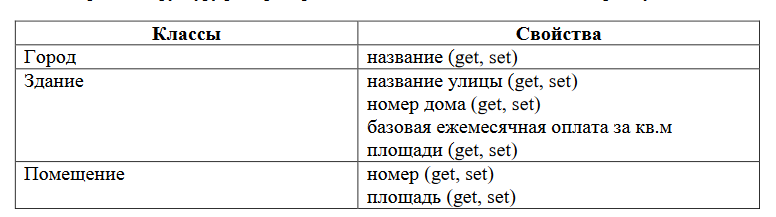
**ЛАБОРАТОРНАЯ РАБОТА №7**

**СОЗДАНИЕ КЛАССОВ И РАБОТА С ОБЪЕКТАМИ**

Цель работы: получить основные понятия и навыки по созданию классов, описанию свойств и работы с объектами.

**Вариант № 8**

а) Смоделировать структуру реестра городского жилья: (в отчет добавить диаграмму классов)

****

Создать один объект класса Город, два здания, и три объекта - помещения, задать значения свойств, вывести на экран.

б) Связать между собой классы Город, Здание и Помещение, так, чтобы каждый город содержал здания, а каждое здание содержало помещения; также каждое помещение должно содержать информацию о том, к какому зданию

оно относится.

- Добавить в класс Город множественное свойство «здания» (get) и методы «добавить здание» (add), «удалить здание» (remove).

- Добавить в класс Здание свойство «помещения» (get), а также методы «добавить помещение» (add), «удалить помещение» (remove). Добавить свойство «общая площадь» (get), рассчитываемое как суммарная площадь всех помещений в здании. - Добавить в класс «Помещение» свойство «здание»

Создать один объект класса Город, два здания, в первое добавить два объекта – помещения, во вторую – одно помещение. Вывести на экран информацию о первом здании (значения всех свойств, включая общую площадь) и помещениях в этом здании.

в) добавить в класс Город метод, осуществляющий поиск здания по названию улицы и номеру дома. Метод содержит входные параметры Название улицы (String) и Номер дома (int или String) возвращает значение типа Здание. Осуществить поиск здания в main(), вывести найденное здание (значения всех свойств) на экран.

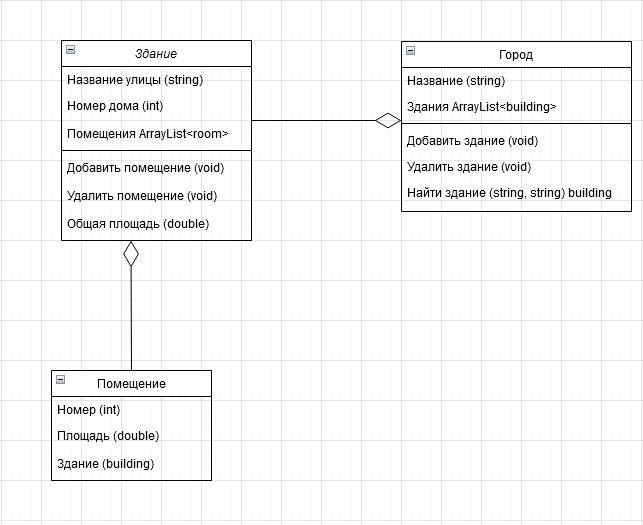


Рисунок 1 ­– Диаграмма классов реестра городского жилья

Класс “Город” содержит в себе поля: название, список зданий. Также класс содержит функции “Добавить здание” “Удалить здание” “Найти здание”. Так как класс “Здание” может существовать независимо от класса “Город”, но класс “Город” содержит в себе поле со списком зданий, то между собой они будут связаны агрегацией.

Класс “Здание” содержит в себе поля: название улицы, номер дома, список помещений. Также класс содержит функции “Добавить помещение” “Удалить помещение” “Общая площадь”.

Класс “Помещение” содержит в себе поля: номер, площадь, здание.

Соответственно, так как класс “Помещение” может существовать независимо от класса “Здание”, но класс “Здание” содержит в себе поле со списком помещений, то между собой они также будут связаны агрегацией.

**Листинг City.java**

import java.util.ArrayList;

public class City {

private String name;

private ArrayList<Building> buildings;

public City(String name) {

this.name = name;

this.buildings = new ArrayList<>();

}

public String getName() {

return name;

}

public void addBuilding(Building building) {

buildings.add(building);

}

public void removeBuilding(Building building) {

buildings.remove(building);

}

public ArrayList<Building> getBuildings() {

return buildings;

}

public Building findBuilding(String streetName, String houseNumber) {

for (Building building : buildings) {

if (building.getStreetName().equals(streetName) && building.getHouseNumber().equals(houseNumber)) {

return building;

}

}

return null; // if building not found

}

}

**Листинг Building.java**

import java.util.ArrayList;

public class Building {

private String streetName;

private String houseNumber;

private ArrayList<Room> rooms;

public Building(String streetName, String houseNumber) {

this.streetName = streetName;

this.houseNumber = houseNumber;

this.rooms = new ArrayList<>();

}

public String getStreetName() {

return streetName;

}

public String getHouseNumber() {

return houseNumber;

}

public void addRoom(Room room) {

rooms.add(room);

}

public void removeRoom(Room room) {

rooms.remove(room);

}

public ArrayList<Room> getRooms() {

return rooms;

}

public double getTotalArea() {

double totalArea = 0;

for (Room room : rooms) {

totalArea += room.getArea();

}

return totalArea;

}}

**Листинг Room.java**

public class Room {

private int roomNumber;

private double area;

private Building building;

public Room(int roomNumber, double area, Building building) {

this.roomNumber = roomNumber;

this.area = area;

this.building = building;

}

public int getRoomNumber() {

return roomNumber;

}

public double getArea() {

return area;

}

public Building getBuilding() {

return building;

}}

**Листинг Main.java**

public class Main {

public static void main(String[] args) {

// Создаем город

City city = new City("Example City");

// Создаем первое здание и добавляем два помещения

Building building1 = new Building("First Street", "123");

building1.addRoom(new Room(101, 50.0, building1));

building1.addRoom(new Room(102, 70.0, building1));

city.addBuilding(building1);

// Создаем второе здание и добавляем одно помещение

Building building2 = new Building("Second Street", "456");

building2.addRoom(new Room(201, 60.0, building2));

city.addBuilding(building2);

// Выводим информацию о первом здании

System.out.println("Building information:");

System.out.println("Street Name: " + building1.getStreetName());

System.out.println("House Number: " + building1.getHouseNumber());

System.out.println("Total Area: " + building1.getTotalArea() + " square meters");

// Выводим информацию о помещениях в первом здании

System.out.println("\nRooms in the first building:");

for (Room room : building1.getRooms()) {

System.out.println("Room Number: " + room.getRoomNumber());

System.out.println("Area: " + room.getArea() + " square meters");

}

// Поиск здания по названию улицы и номеру дома

Building foundBuilding = city.findBuilding("First Street", "123");

if (foundBuilding != null) {

// Выводим информацию о найденном здании

System.out.println("\nFound Building:");

System.out.println("Street Name: " + foundBuilding.getStreetName());

System.out.println("House Number: " + foundBuilding.getHouseNumber());

System.out.println("Total Area: " + foundBuilding.getTotalArea() + " square meters");

} else {

System.out.println("\nBuilding not found.");

}

}

}

**Результат**

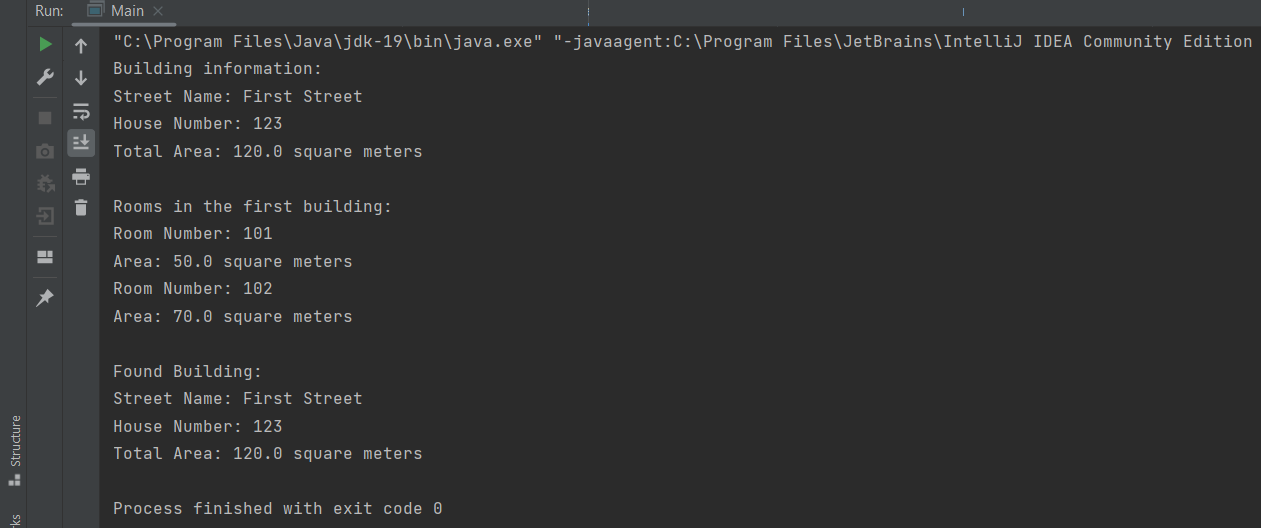


Рисунок 2 – результат выполнения программы