Картина, която съдържа текст, графична колекция

Описанието е генерирано автоматичноТЕХНИЧЕСКИ УНИВЕРСИТЕТ – ВАРНА

Факултет по изчислителна техника и автоматизация

Катедра „СИТ“

**СЕМЕСТРИАЛНА ДОМАШНА РАБОТА**

по дисциплината „Базово програмиране”

на тема: „*Информационна система Агенция за имоти*”

Вариант: 348

|  |  |
| --- | --- |
| Изготвил: Атанас Петров Маргаритов | Проверил: |
| Специалност: Киберсигурност |  |
| Група: 1А |  |
| Факултетен номер: 23621811 |  |

2024

Съдържание

[I. Задание на проекта 4](#_Toc154664811)

[II. Анализ на решението 6](#_Toc154664812)

[1. Структура за данните в програмата 6](#_Toc154664813)

[2. Реализация на условие: A Меню за избор на функциите от програмата 7](#_Toc154664814)

[3. Реализация на условие: B Добавяне 9](#_Toc154664815)

[4. Реализация на условие: C Извеждане на екрана](#_Toc154664816) 16

[5. Реализация на условие: D Търсене и извеждане на екрана 2](#_Toc154664817)0

[6. Реализация на условие: E Подреждане на основния масив с имоти, без да се извежда на екрана](#_Toc154664818)  23

[7. Реализация на условие: F Управление на файл 2](#_Toc154664819)6

[8. Реализация на условие … - допълнение първо Подменю 1 2](#_Toc154664820)9

[9. Реализация на условие … - допълнение второ Огледи на имот 36](#_Toc154664821)

[10. Реализация на допълнение трето 45](#_Toc154664822)

[III. Упътване за употреба 49](#_Toc154664823)

Задание на проекта

Да се напише компютърна програма, реализираща информационна система, която

поддържа агенция за имоти. Програмата съхранява и обработва данни за имотите в агенцията

[референтен номер (уникален, състои се от 6 символа, комбинация от цифри и букви), квартал,

улица, площ, цена, брой стаи, статус (в продажба, запазен или продаден)]. Максималният брой

имоти в агенцията е 50.

Базова задача

A. Меню за избор на функциите от програмата

Функции от програмата са:

B. Добавяне:

a. Добавянето трябва да позволява да се въвеждат различен брой нови имоти с

координати в града (в града има 5 квартала - Младост, Чайка, Левски, Център,

Гръцки), като се допълва масива с имоти. Не трябва да превишава максималния

брой имоти на агенцията (50).

Пример: Въвежда се цяло число n, което позволява да се въведат n на брой имоти,

като n не може да надвишава свободните елементи в масива с имоти.

C. Извеждане на екрана

a. Извеждане на всички имоти в оформен вид

D. Търсене и извеждане на екрана:

a. Намиране и извеждане на имота с най-ниска цена на квадратен метър;

b. Търсене на всички имоти от даден квартал.

E. Подреждане на основния масив с имоти, без да се извежда на екрана:

a. Подреждане на имотите по нарастващ ред на площта им.

F. Управление на файл:

a. Извеждане на масива с имоти във файл (двоичен)

b. Въвеждане на масива с имоти от файл (двоичен)

Допълнение първо (+ базова задача)

G. Създайте подменю, в което се влиза от основното, с нови функции за:

a. Извеждане на имотите в продажба, подредени възходящо по цена на квадратен

метър;

b. Търсене и извеждане на имоти по въведени брой стаи и максимална цена, която

клиентът е склонен да заплати;

Допълнение второ (+ базова задача)

H. Огледи на имот

a. Добавете в структурата на имота брой огледи със стойност по подразбиране 0.

b. Огледът започва с въвеждане на референтен номер. Ако в имотите, които

предлага агенцията, подобен референтен номер е открит, да се провери неговия

статус:

a. В продажба. Регистрира се направения оглед, като броят огледи се

увеличи с 1. Отправя се запитване дали да се смени статуса на имота на

„запазен“ или „продаден“.

Ако изборът е „да“, промяната да се извърши.

Ако изборът е „не“, да се отправи запитване дали да се

визуализират други предложения със същия брой стаи 1) от същия

Квартал 2) от съседни квартали. Да се изведат резултатите спрямо

направения избор.

b. „Запазен“ или „продаден“. Да се изведе

съобщение, че имотът не е обект на

продажба.

I. Да се добави функционалност, която да извежда квартала с най-трудно продаваеми

имоти (най-голям среден брой огледи на имоти, които все още са в продажба).

Допълнение трето (+ базова задача)

J. Данните в програмата да се попълват автоматично от файл при стартиране и да се

записват автоматично във файл при затваряне на програмата.

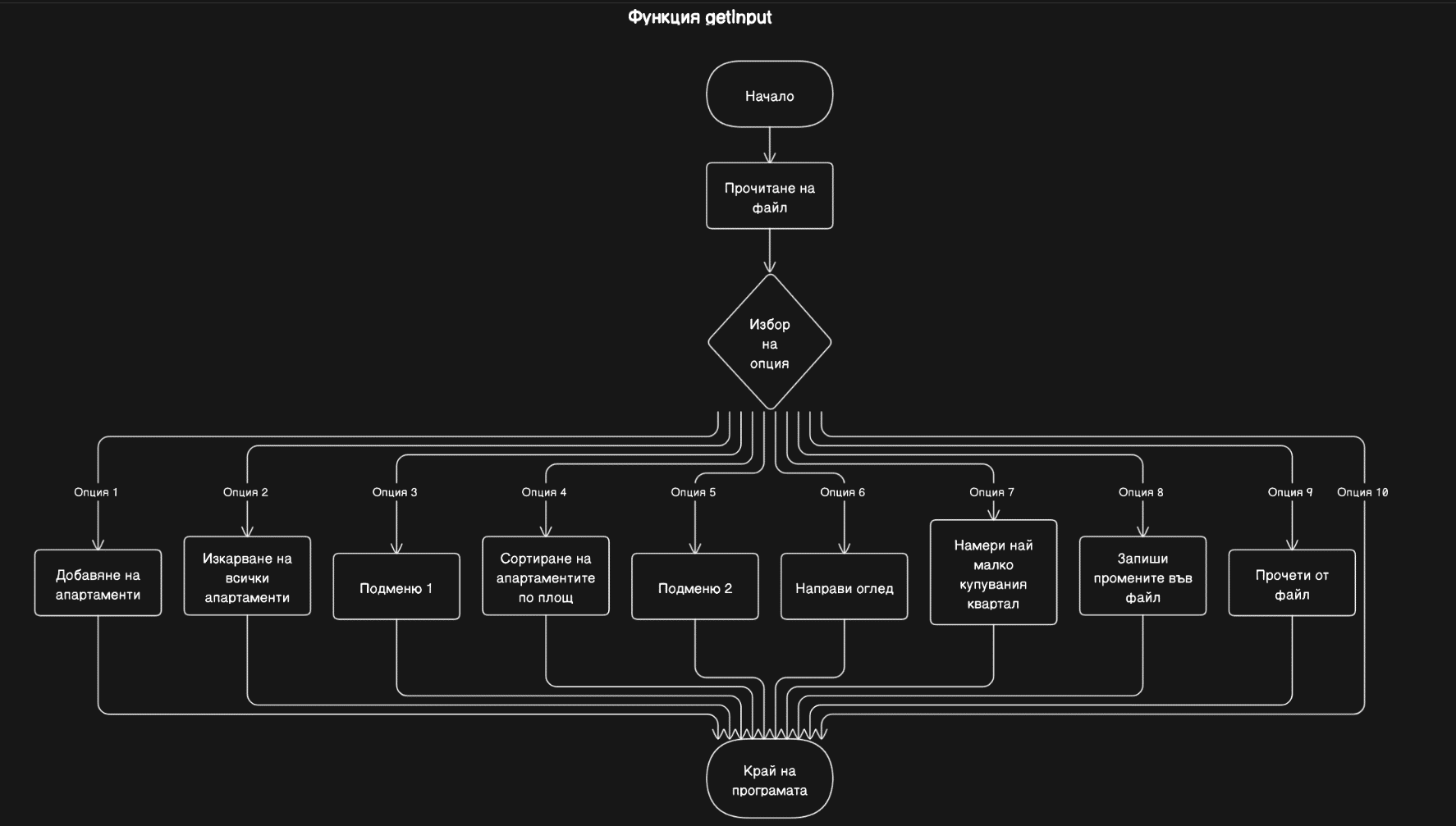
Анализ на решението

Структура за данните в програмата

|  |  |  |
| --- | --- | --- |
| Структура | Обяснение | Примерени стойности |
| struct Apartment  {  string ID;  string street;  int neighbourhood;  double area;  double price;  int rooms;  int status = 0;  int inspections = 0;}; | * ID уникален номер на апартамент(Генерира се автоматично) * Квартал в, който се намира имота от 1 до 5   (1-Чайка 2-Левски 3-Център 4-Гръцка 5-Младост)   * Улица на, която се намира апартамента * Площ на апартамента * Колко стаи има апартамента * Състояние на апартамента   (0-Продаден 1- В продажба 2- Запазен)   * Брой огледи направени | Car Petur  2  100  150000  3  1 |

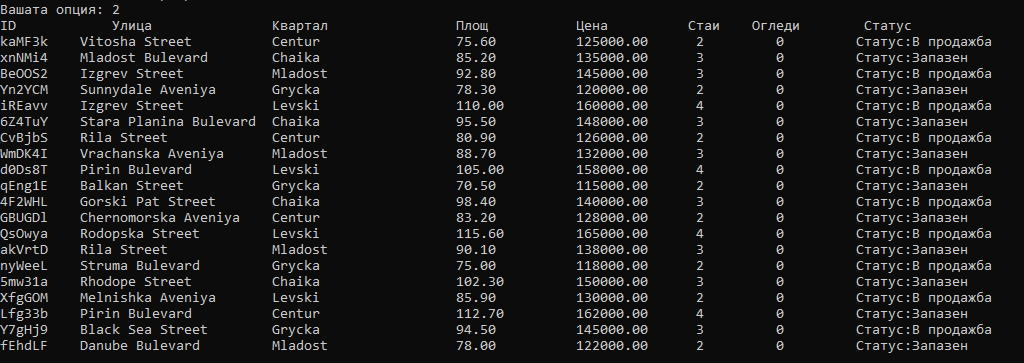
Реализация на условие А

Условие А е меню, което е създадено да прави работата с програмтата по-лесна. (Опция 1 – добави апартаменти, Опция 2- Изпечатай всички апартаменти, Опция 3 -Меню 1 Опция 4- Сортира по площ възходящо всички апартаменти без да ги изпечатва, опция 5- Меню2, Опция 6 -Правене на оглед на апартамент Опция 7- Намиране на квартала с апартаменти с най-много огледи без продабжа, Опция 8 – записване на масива във файл, Опция 9- четене от файл, Опция 10 – Изход от програмата)

* 1. Прави се един switch case със 9 кейса като два от тях съдържат подменю (3 и 5) с по един switch case в себе си.
  2. Блок схема на главното меню

1. void getInput(int& count, Apartment allAparts[])
2. {
3. ReadFile(allAparts, count);
4. int input;
5. int secondInput;
6. int thirdInput;
7. do
8. {
9. cout << endl << "Опция 1: Добавяне на апартаменти, " << endl
10. << "Опция 2: изкарване на всички апартаменти" << endl
11. << "Опция 3: Подменю 1" << endl
12. << "Опция 4: Сортиране на апартаментите по площ" << endl
13. << "Опция 5: Подменю 2" << endl
14. << "Опция 6: Направи оглед" << endl
15. << "Опция 7: Намери най-малко купувания квартал" << endl
16. << "Опция 8: Запиши промените във файл" << endl
17. << "Опция 9: Прочети от файл" << endl
18. << "Опция 10: Излез от програмата" << endl
19. << "Вашата опция: ";
20. validates(input);
21. switch (input)
22. {
23. case 1:
24. {
25. int numToAdd;
26. do
27. {
28. cout << "Колко апартамента искате да добавите (максимум 50)" << endl;
29. validates(numToAdd);
30. if (numToAdd > 50)
31. {
32. cout << "Невалиден брой апартаменти (максимум 50)" << endl;
33. }
34. if (numToAdd <= 0)
35. {
36. cout << "Невалиден брой апартаменти (минимум 1)" << endl;
37. }
38. if (numToAdd + count > 50)
39. {
40. cout << "Невалиден брой апартаменти (максимален капацитет 50)" << endl;
41. }
42. } while (numToAdd > 50 || numToAdd < 0 || numToAdd + count > 50);
43. string Neighb;
44. for (int i = 1; i <= numToAdd; i++)
45. {
46. if (!(numToAdd > 50 || numToAdd <= 0 || numToAdd + count > 50))
47. {
48. CreateAnAppartment(allAparts, count, Neighb);
49. }
50. }
51. if (!(numToAdd > 50 || numToAdd <= 0 || numToAdd + count > 50))
52. {
53. cout << numToAdd << " Добавени апартаменти";
54. }
55. }break;
56. case 2:
57. displayArray(allAparts, count);
58. break;
59. case 3:
60. {
61. cout << "1- Показва апартаменти от посочен квартал" << endl
62. << "2- Намиране на апартамента с най-добро съотношение цена на квадратен метър" << endl;
63. do
64. {
65. validates(thirdInput);
66. } while (!thirdInput == 1 && !thirdInput == 2);
67. switch (thirdInput)
68. {
69. case 1:
70. displayBestValue(allAparts, count);//извежда апартамента с най-добро съотношение цена на квадратен метър
71. break;
72. case 2:
73. inNeighbour(allAparts, count);//извежда имоти по цена и брой стаи
74. break;
75. }
76. }
77. break;
78. case 4:
79. sortByArea(allAparts, count);
80. break;
81. case 5:
82. {
83. cout << "Изберете опция 1- подрежда апартаментите по цена на квадратен метър и ги извежда" << endl
84. << "Опция 2- извежда апартаменти по максимална цена и брой стаи" << endl;
85. do
86. {
87. validates(secondInput);
88. } while (secondInput != 1 && secondInput != 2);
89. switch (secondInput)
90. {
91. case 1:
92. displayByPriceToArea(allAparts, count);
93. break;
94. case 2:
95. displayRoomsByPrice(allAparts, count);
96. break;
97. if (secondInput != 1 && secondInput != 2)
98. {
99. cout << "Невалидна опция изберете 1 или 2" << endl;
100. }
101. }
102. }
103. break;
105. case 6:
106. propertyInspection(allAparts, count);
107. break;
108. case 7:
109. FindUnwantedNeighbourhood(allAparts, count);
110. break;
111. case 8:
112. WriteToFile(allAparts, count);
113. break;
114. case 9:
115. ReadFile(allAparts, count);
116. break;
117. }
118. } while (input != 10);
119. WriteToFile(allAparts, count);
120. }

Числото: 2 ще извика функцията displayArray и ще изпечата всички апартаменти:



Функция с която се изпълнява алгоритъма:

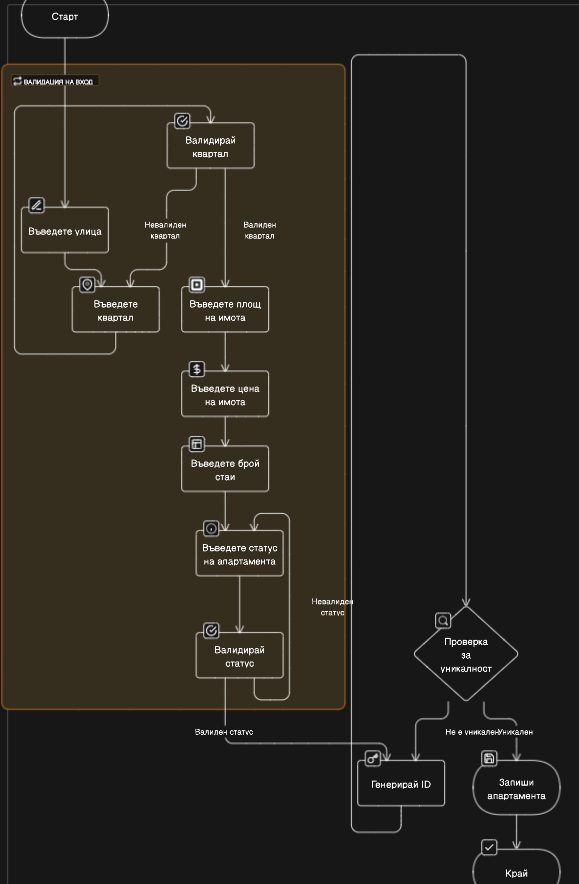
Реализация на условие B

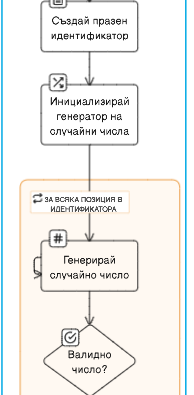
Анализ на алгоритъма, който трябва да се реализира

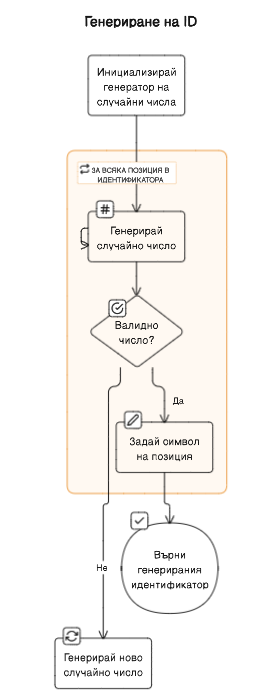
Моля, опишете.

За добавянето на нов елемент в масива от Apartment трябва да се валидира всяка една променлива, която се подава към инстанцията и да се генерира автоматично уникално ID.

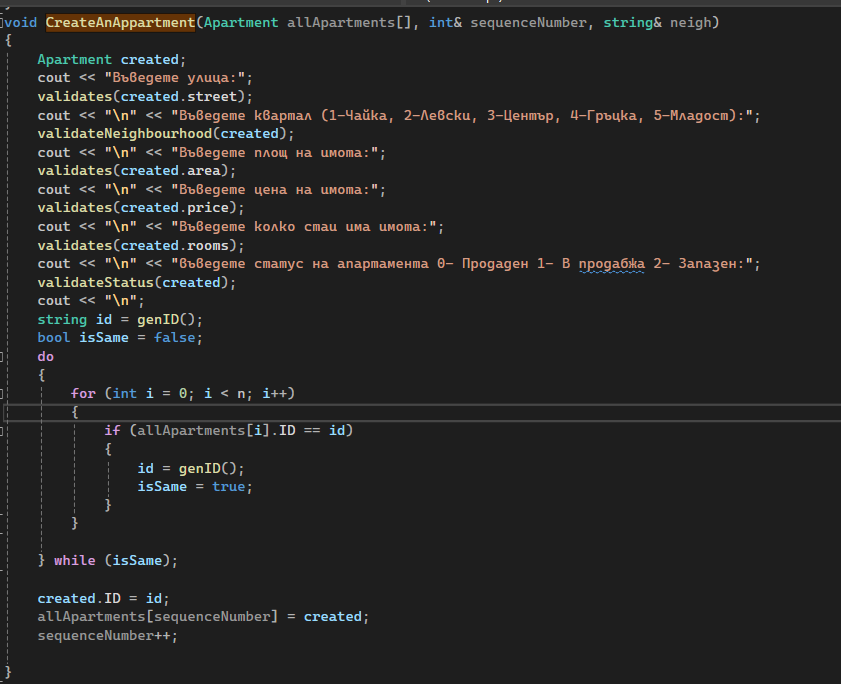
Блок схема на алгоритъма







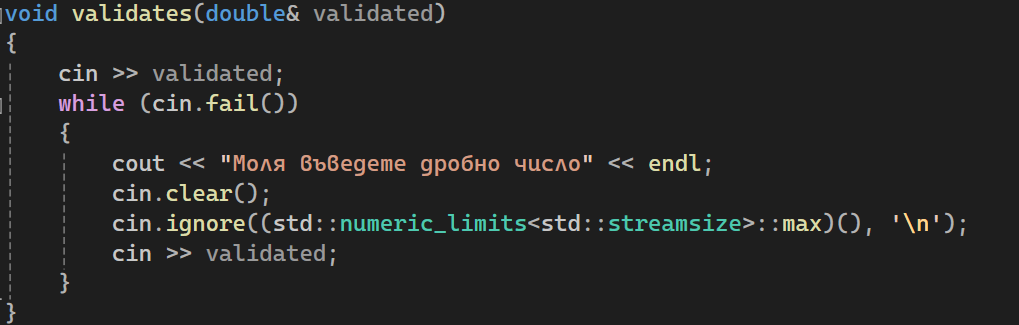
Функция, с която е реализиран алгоритъма

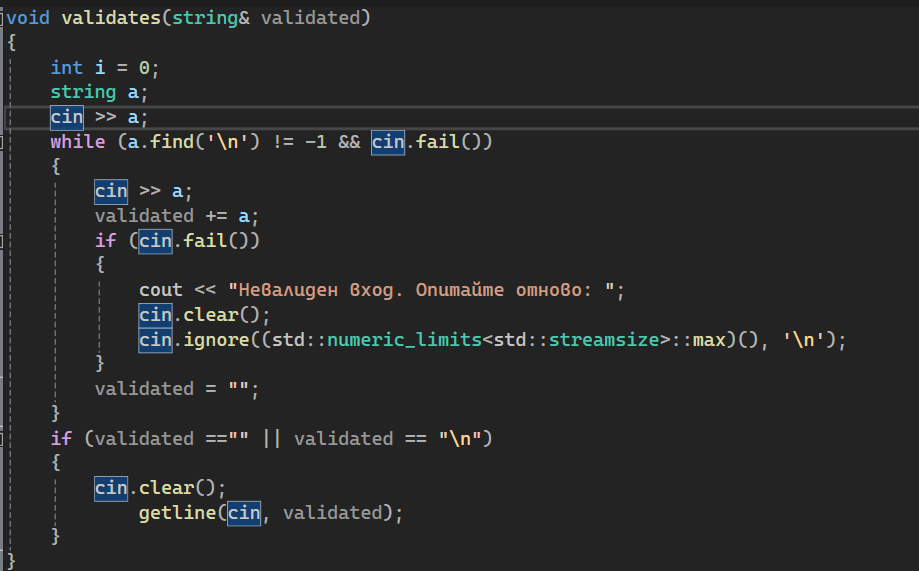


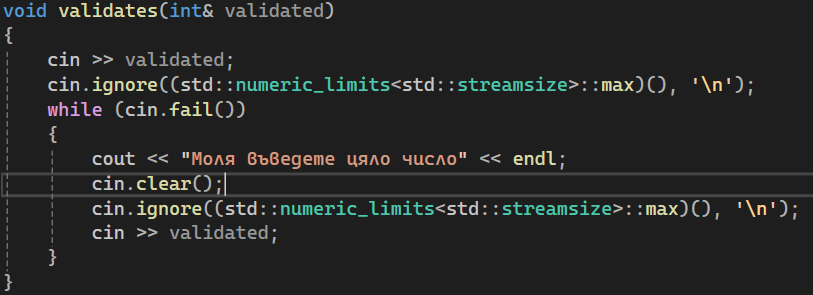


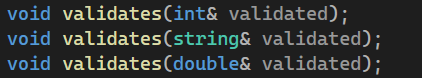
Функцията създава нов апартамент със уникално ID, което се генерира от функцията genID с блок схема:

#### А данните се валидират с функциите Validates, ValidateNeighbourhood, ValidateStatus





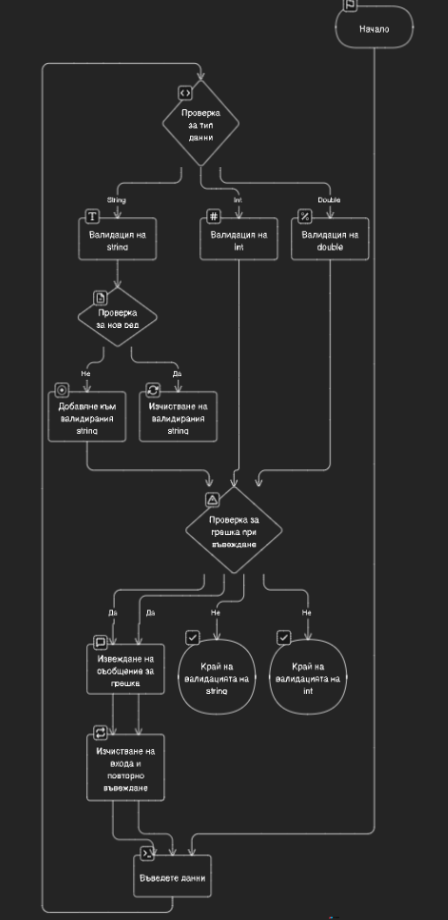




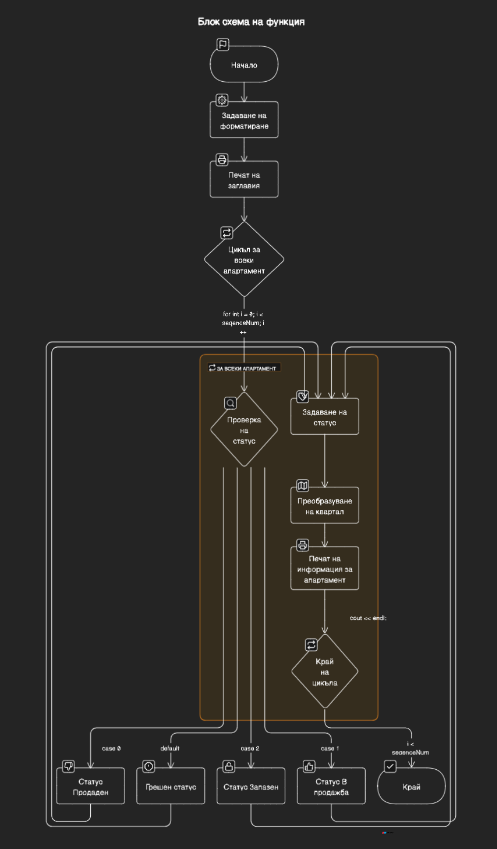
Фигура 1Прототипи

Която има overload със string, с int и с double

Блок схема на overloading-а на validates

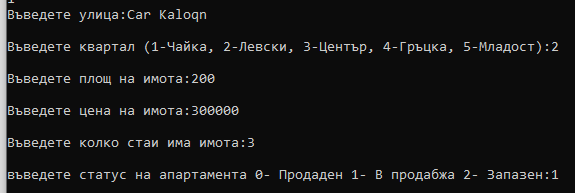
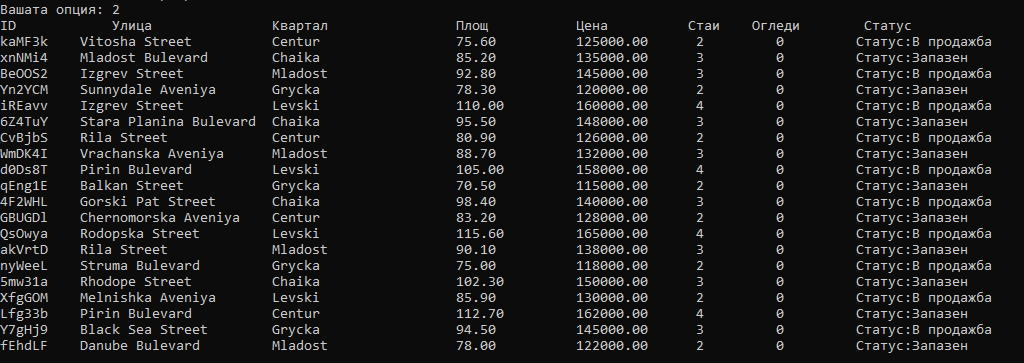
`

ValidateStatus



#### Входни данни на функцията

Примерни входни данни:



Car Kaloqn отива в пример.street 2 отива в пример.neighbourhood

200 отива в пример.area 300000 отива в пример.price 3 отива в пример.rooms

1 отива в пример.status

Реализация на условие C

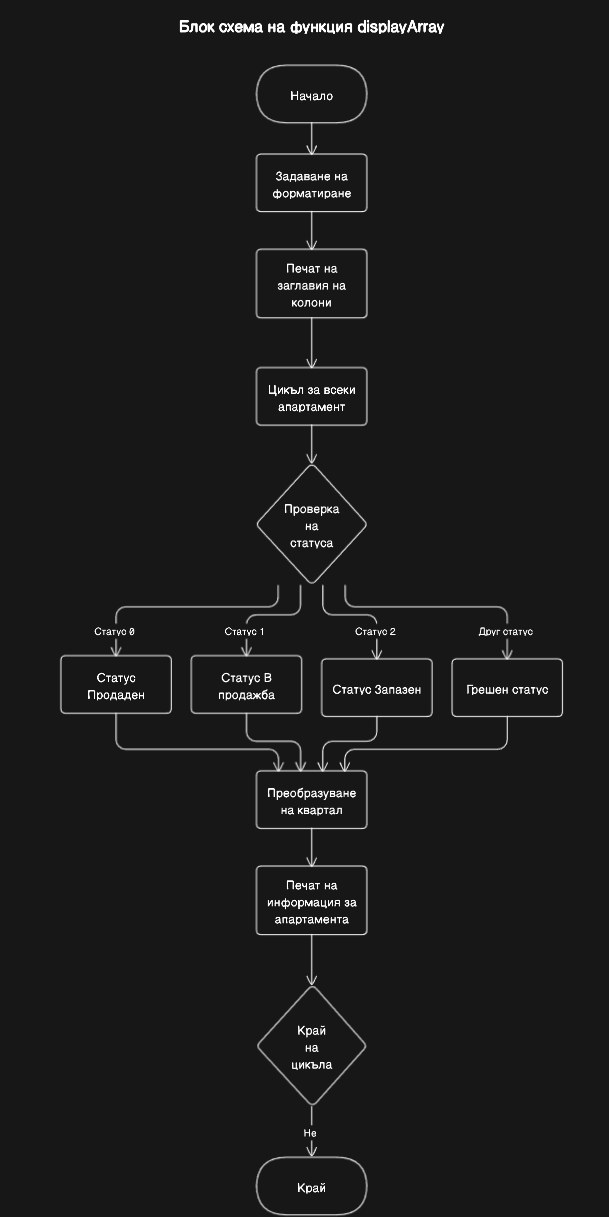
Анализ на алгоритъма, който трябва да се реализира

Фунцкията ще приеме главния масив и числото count, което показва колко записа има в масива. Ще има един for цикъл, който изпечатва всяка променлива във структурата и този масив ще се върти до стойноста на числото count. Във Функцията displayArray се съдържа и фунцкията convertNeighbourhood, която взима целочислена стойност от структурата (от 1 до 5) и я превръща в отговарящия на числото квартал.

Моля, опишете.

Блок схема на алгоритъма



Тук се представя функцията, с която се реализира алгоритъма с нейния прототип. Описва се как трябва да се използва и какво се случва след нейната употреба.

Функция, с която е реализиран алгоритъма



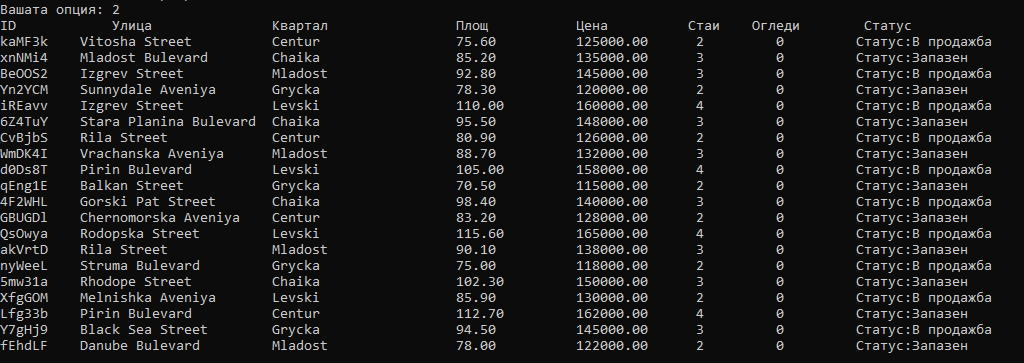




Функцията взима всеки един елемент от подадения масив и ги изпечатва на конзолата по удобен за потребителя начин.

#### Входни данни на функцията

Примерни входни данни:

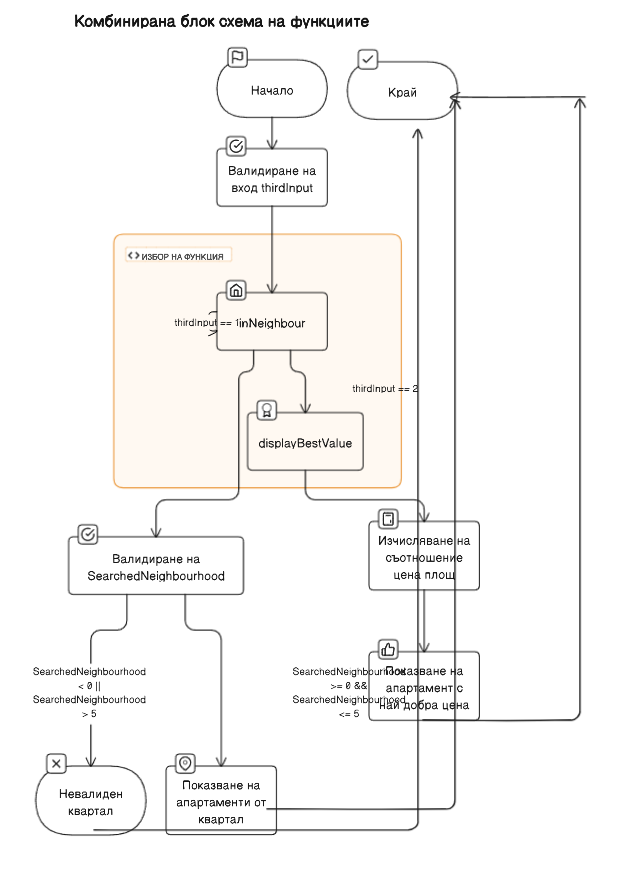


Реализация на условие D

Трябва да се направи switch case, който да действа като подменю и от него да се избира, кой от двата варианта потребителя иска да ползва. При двата случая се викат различни функции. И в двете функции ще се върти по един цикъл, който в единия случай ще намира апартамента с най-ниска цена на квадратен метър, а в другия случай ще търси апартаменти в един и същи квартал

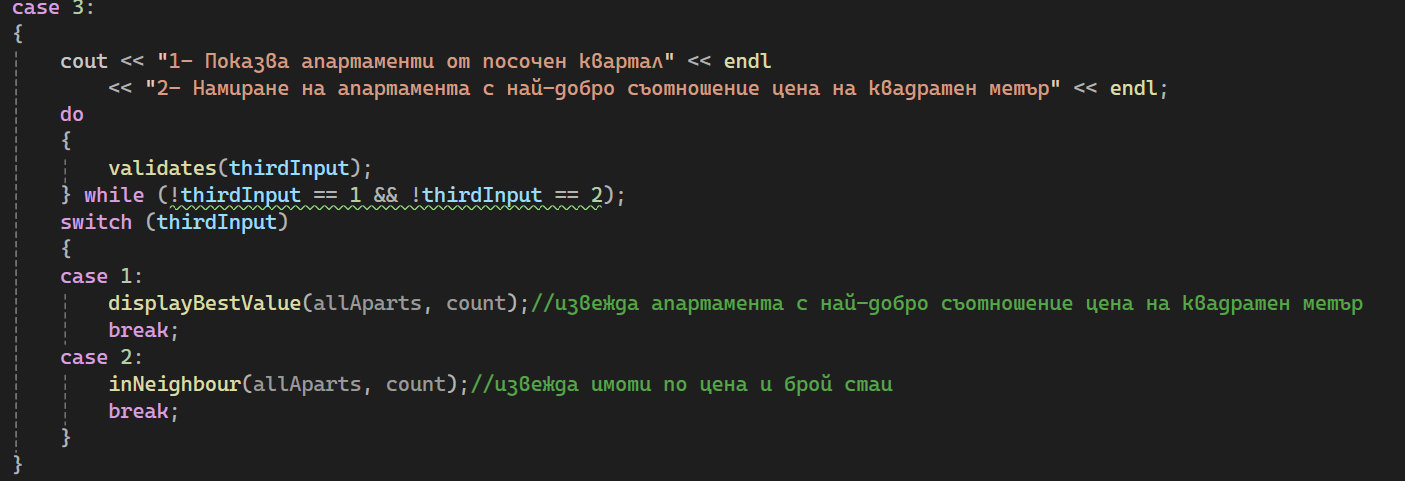
Блок схема на алгоритъма

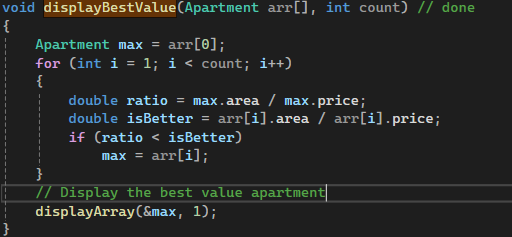
Обща схема за цялото условие

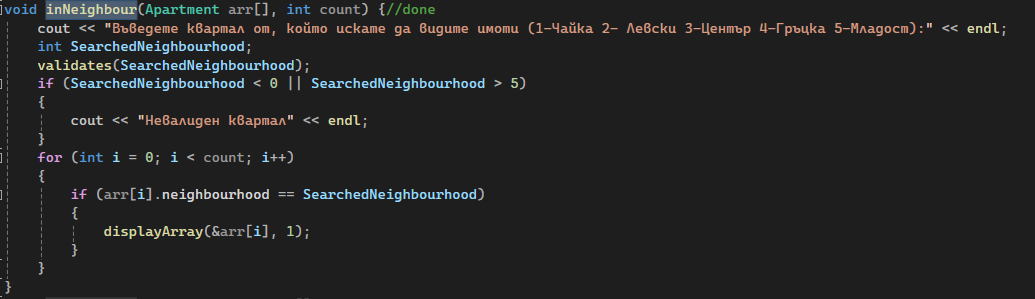


Функция, с която е реализиран алгоритъма

Тук се представя функцията, с която се реализира алгоритъма с нейния прототип. Описва се как трябва да се използва и какво се случва след нейната употреба.



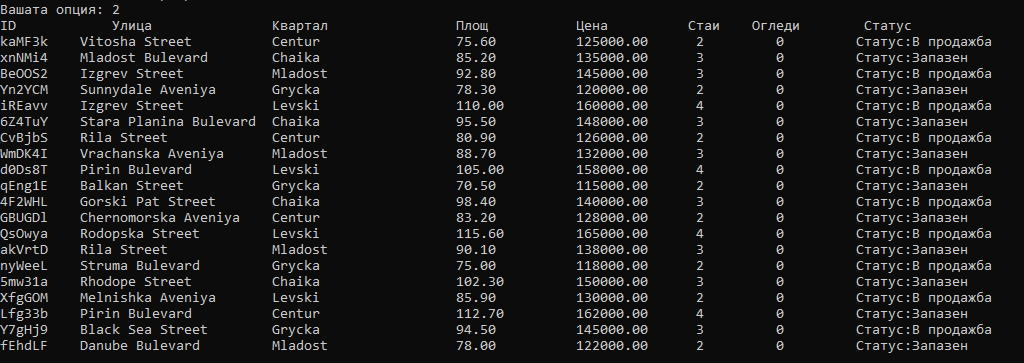


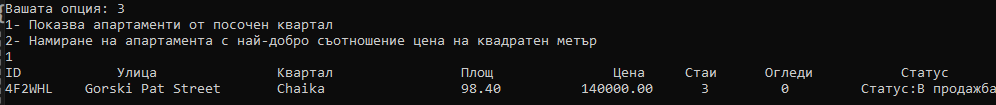


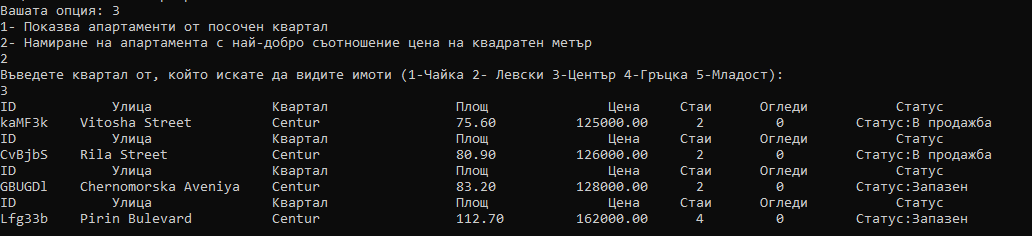




#### Входни и изходни данни на функцията







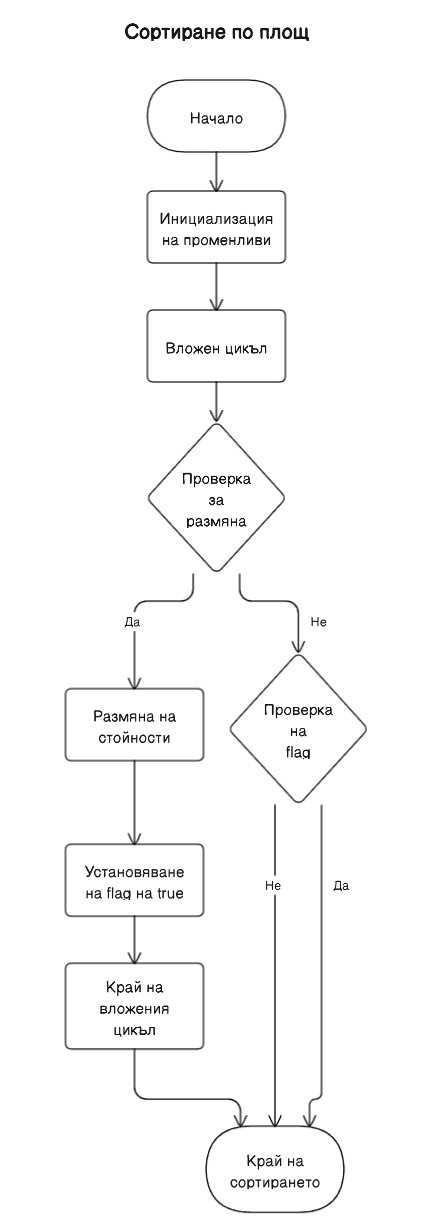
#### Изходни данни на функцията или данни, които се извеждат

Реализация на условие Е

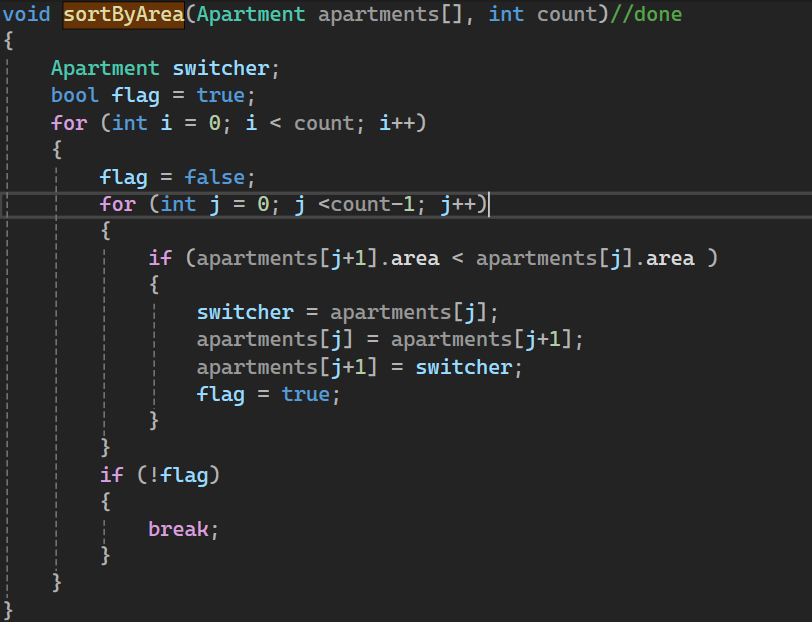
Анализ на алгоритъма, който трябва да се реализира

За да се реализира това условие ще се направи buble sort алгоритъм, който ще подрежда апартаментите в масива според тяхната площ

Блок схема на алгоритъма



Функция с която е реализиран алгоритъма





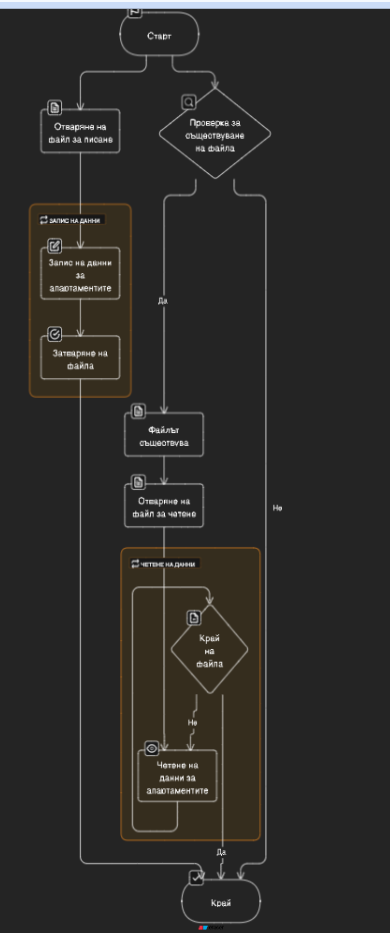
Функцията не показва нищо на екрана, но сортира елементите по площ.

Реализация на условие F

За да се реализира четене и писане от файл, трябва четенето да се осъществява по същия начин и последователност, както писането. Също така, защото програмата е реализирана със string а не със cstring, за да се прочете или запише в бинарен файл първо стринга трябва да бъде конвертиран в cstring, защото така ще знаем дължината му. А при прочитането първо ще прочитаме един integer, който ще показва колко символа трябва да се прочетат. Заради реализацията със string също така не е възможно да се чете цяла структура наведнъж.

Блок схема на алгоритъма

Обща схема за четене и писане от бинарен файл



Функция с която е реализиран алгоритъма

#### 

#### 



Чрез тези две функции данните се записват и четат от бинарен файл. За ID се четат и пишат винаги 7 байта защото то е с фиксирана дължина 6 символа + 1 терминатор.

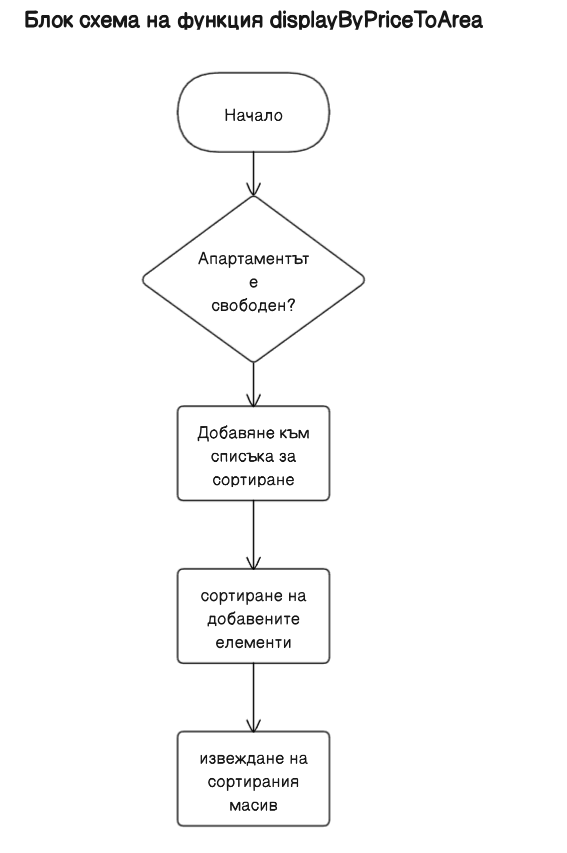
Реализация на условие G

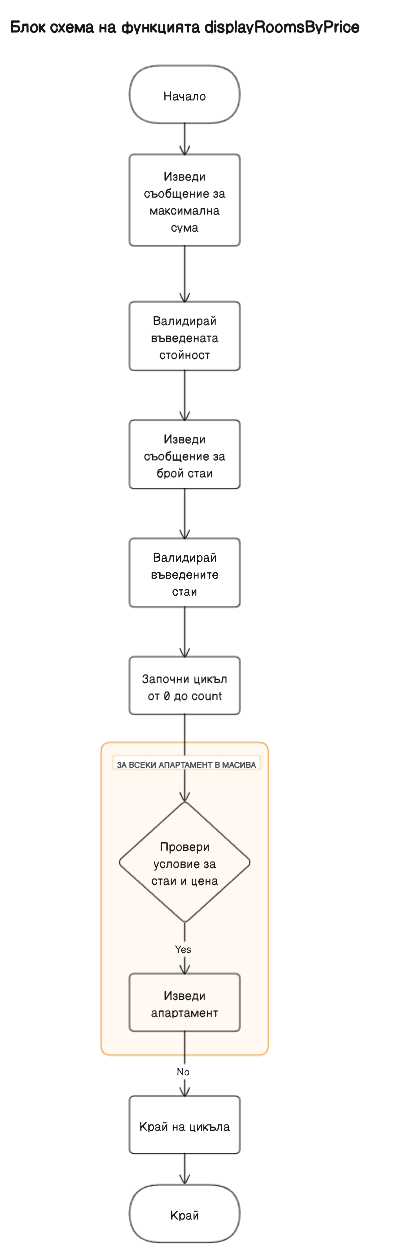
Първо трябва да се направи подменю от, което ще се избира между две функции, което се прави със switch case. За реализирането на извеждане на имотите в продажба, подредени възходящо по цена на квадратен метър

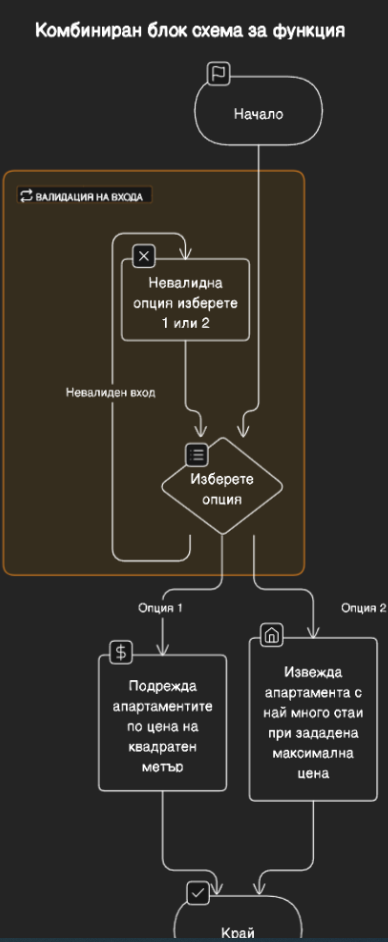
Първо всички имоти в продажба трябва да бъдат записани в нов временен масив

Второ временния масив трябва да бъде сортиран(buble sort в случая) по цена /площ, а след това се вика функцията displayArray за да отпечата масива на концолата, а за функцията displayRoomsByPrice се въвежда едно число за максимална сума и едно число, за колко стаен да е апартамента. След това се върти един цикъл, който проверява тези условия и, ако апартамента отговаря на условията го отпечатва.

Блок схема на алгоритъма

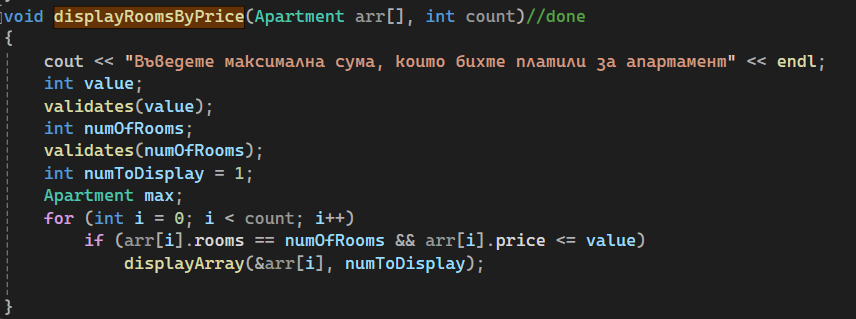


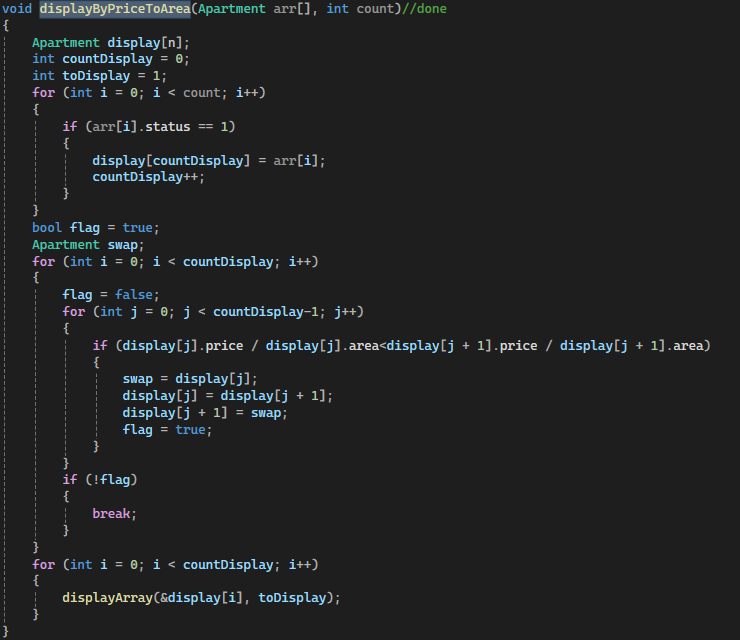




Функция с която е реализиран алгоритъма

#### 

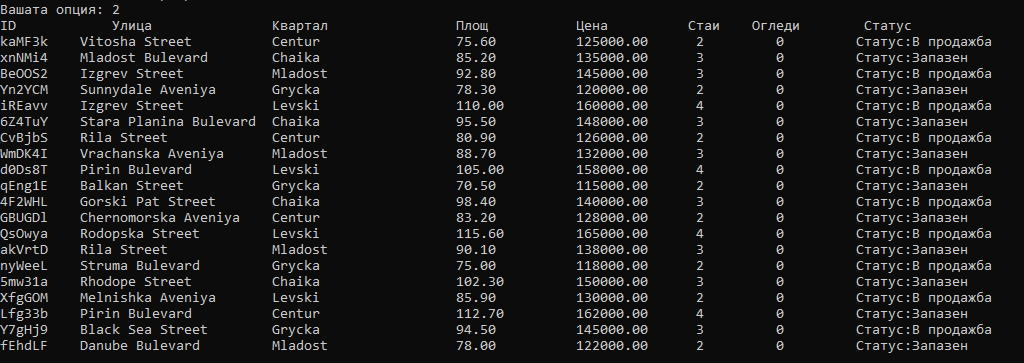


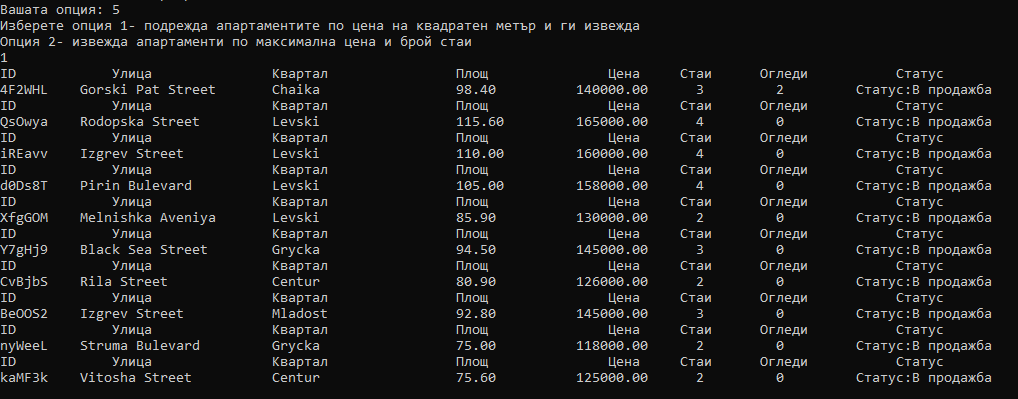


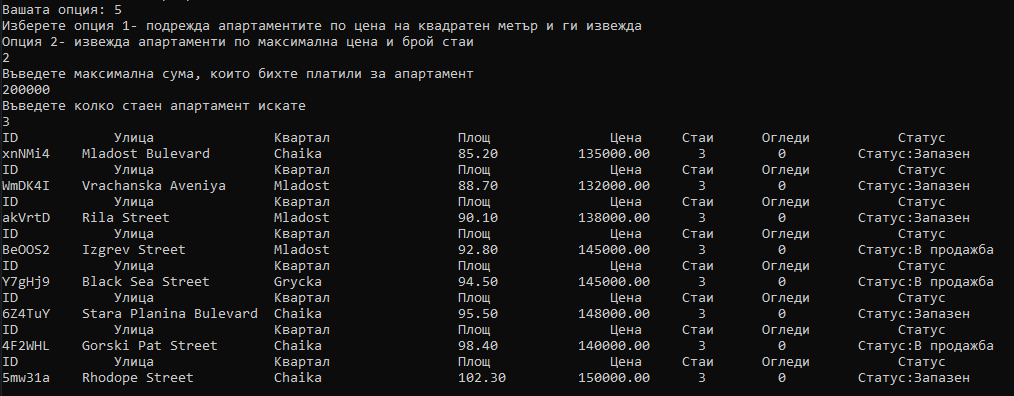




#### Входни и изходни данни на функцията





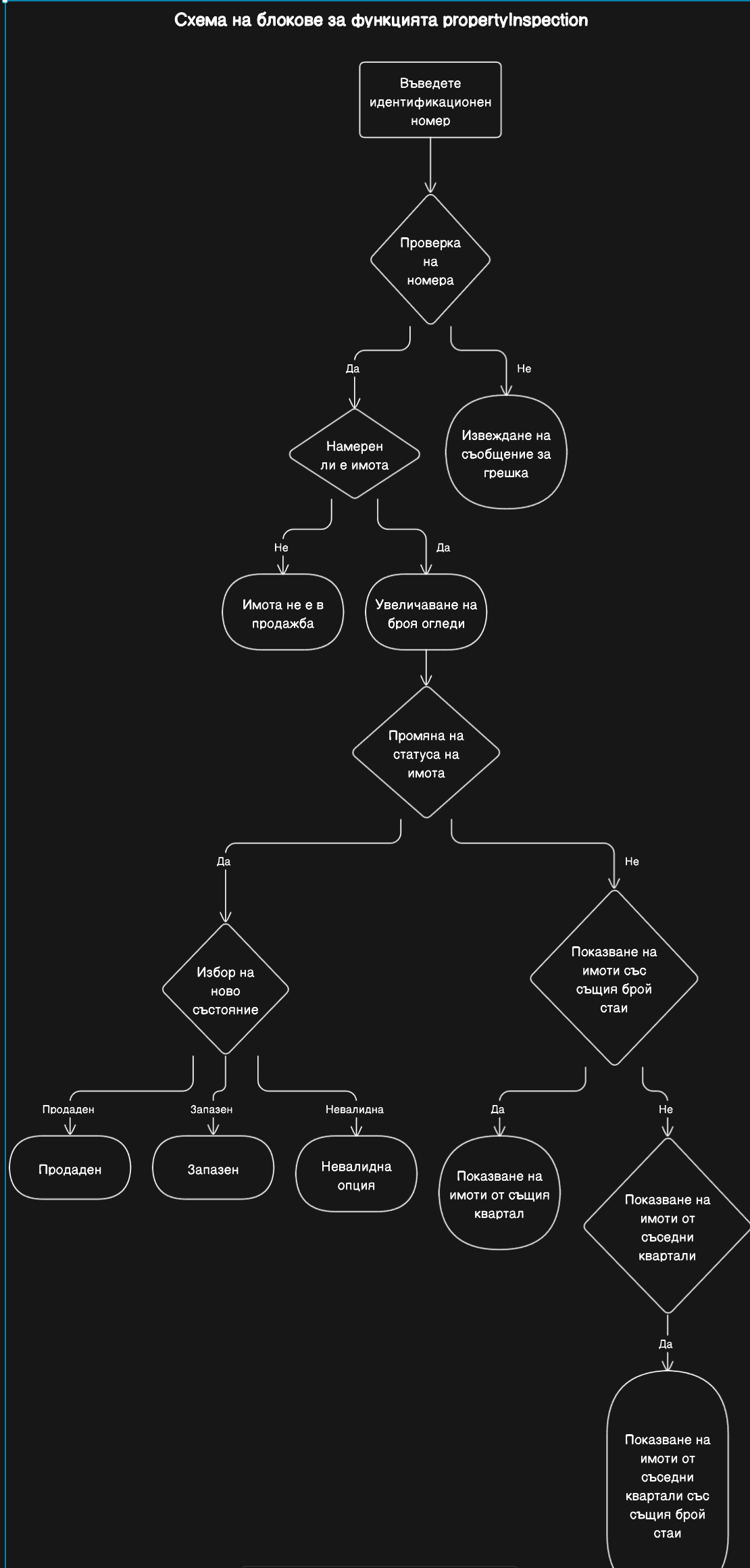


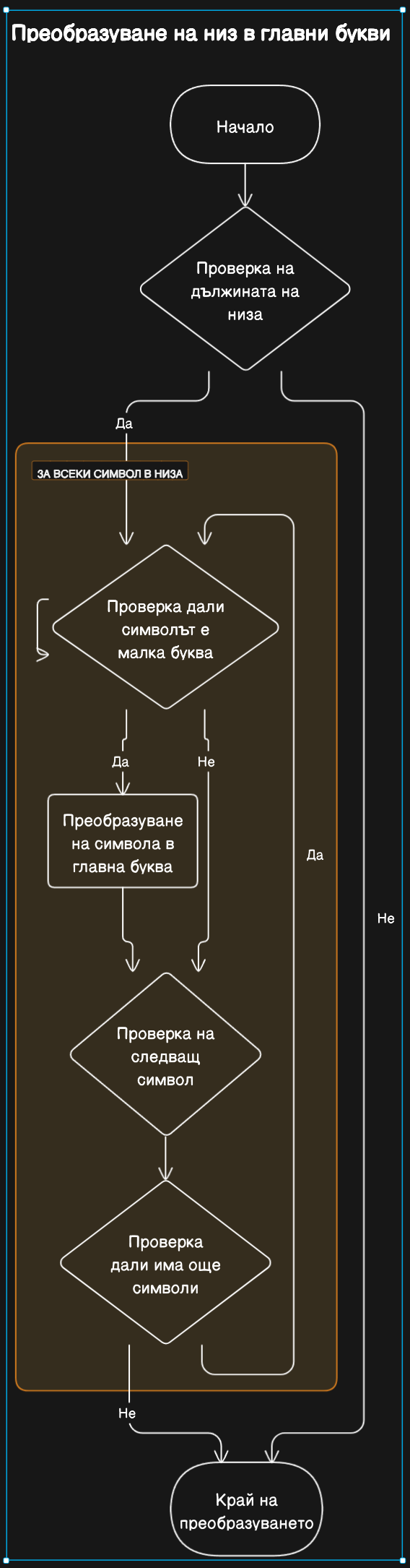
Реализация на условие … - допълнение Второ условие - H

Анализ на алгоритъма, който трябва да се реализира

За да се реализира функцията трябва да се сравни четения от конзолата идентификацнионен номер и този от масива, но за да се улесни работата на потребителя е имплементирана функция, която първо прави и двата на главни букви. Ако номерата съвпадат се проверява дали имота е в продажба, ако е в продажба се пита потребителя как иска да бъде променен статуса на апартамента. Ако потребителя не иска да промени статуса на апартамента, тогава се пита дали иска да види апартаменти от същия квартал със същия брой стаи, или иска да види апартаменти от съседен квартал със същия брой стаи. Търсенето в съседни квартали е реализирано като се проверява дали числото на квартала на апартамента от масива е по-голямо или по-малко с едно от числото на квартала на селектирания апартамент. Ако е се извежда на екрана, ако не е не се. А търсенето в съседни квартали е направено чрез сравнение, дали селектирания имот има еднакъв квартал и брой стаи, както този от масива и ако да се изписва на екрана. Това се повтаря за всеки един имот от масива.

Блок схема на алгоритъма

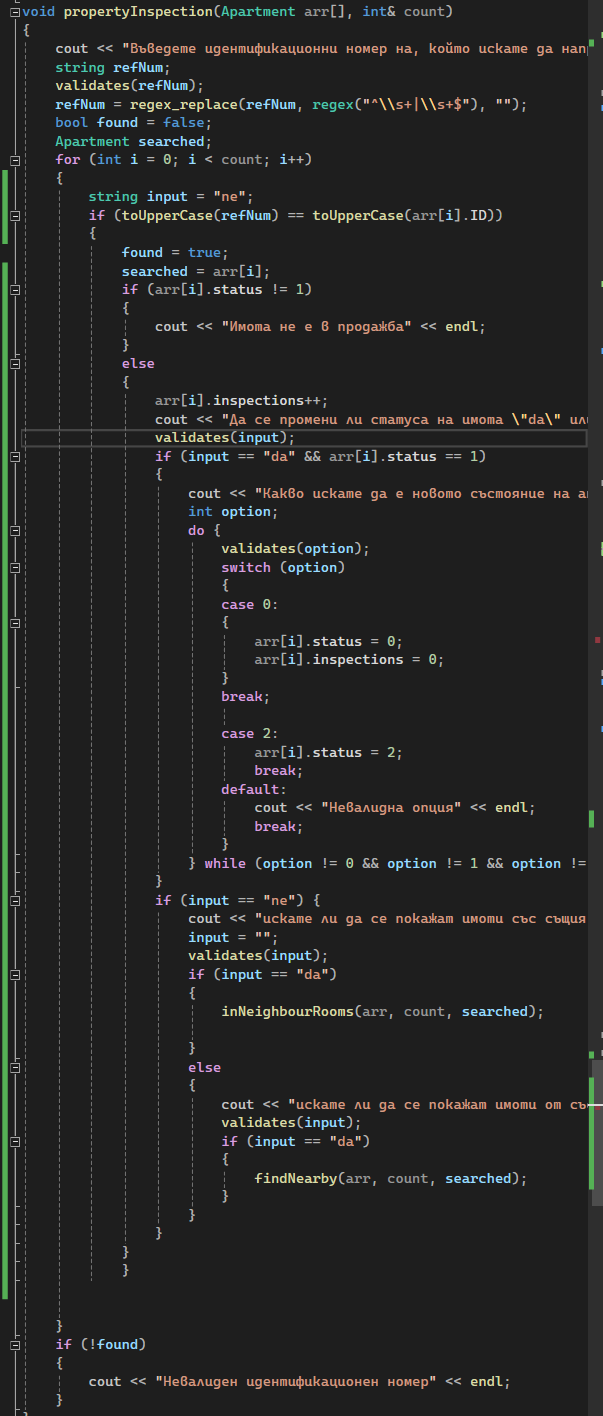


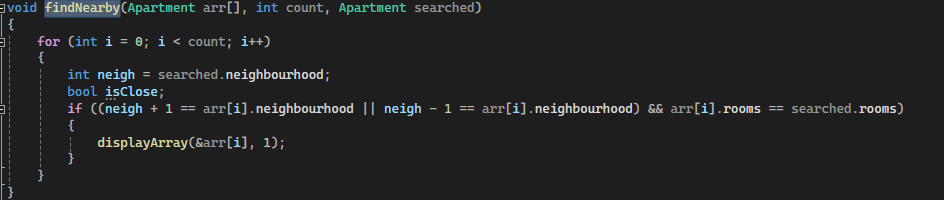


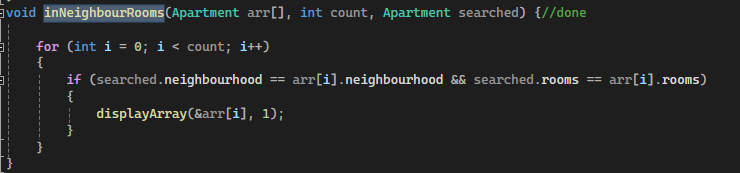




Функция с която е реализиран алгоритъма





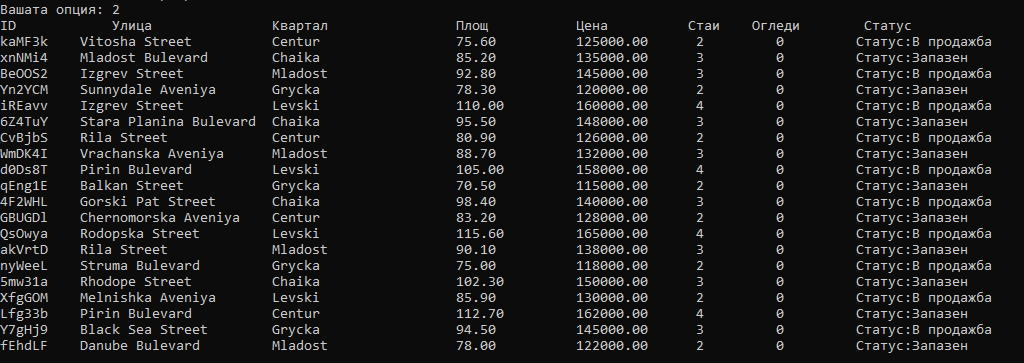








#### Входни данни на функцията



#### 

#### За улесняването на потребителя е използвана функцията regex\_replace(), която се намира в библиотеката regex

refNum = regex\_replace(refNum, regex("^\\s+|\\s+$"), "");

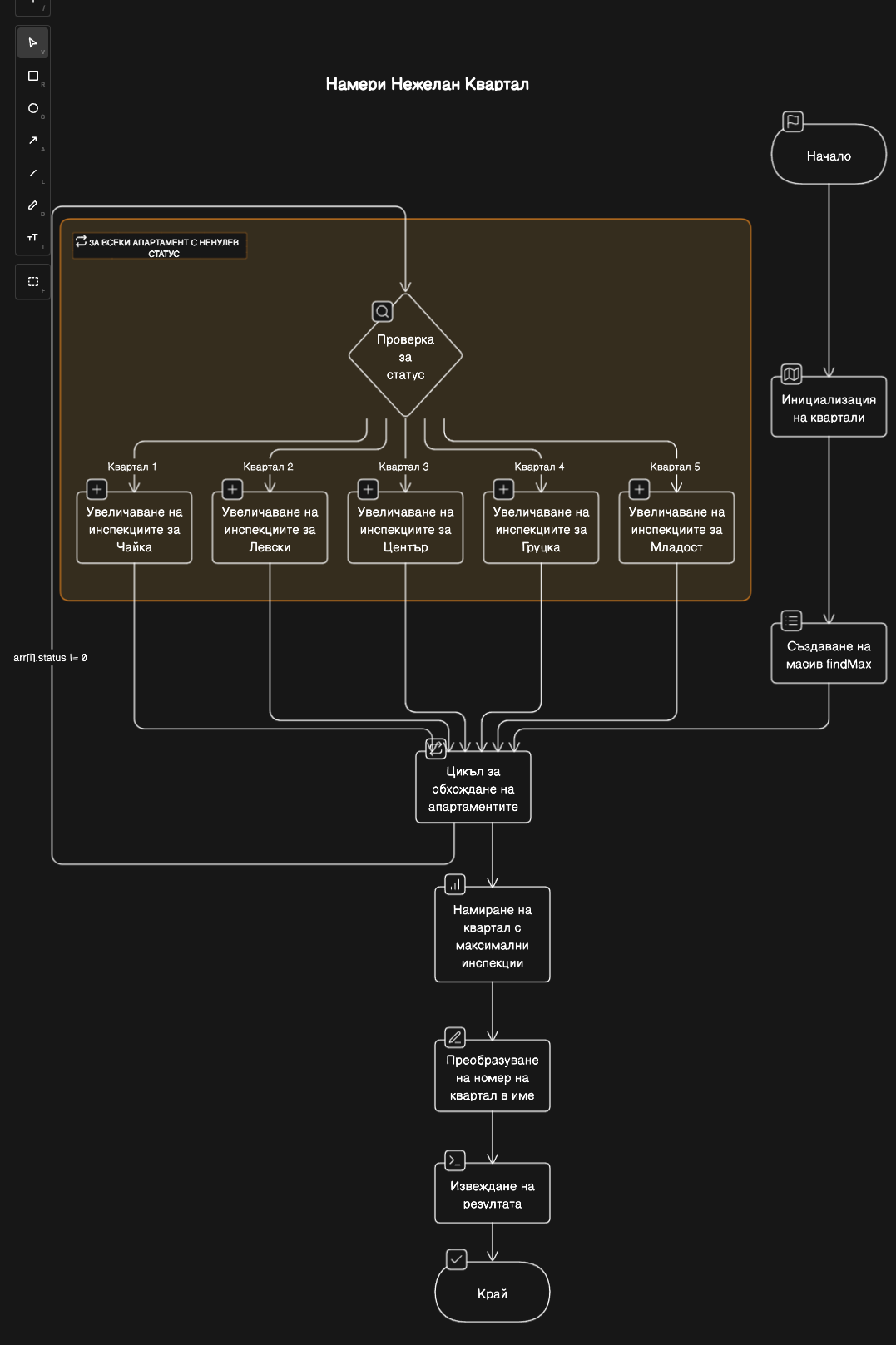
въпросния ред маха нежеланите спейсове, табулации и нови редове отляво и отдясно. Може да се прочете повече на сайта regex101.com

Реализация на условие I

Анализ на алгоритъма, който трябва да се реализира

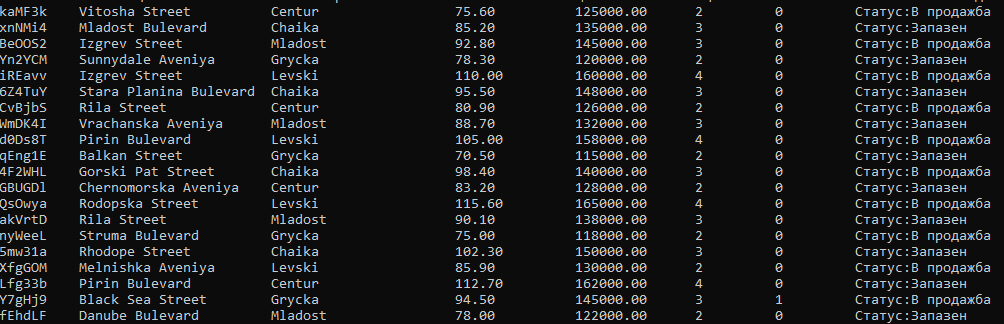
Създава се масив с пет апартамента, които репрезентират всеки квартал.

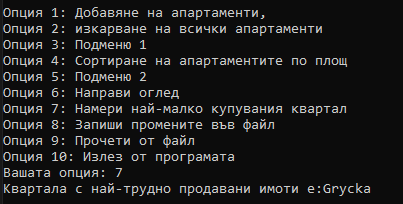
След това се върти един for цикъл, до count(броя на записите в главния масив) и се проверява всеки един апартамент от кой квартал и, колко огледа има. Броя огледи се прибавят към апартамента репрезентиращ съответния квартал.

Функция с която е реализиран алгоритъма

#### 

#### Входни данни на функцията



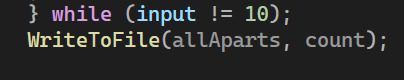


Тук се представят входните данни на функцията: какви са, защо са избрани точно те, за какво ще се използват.

#### Изходни данни на функцията или данни, които се извеждат

Реализация на условие J

Данните се записват във файла при излизане от програмата чрез опция 13



Упътване за употреба

Програмата е направена да извежда част от информацията на български, но тя няма способността да взима информация на кирилица, така че потребителите трябва да използват латиница, когато попълват данните. Не забравяйте, че ако излезете от програмата чрез хикса, данните, които сте въвели няма да бъдат записани във файла. Инструкции за ползване на мeнюто:

(Опция 1 – добави апартаменти, Опция 2- Изпечатай всички апартаменти, Опция 3 -Меню 1 Опция 4- Сортира по площ възходящо всички апартаменти без да ги изпечатва, опция 5- Меню2, Опция 6 -Правене на оглед на апартамент Опция 7- Намиране на квартала с апартаменти с най-много огледи без продабжа, Опция 8 – записване на масива във файл, Опция 9- четене от файл, Опция 10 – Изход от програмата)