**Министерство образования республики Молдова Технический университет Молдовы**

**Кафедра «Автоматика и информационные технологии»**

**Отчёт**

**Лабораторная работа № 1**

**Тема: Структура – механизм абстракции**

**Вариант 6**

**Выполнил студент гр. TI-197** Макеев Никита

**Проверил:** Брынзан Л.

Оглавление

[ЛАБОРАТОРНАЯ РАБОТА №1 3](#_Toc51538475)

[Тема: Структура – механизм абстракции 3](#_Toc51538476)

[Цели работы: 3](#_Toc51538477)

[Задание: 3](#_Toc51538478)

[Реализация задания: 3](#_Toc51538479)

[(Листинг номер 1). 5](#_Toc51538480)

[(Листинг программы 2). 6](#_Toc51538481)

[(Листинг программы 3,4). 7](#_Toc51538482)

[(Листинг номер 5). 8](#_Toc51538483)

[(Листинг номер 6). 9](#_Toc51538484)

[(Листинг номер 7). 9](#_Toc51538485)

[Код лабораторной работы 10](#_Toc51538486)

[Вывод: 10](#_Toc51538487)

# ЛАБОРАТОРНАЯ РАБОТА №1

## **Тема: Структура – механизм абстракции**

**Цели работы: изучение основ абстракции;изучение правил определения и использования структур данных;создание переменных типа структуры, обращение к полям;изучение принципов программирования основанное на работе со структурами.**

## **Задание:**

Создать абстрактный тип данных (структура) - дом, у которой есть название фирмы строителя, адрес, количество этажей и квартир. Определить функции установки, изменения данных, сравнения домов. Для задания текстовых полей использовать оператор new. Освободить память. В main-е, привести пример сортировки домов по количеству этажей + адресу в алфавитном порядке.

# Реализация задания:

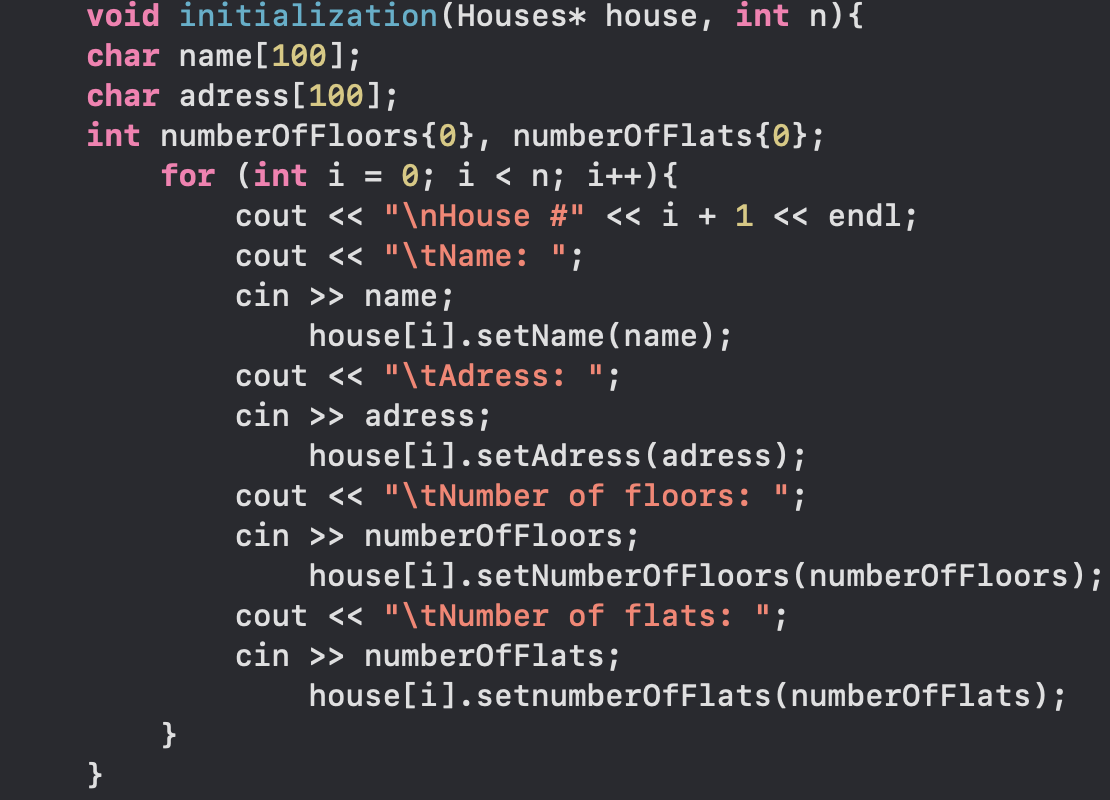
Создаем абстрактный тип данных (структуру) с называнием дом , у которой объявляем переменные с названием фирмы строителя , адрес дома , количество этажей и квартир. И далее создаем методы get и set для соответствующих полей.

( см.листинг 1).Данные поля имеют модификаторы private для того чтобы у пользователя не было доступа к ним, а сами методы get и set имеют модификаторы public для того чтобы была возможность получать данные полей в дальнейших функциях или методах.



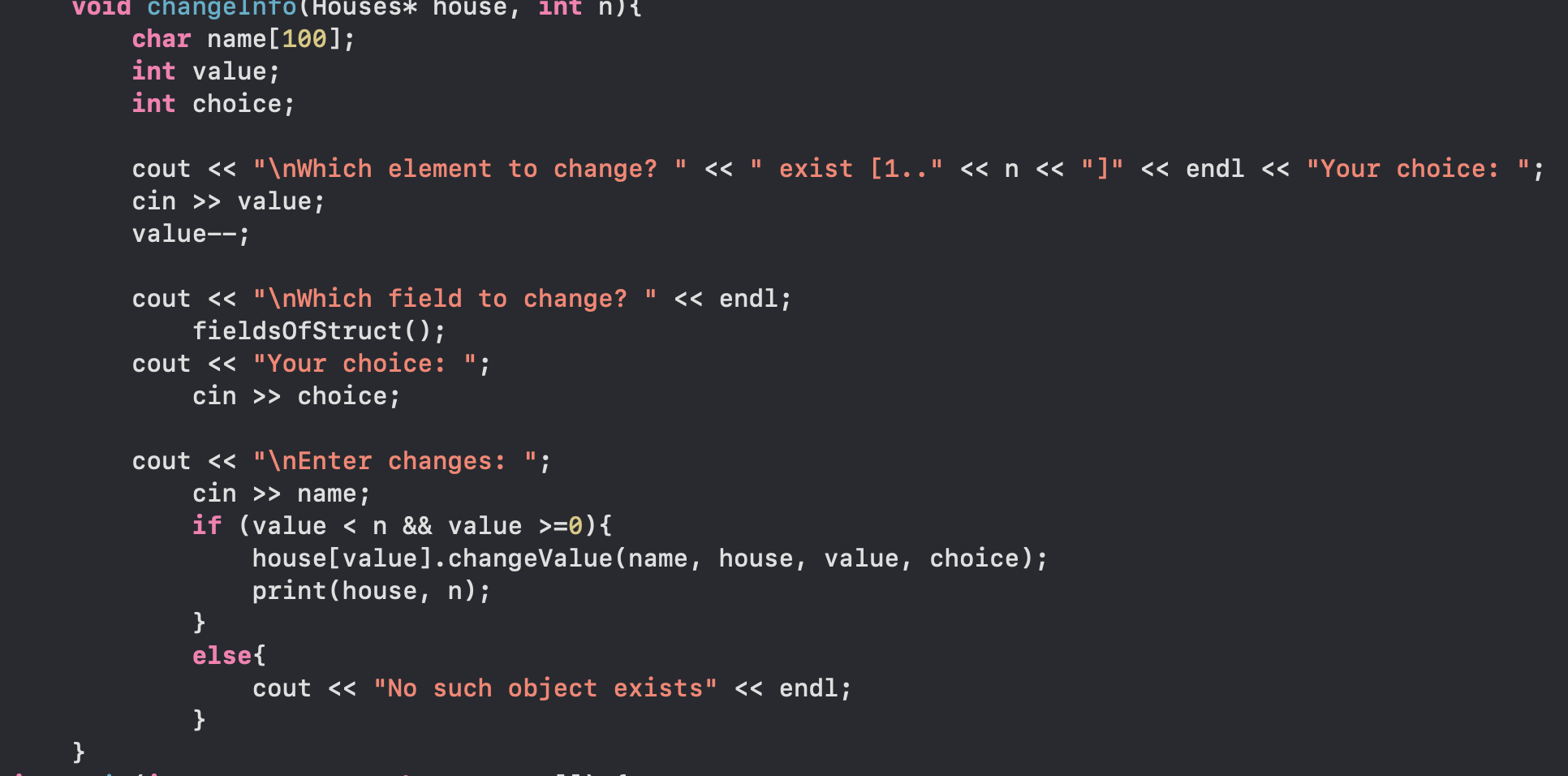
## **(Листинг номер 1).**

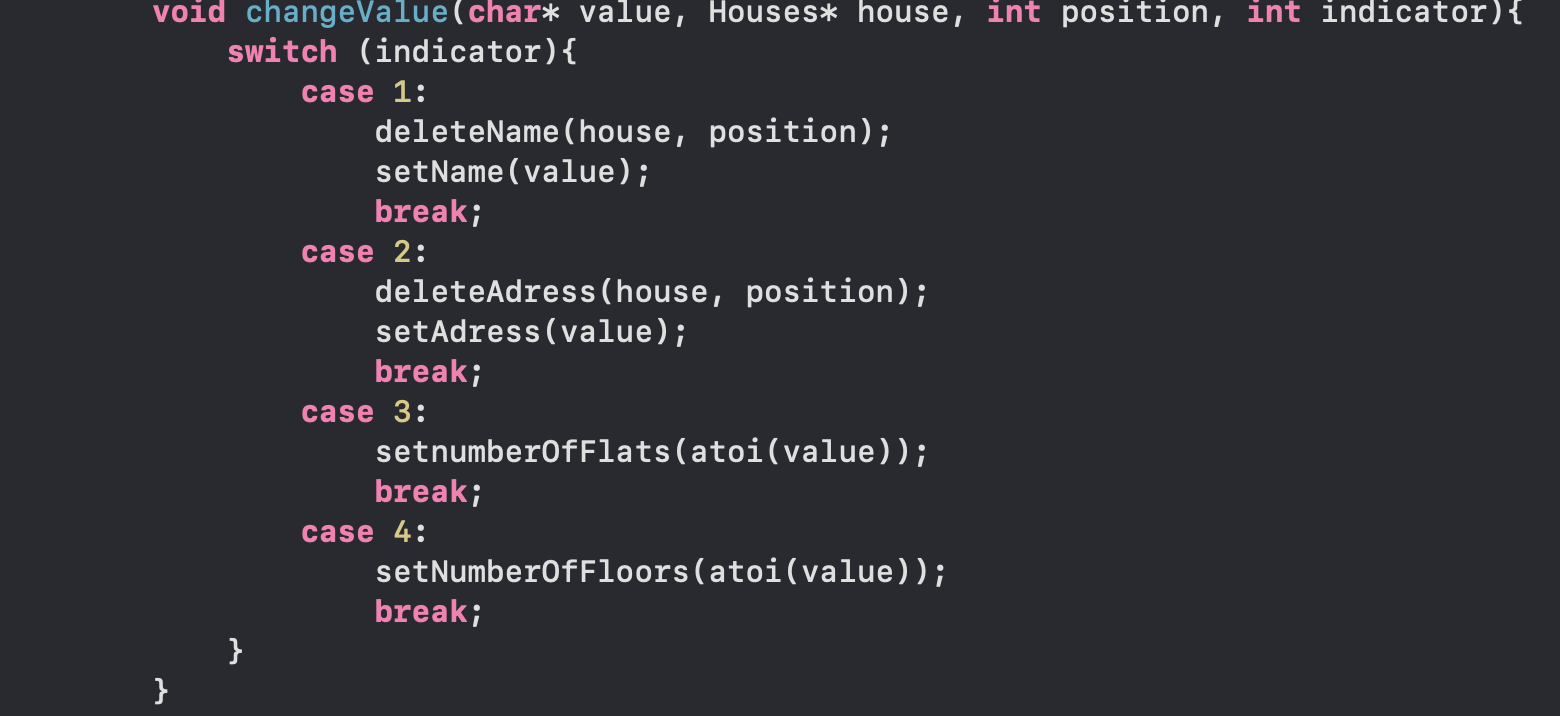
Следующие этапы – это создание функций для реализации установки, изменения данных и сравнения домов (см листинг номер 2,3,4,5).



## **(Листинг программы 2).**

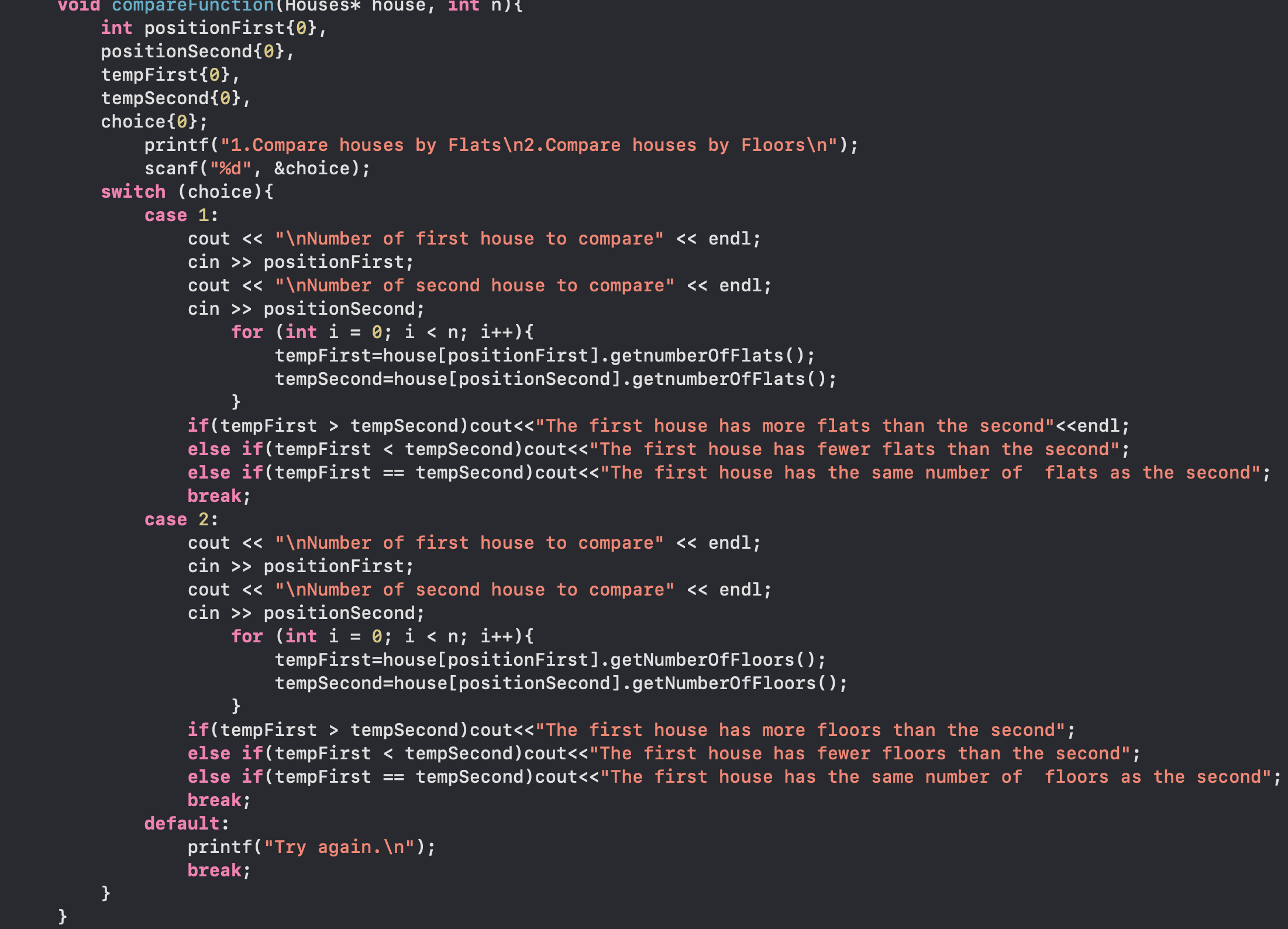
Данной функции мы передаем 2 переменные n – кол-во домов и house – сами дома, которые имеют различные поля. В самой функции было объявлено 4 переменные, которые хранят значения введенные пользователем, также был реализован цикл for для того чтобы заполнять определенное кол-во полей структуры данными, кол-во и сами дынные пользователь вводит самостоятельно.





## **(Листинг программы 3,4).**

В данной функции мы изменяем данные дома, номер которого вводит сам пользователь номер дома записывается в переменную value, которая в свою очередь меняет свое значение на -1 позицию это вызвано тем, что в языках программирования отсчет начинается с 0, а в обычной жизни с 1. Был также реализован оператор if для того чтобы проверить существует ли дом, под номером, который ввел сам пользователь. Если такой дом существует, то вызывается метод changeValue(), который и запрашивает новые данные и удаляет старые.



## **(Листинг номер 5).**

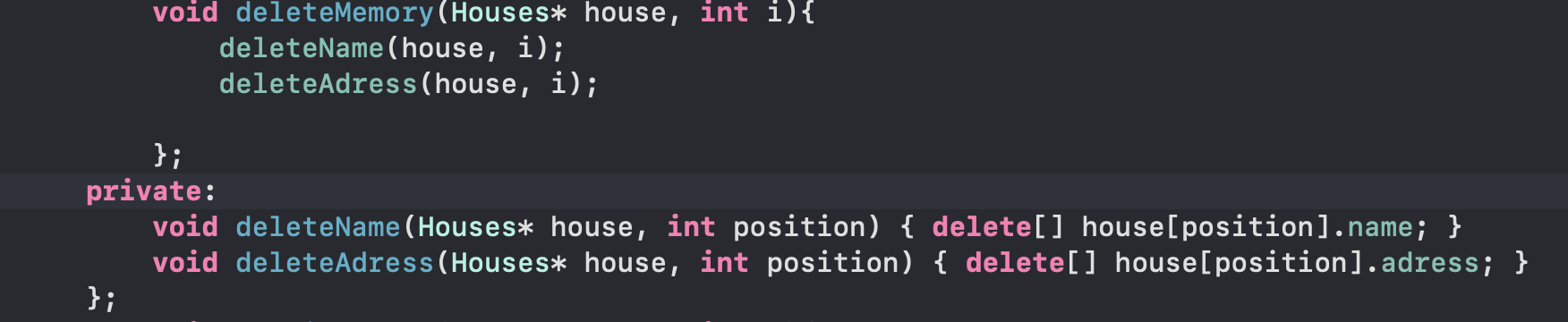
В данной функции реализуется сравнение домов, какие номера домов сравнить пользователь вводит сам. Функция сравнивает дома на выбор или по кол-ву этажей, или по кол-во квартир, это пользователь также выбирает сам. Для самого сравнения были созданы 4 переменные, 2-е переменные получают значения позиции домов, а остальные две сделаны, как вспомогательные для того чтобы из цикла for получить данные по этим домам и сравнить их.

Второй этап – это создание функций для реализации сортировки домов и очистка памяти (см листинг номер 6,7).



## **(Листинг номер 6).**

В данной функции был реализован алгоритм сортировки пузырьком. Также, как и в других функция пользователь сам выбирает по какому параметру ему сортировать дома. Например, по кол-ву квартир или по названию фирмы строителя.



## **(Листинг номер 7).**

Для удаления памяти был инициализирован метод с идентификатором public, который в свою очередь вызывает два метода, (где и происходил очистка памяти) с идентификаторами private, это было сделано для того чтобы у пользователя не было доступа напрямую для удаления двух динамических переменных

# Код лабораторной работы доступен по ссылке https://github.com/Randwow/oop\_lab1

# Вывод:

В данной лабораторной работе былизучен абстрактный тип данных, изучены правила определения и использования структур данных, создание переменных типа структуры, обращение к полям и изучение принципов программирования, основанное на работе со структурами. Также были реализованы и изучены методы get и set, функции очистки памяти, сортировки массива структур, инициализация, изменения данных и сравнение элементов массива структур.