



**СУ “Св. Климент Охридски”,  
ФМИ – Софтуерно инженерство  
Курсов проект по Обектно-ориентирано  
Програмиране практикум**

# **Real Estate**

Румен Събев Чолаков, Факултетен № 61921

## Съдържание

1. Въведение	2
2. Описание на приложените алгоритми	2
3. Описание на програмния код	2
4. Използвани технологии	4

### 1. Въведение

Проектът се състои от класовете Estate (абстрактен клас), Flat, House, Plot (наследници на Estate), RealEstate, който съдържа контейнер от указатели към обекти клас Estate и набор от методи описани в заданието.

### 2. Описание на приложените алгоритми

За сортирането на имотите по цена съм използвал bubble sort. Чрез него елементите се подреждат в подходящия ред. Bubble sort алгоритъмът минава през масива и постепенно подрежда елементите в нарастващ или намаляващ ред спрямо цената им и след това прави финално обхождане за да провери дали всеки елемент е на мястото си.

### 3. Описание на програмния код

#### 3.1 Estate

Съхранява информация за адрес на имота, собственик, цена, площ. Освен необходимите конструктори, методи е дефиниран и метод за печат.

#### 3.2 Flat

Наследник на клас **Estate**. Съдържа информация за брой стаи и етаж.

#### 3.3 House

Наследник на клас **Estate**. Съдържа с информация брой стай, брой етажи, площ на двора.

### 3.4 Plot

Наследник на клас **Estate**. Съдържа информация за налични комуникации (списък от елементи на избран тип Communication (вода, ток, телефон, път, канал)

### 3.5 bool ErrorCheck(istream& s)

Проверява дали е въведен коректна информация от потребителя и се грижи да не възникват грешки, ако тя не е такава. Освен това служи за условие на цикли позволяващи повторно въвеждане на информацията.

### 3.6 RealEstate

Съдържа контейнер от указатели към обекти клас Estate следните методи:

**searchType()** – търсене по вида на имота

**searchOwner()** – търсене по собственика на имота

**searchAddress(string)** – търсене по адреса на имота. При викане на функцията от потребителя може да се търси и по част от адреса например само населеното място, а когато е извикана от **RemoveEstate()** е нужен целият адрес

**searchArea()** – търсене на имоти с площ в даден диапазон

**searchPrice()** – търсене на имоти с цена в даден диапазон

**searchYardArea()** – търсене на имоти с площ на двора в даден диапазон

**searchNumberRooms()** – търсене на имоти с брой на стаите в даден диапазон

**searchNumberFloors()** – търсене на имоти с брой на етажите в даден диапазон

**searchOnFloor()** – търсене на имоти, които са на етаж, който е в даден диапазон

**searchCommunication()** – търсене на имоти със съответните комуникации

**TypeSort()** – извежда всички имоти от зададен вид, сортирани по цена в низходящ ред. За намирането на имотите използва **searchType()**, а за сортирането Bubble sort

**PrintAddress()** – връща и извежда списък от имоти от зададено селище. За намирането на имотите използва

**searchAddress()** и ползвайки указателите към съответните имоти вика метода за принтиране на дадения обект.

**AddEstate()** и **RemoveEstate()** – методи за добавяне на имот и изтриване на имот. **RemoveEstate()** използва **searchType()**, **searchOwner()** и **searchAddress(string)** за да намери имота, който да премахне.

#### 4. Използвани технологии

IDE: Code Blocks

Език за програмиране: C++