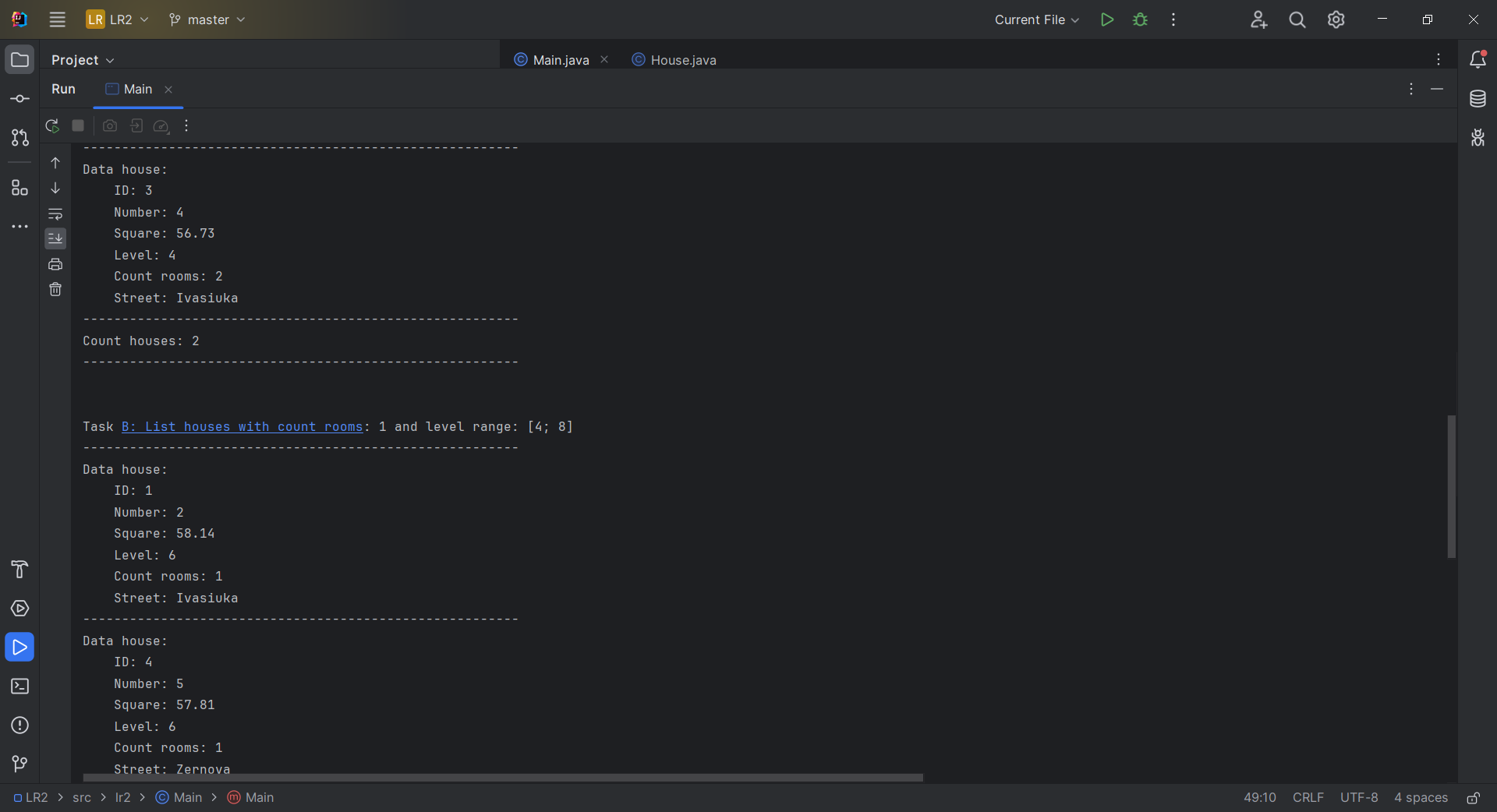


Рисунок 3 Результат програми виконання (продовження)

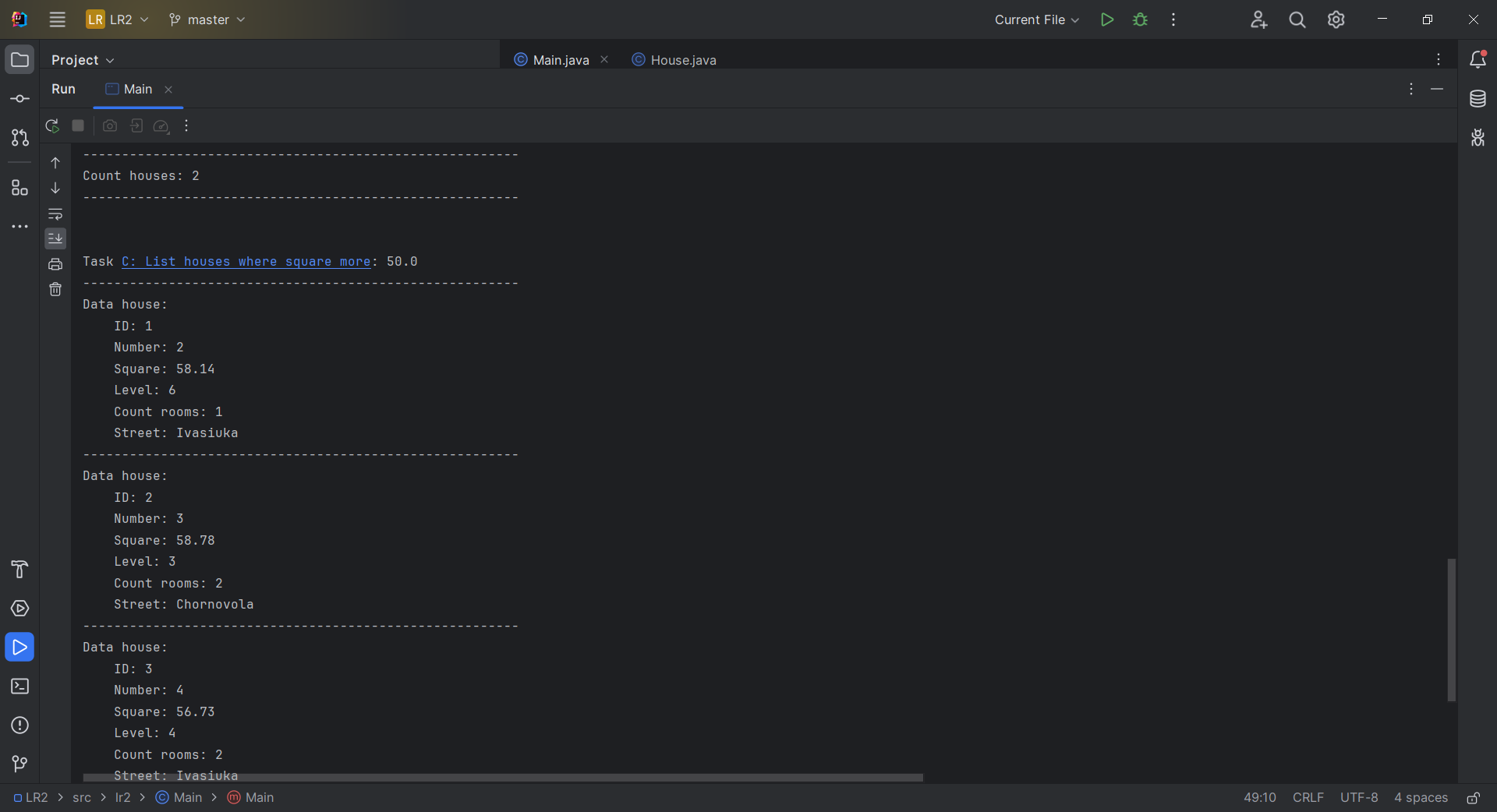


Рисунок 4 Результат програми виконання (продовження)

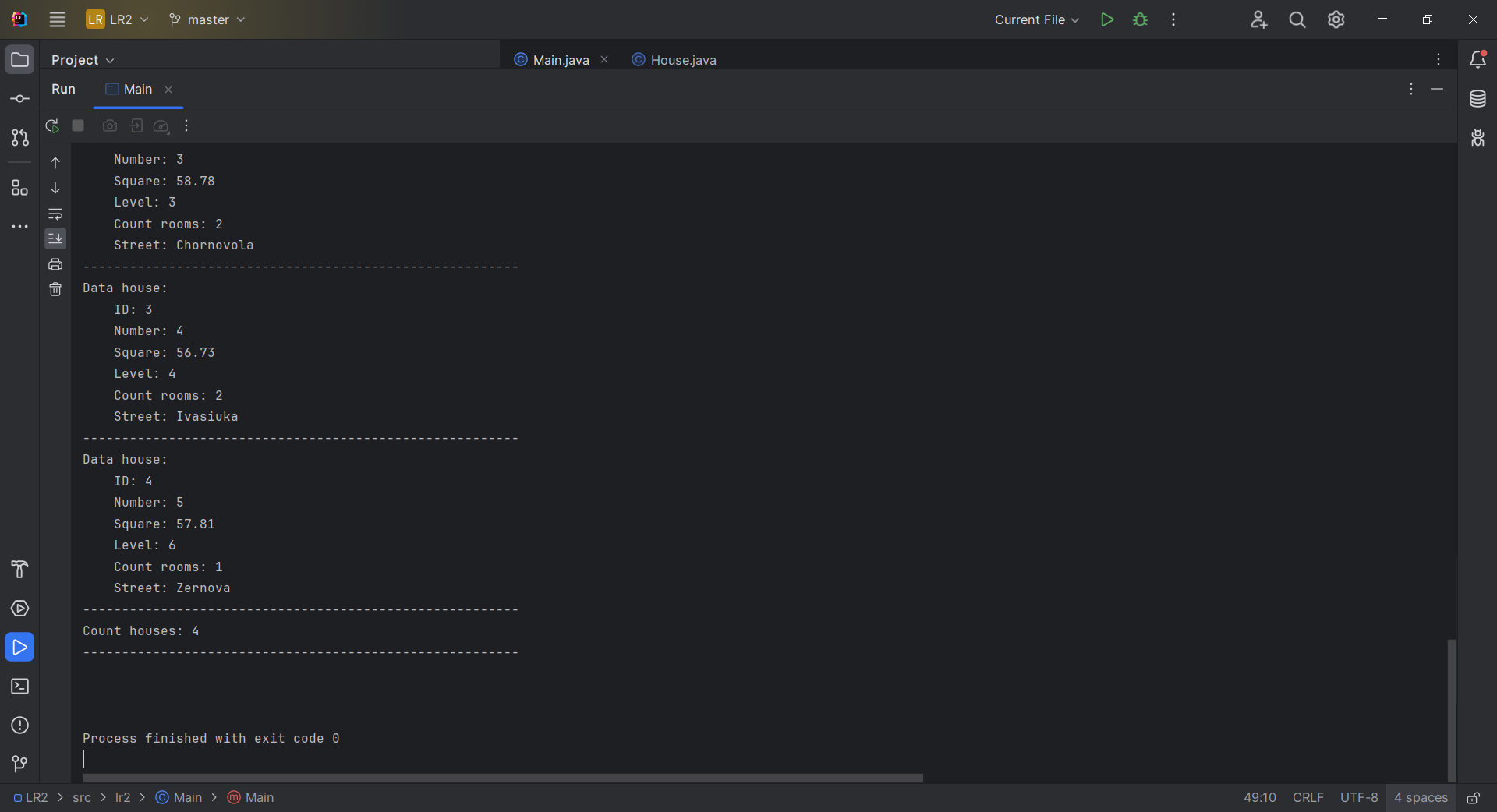


Рисунок 5 Результат програми виконання (продовження)

1. **Висновки**

У даній лабораторній роботі я реалізував поставлене індивідуальне завдання та зміг зробити клас у якому будуть зберігатися дані про будинок (квартиру) та поставлені пункти в Основному класі. Усі класи розміщуються у різних пакетах – lr2\_house й lr2.

Клас будинку (квартири) містить відповідні поля даних, які описані у завданні та сам лічильник об'єктів для їх ідентифікації (статичне поле). Також, є конструктор з параметрами, геттери й сеттери та перевантажений метод toString().

Основний клас містить поля даних масив класу будинку (квартир), масив вулиць (для генерації), рядок (для розподілу даних при виводу об’єктів). Щодо методів, то клас має конструктор з параметром кількості будинків, який генерує дані для будинку (квартири), методи пунктів завдань (taskA(), taskB(), taskC()), перевантажений метод toString(), для округлення чисел до двох чисел після коми roundDouble(double number), для виводу загальної кількості будинків outputCountHousesInTask(int counter).

Аналізуючи результати виконання програми було визначено, що програма працює коректно, бо виводить усі дані згідно поставлених умов у пунктах завдання.

Для лабораторної роботи було створено репозиторій на GitHub: <https://github.com/ManFromLviv/nulp_prikladne_programuvannia_lr2/blob/master/src/lr2/Main.java> .