

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение
высшего образования
«Магнитогорский государственный технический университет им. Г. И.
Носова»
(ФГБОУ ВО «МГТУ им. Г.И. Носова»)

Кафедра вычислительной техники и программирования

Лабораторная работа №4

**по дисциплине «Метрология и стандартизация программного
обеспечения»**

**название лабораторной работы: «Оценка качества программы с
помощью объектно-ориентированных метрик. Часть 1»**

Исполнитель: Варламов М.Н., студент 3 курса, группа АВб-19-1

Руководитель: Сибилева Н.С., ст. преподаватель каф. ВТиП

Магнитогорск, 2022

Оглавление

Задание	3
Структурный анализ программного кода	3
Анализ качества программного кода с помощью метрик	5
Выводы	5

1. Задание

Разработать структуру классов. Базовый класс – помещения. Производные – квартира и офис. Создать класс Дом, который может содержать оба вида объектов. Предусмотреть метод подсчета отдельно квартир и офисов (использовать оператор instanceof).

2. Структурный анализ программного кода

На рисунке 1 представлены блок-схемы алгоритмов каждого из методов класса.

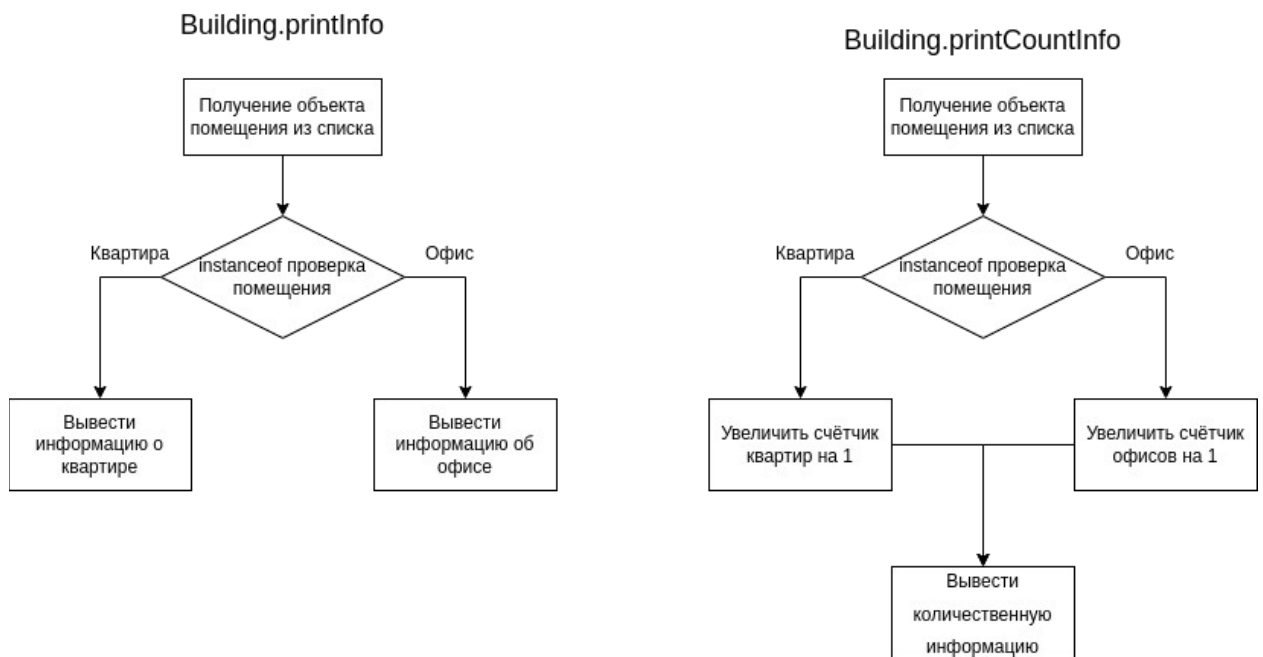


Рисунок 1 - блок-схемы алгоритмов методов класса.

На рисунках 2, 3 и 4 представлен исходный код программы.

```
public interface Room {
    1 usage 2 implementations
    String getInfo();
}
```

Рисунок 2 - Интерфейс помещения.

```

package Lab4;

public class Office implements Room {
    2 usages
    private final int placesCount;

    1 usage
    public Office(int placesCount) {
        super();
        this.placesCount = placesCount;
    }

    1 usage
    @Override
    public String getInfo() { return "Work places count: " + placesCount; }
}

```

Рисунок 3 - Класс офиса

```

public class Building {
    6 usages
    private final List<Room> rooms;
    1 usage
    public Building() { rooms = new ArrayList<>(); }
    2 usages
    public void addRoom(Room room) { rooms.add(room); }
    1 usage
    public void printInfo() {
        for (int i = 0; i < rooms.size(); i++) {
            Room room = rooms.get(i);
            String type = "";
            if (room instanceof Office)
                type = "office";

            if (room instanceof Flat)
                type = "flat";

            System.err.println("On " + (i + 1) + " floor placed " + type + ". Info " + room.getInfo());
        }
    }

    1 usage
    public void printCountInfo() {
        int officeCount = 0;
        int flatCount = 0;

        for (Room room : rooms) {
            if (room instanceof Office)
                officeCount++;
            if (room instanceof Flat)
                flatCount++;
        }

        System.err.println(
            "Flats count: " + flatCount + "\n" +
            "Offices count: " + officeCount + "\n" +
            "Common count " + rooms.size());
    }
}

```

Рисунок 4 - Класс строения

На рисунке 5 представлен результат работы программы.

```
Added!
1 - Add office
2 - Add flat
3 - Print building info
4 - Print building count info
5 - Exit
Choose action:2
Enter rooms count :3

Added!
1 - Add office
2 - Add flat
3 - Print building info
4 - Print building count info
5 - Exit
Choose action:2
Enter rooms count :4

Added!
1 - Add office
2 - Add flat
3 - Print building info
4 - Print building count info
5 - Exit
Choose action:4
Flats count: 2
Offices count: 2
Common count 4
1 - Add office
2 - Add flat
3 - Print building info
4 - Print building count info
5 - Exit
Choose action:|
```

Рисунок 5 результат работы программы.

На рисунке 2 представлена блок схема общей работы программного продукта.



Рисунок 2 - блок схема общей работы программного продукта.

3. Анализ качества программного кода с помощью метрик

Далее следует таблица 1 с метриками Мартина.

Название	Расчёт	Значение
Центростремительное сцепление (Ca)	-	0
Центробежное сцепление (Ce)	-	0
Нестабильность (I)	$Ce / (Ca + Ce)$	0
Абстрактность (A)	$nA / nAll$	0.25
Расстояние до главной последовательности (D)	$ (A + I - 1) / \sqrt{2} $	0.53
Нормализованное расстояние до главной последовательности (Dn)	$ A + I - 2 $	1.75

4. Выводы

Была разработана структура классов. Базовый класс – помещения. Производные – квартира и офис. Создать класс Дом, который может содержать оба вида объектов. Предусмотреть метод подсчета отдельно квартир и офисов (использовать оператор instanceof).

С помощью метрик Мартина было выяснено, что написанная структура является полностью стабильна и мало абстрактна, так как показатель абстрактности = 0.25, а показатель стабильности 0.