

MINISTERUL EDUCAȚIEI, CULTURII ȘI CERCETĂRII

AL REPUBLICII MOLDOVA

Centrul de Excelență în Informatică și Tehnologii Informaționale

RAPORT

LA PRACTICA DE ÎNȚIERE ÎN SPECIALITATE

SPECIALITATEA "Programare și analiza produselor program"

TEMA : "Квартиры"

A elaborat elevul

Ганзюк Александр, P-2114R

Conducătorul practicii

Șarapanovscaia Irina

Chișinău 2022

Содержание

Введение	4
Условие	5
Входные данные	6
Описание подпрограммы:	7
<i>Меню</i>	7
<i>Информация о квартирах</i>	9
<i>Информация об аренде</i>	12
<i>Добавление информации о новой квартире</i>	15
<i>Добавление информации о новой аренде</i>	19
<i>Удаление информации о квартире, код которой введен с клавиатуры</i>	23
<i>Удаление информации об аренде, код которой введен с клавиатуры</i>	27
<i>Редактирование информации о квартире, код которой введен с клавиатуры</i>	29
<i>Редактирование информации об аренде, код которой введен с клавиатуры</i>	35
<i>Создание текстового файла, содержащего информацию об арендах, срок которых истек</i>	41
<i>Список свободных квартир в порядке возрастания количества комнат</i>	45
<i>Квартира, пользующаяся наибольшим спросом</i>	48
<i>Средняя цена всех квартир с указанным количеством комнат</i>	51
Вспомогательные функции	53
<i>Смена цвета</i>	53
<i>Управление мышкой в главном меню</i>	53
<i>Управление мышкой в редактировании квартир</i>	57
<i>Управление мышкой в редактировании аренд</i>	60
<i>Управление мышкой в главном меню</i>	63
<i>Считывание информации о квартирах</i>	65
<i>Считывание информации об арендах</i>	66
<i>Перезапись файла квартир</i>	67
<i>Перезапись файла аренд</i>	67
Microsoft Excel:	68
Apartment	68
Rent.....	77
Статистика	79
Графическое представление	79
ComboBox	80
Список с картинками	81
Вывод	82
Библиография	83

Приложение.....	84
1) <i>Apartments.h</i>	84
2) <i>Apartments.cpp</i>	85
3) <i>main.cpp</i>	127

Введение.

Период практики: 3 недели.

С 30 мая 2022 по 17 июня 2022.

Время практики с 8:00 по 10:50.

Цели практики введения в специальность:

- ✓ укрепление теоретических знаний и углубление практических навыков, полученных студентами в период изучения модулей;
- ✓ применение технологий разработки программных продуктов, электронных таблиц, текстовых документов и электронных презентаций;
- ✓ развитие навыков самостоятельной и командной работы;

В соответствии с общими целями, практика введения в специальность в соответствии с принципом «Разработка алгоритмов, предназначенных для решения сложных комплексных задач». В результате представления таких алгоритмов с помощью языка программирования высокого уровня обычно получают большие компьютеризированные модели, основанные на широком использовании различных типов структурированных данных: массивов, структур, строк, файлов и т.д.

Практика проходила в нашем колледже. Начиная с 30 мая до 17 июня мы приходили в колледж в 8:00. Мне очень понравился период практики и представленные задания, было интересно их выполнять.

Во время практики я использовал такие инструменты как: Code Blocks, Microsoft Excel, Microsoft Word и Microsoft PowerPoint.

Условие

Информация о квартирах (*код квартиры, район, адрес, количество комнат, этаж, цена*) содержится в текстовом файле **Apartment.txt**. Каждая строка тестового файла соответствует информации об отдельной квартире и наоборот.

Информация об аренде квартир (*код квартиры, фамилия и имя арендатора, дата аренды, срок аренды*) содержится в текстовом файле **Rent.txt**. Каждая строка тестового файла соответствует информации об отдельной аренде и наоборот.

C++

Создайте консольное приложение, содержащее меню и функции, которые по выбору пользователя выполняют следующие действия:

1. Вывод на экран информации о квартирах.
2. Вывод на экран информации об аренде.
3. Добавление информации о новой квартире.
4. Добавление информации о новой аренде.
5. Удаление информации о квартире, код которой введен с клавиатуры.
6. Удаление информации об аренде порядковый номер которой вводится с клавиатуры.
7. Редактирование информации о квартире, код которой введен с клавиатуры.
8. Редактирование информации об аренде порядковый номер которой вводится с клавиатуры.
9. Создание текстового файла **Expire.txt**, содержащего информацию об арендах, срок которых истек на дату, введенную с клавиатуры.
10. Вывод на экран списка свободных квартир в порядке возрастания количества комнат.
11. Вывод на экран информации о квартире, пользующейся наибольшим спросом.
12. Вывод на экран средней цены всех квартир с указанным количеством комнат.

MS Excel

Выполните следующие задания в приложении MS Excel:

1. Создайте файл **Agency**. Выполните импорт данных из текстовых файлов **Apartment.txt** и **Rent.txt** в отдельные листы с соответствующим названием.
2. Добавьте для каждой таблицы название и озаглавьте соответствующие колонки.
3. Отформатируйте таблицы: установите границы, цвет, размер, тип шрифта и т.д.
4. Значение *код квартиры* в таблице **Аренда** должно выбираться из раскрывающегося списка. Дата, месяц и год аренды объедините в одну ячейку
5. В таблице **Квартиры** добавьте еще одну колонку, содержащую фото каждой квартиры. Для названия таблицы **Аренды** добавьте комментарий, содержащий логотип риэлтерской фирмы.
6. Для таблицы **Аренда** добавьте новую колонку, содержащую сумму к оплате и новую строку, содержащую общую сумму всех аренд.
7. Вычислите количество квартир для каждого сектора и их среднюю цену. Соответствующие вычисления сохраните в листе **Статистика**.
8. Представьте графически количество 1, 2, 3, 4 –х комнатных квартирах.
9. Выведите информацию о всех арендах квартир, код которой выбирается из раскрывающегося списка (*ComboBox*).
10. Выведите фото квартиры, код которой выбирается из раскрывающегося списка

MS Word и MS Power Point

Создайте отчет по практике в приложении MS WORD согласно установленным требованиям.

Оформите презентацию в приложении MS POWER POINT о выполненной работе согласно установленным требованиям и рекомендациям.

Входные данные

Apartment.txt

Apartment.txt – Блокнот				
Файл	Правка	Формат	Вид	Справка
1	Centru	Ismail-26	2 1	45000
2	Buiucani	Milano-12	3 3	74000
3	Rishkanovka	Andrei_Lupan-12	2 2	47000
4	Botanica	Sarmizegetusa-24	1 2	20000
5	Rishkanovka	Doina-3	2 9	31000
6	Botanica	Nicolae_Zelinski	4 11	89000
7	Cekani	Sargidava-4	1 7	25000
8	Centru	Bucuresti-3	2 4	38000
9	Rishkanovka	Doina-10	3 1	75000
10	Botanica	Sarmizegetusa-11	2 14	47000
11	Cekani	Industrialala-16	1 9	25000
12	Buiucani	Suciava-4	3 7	59000
13	Botanica	Burebista	2 2	46000
14	Cekani	Otovasca-16	1 15	27600
15	Centru	Bucuresti-72	2 4	41820
16	Buiucani	Ion_Creanga-11	5 7	114200
17	Buiucani	Milano-55	3 2	76200
18	Cekani	Butuceni-18	4 1	96400
19	Rishkanovka	Criuleni-1	1 8	29600
20	Botanica	Padurii-44	2 7	52000
21	Rishkanovka	Doina-45	4 6	76900
22	Centru	Tigina-25	1 5	29650
23	Buiucani	Sucevita-22	1 4	35600
24	Rishkanovka	Timisoara-18	2 11	49900
25	Centru	Ismail-47	3 12	67500
26	Buiucani	Alba_Iulia-22	4 15	89600
27	Botanica	Gradina_Botanica-19	3 8	72000
28	Centru	Columna-13	5 4	156000
29	Cekani	Otovasca-19	2 8	41600
30	Rishkanovka	Ceucari-22	1 14	29900
31	Botanica	Prigoreni-12	2 3	41900
32	Centru	Bucuresti-11	4 11	91200
33	Centru	Tigina-56	3 15	35200
34	Cekani	Maria_Dragan-8	1 3	39600
35	Buiucani	Industrialala-48	2 4	69000
36	Rishkanovka	Andrei_Lupan-72	2 8	72300
37	Centru	Alexei_Mateevici-81	1 9	31600
38	Botanica	Simeon_Murafa-12	2 7	54200
39	Cekani	Mesterul_Manole-20	4 5	99600
40	Centru	Bucuresti-17	5 4	105400
41	Buiucani	Industrialala-55	1 6	29900
42	Centru	Alexei_Sciusev-43	5 2	142000
43	Buiucani	George_Cosbuc	2 1	57600
44	Rishkanovka	Alecu_Russo-115	3 1	86200
45	Centru	Columna-54	4 2	97500
46	Cekani	Sargidava-19	1 9	36100
47	Botanica	Leuseni-4	2 13	61000
48	Cekani	Ilie_Replin-2	3 14	78900
49	Rishkanovka	Petru_Zadnipru-39	2 15	54100
50	Centru	Serghei_Lazo-14	4 11	98600

Скриншот 1

Rent.txt

Rent.txt – Блокнот				
Файл	Правка	Формат	Вид	Справка
1	Martinez	Lorenzo	30 04	2022 2
2	Patterson	Camden	13 07	2020 3
3	Scott	Quenten	04 12	2015 1
4	Brown	Vidal	25 07	2017 4
5	Gonzalez	Duke	18 11	2020 5
3	Howard	Henrik	13 05	2022 2
7	Promise	Alexander	06 11	2021 3
8	Xenos	Howard	06 08	2015 1
6	Williams	Natalie	09 04	2019 2
9	Irving	James	01 09	2020 1
3	Davis	Dean	10 03	2017 2
11	Johnson	Isaias	05 01	2018 3
10	Ferguson	Harold	19 11	2021 2
11	Martinez	Nikko	11 12	2022 4
13	Johnson	Lorenzo	15 04	2022 5
14	Gray	Marcus	01 05	2022 4
12	Jones	Karen	02 08	2020 1
15	Davis	Gordon	08 10	2018 3
16	Allen	Torin	12 01	2021 2
17	Brooks	Thaddeus	28 09	2019 1
18	King	Briggs	04 12	2018 1
19	Hernandez	Eliau	14 01	2021 2
20	Nelson	Ezekiel	14 09	2021 2
21	Hernandez	Diego	09 02	2022 4
22	Moore	Malachi	09 09	2015 5
23	Scott	Liam	05 06	2020 2
24	Henderson	Zander	07 01	2015 1
25	Young	Gentry	16 09	2020 2
26	Perez	Nick	26 01	2021 4
27	Walker	Josiah	15 02	2022 4
28	Rogers	Qadry	19 08	2022 2
29	Barnes	Uziel	15 09	2022 1
24	Thomas	Jase	03 09	2018 6
31	Harris	Taj	22 04	2018 2
32	Powell	Keaton	26 12	2019 1
33	Hall	Enzo	19 10	2015 2
30	Ross	Miriam	22 12	2010 5
34	Hernandez	Trenton	11 02	2015 3
35	Hayes	Lewis	24 09	2020 3
36	Jenkins	Ruben	17 02	2015 2
37	Wright	Kenneth	29 04	2022 1
33	Young	Omar	29 01	2018 2
38	Hill	Ricky	30 05	2020 3
39	Taylor	Abel	10 12	2018 3
40	Torres	Ian	12 02	2018 4
41	Ward	Bennett	31 12	2018 5
42	Coleman	Soren	13 01	2018 5
33	Foster	Juan	30 11	2021 2
44	Johnson	Lorenzo	01 05	2021 1
45	Gray	Giovanni	06 05	2015 6
43	Vasquez	Jesus	15 01	2022 1
33	Rodriguez	Felipe	12 07	2020 1
47	Howard	Finn	06 02	2015 3
48	Harris	Urias	11 05	2018 1
49	Adams	Keith	05 11	2020 1
50	Lopez	Tripp	21 07	2022 4

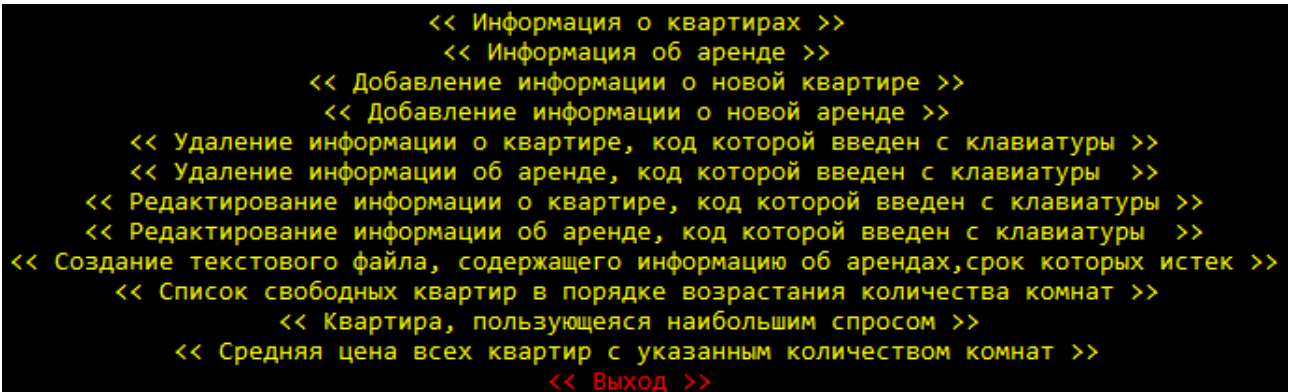
Скриншот 2

Описание подпрограммы:

Меню

Описание: в моем проекте меню используется для того, чтобы взаимодействовать с функциями. Моё меню предоставляет пользователю много возможностей, такие как: вывод информации о квартирах и арендах, добавление информации о новой квартире и аренде, удаление информации о квартире и аренде, редактирование информации о квартире и аренде, создание текстового файла, содержащего информацию об арендах, срок которых истек, список свободных квартир в порядке возрастания количества комнат, вывод квартиры, пользующейся наибольшим спросом, вывод средней цены всех квартир с указанным количеством комнат и выход. Для управления в меню используется мышка.

Меню



```
<< Информация о квартирах >>  
  << Информация об аренде >>  
    << Добавление информации о новой квартире >>  
    << Добавление информации о новой аренде >>  
      << Удаление информации о квартире, код которой введен с клавиатуры >>  
      << Удаление информации об аренде, код которой введен с клавиатуры >>  
        << Редактирование информации о квартире, код которой введен с клавиатуры >>  
        << Редактирование информации об аренде, код которой введен с клавиатуры >>  
          << Создание текстового файла, содержащего информацию об арендах, срок которых истек >>  
          << Список свободных квартир в порядке возрастания количества комнат >>  
            << Квартира, пользующаяся наибольшим спросом >>  
            << Средняя цена всех квартир с указанным количеством комнат >>  
              << Выход >>
```

Скриншот 3

Фрагмент кода

```
void menu(){ // Функция для вывода меню
    setlocale(LC_ALL, "Russian"); // выбор русского языка в консоли
    setcolor(14,0); // Смена цвета на жёлтый
    cout<<"\n\n";
    cout<<internal<<setw(73)<<"<< Информация о квартирах >>"<<endl;
    cout<<setw(72)<<"<< Информация об аренде >>"<<endl;
    cout<<setw(81)<<"<< Добавление информации о новой квартире >>"<<endl;
    cout<<setw(80)<<"<< Добавление информации о новой аренде >>"<<endl;
    cout<<setw(94)<<"<< Удаление информации о квартире, код которой введен с клавиатуры
>>"<<endl;
    cout<<setw(94)<<"<< Удаление информации об аренде, код которой введен с клавиатуры
>>"<<endl;
    cout<<setw(97)<<"<< Редактирование информации о квартире, код которой введен с
клавиатуры >>"<<endl;
    cout<<setw(97)<<"<< Редактирование информации об аренде, код которой введен с
клавиатуры >>"<<endl;
    cout<<setw(102)<<"<< Создание текстового файла, содержащего информацию об арендах,
срок которых истек >>"<<endl;
    cout<<setw(94)<<"<< Список свободных квартир в порядке возрастания количества
комнат >>"<<endl;
    cout<<setw(82)<<"<< Квартира, пользующаяся наибольшим спросом >>"<<endl;
    cout<<setw(90)<<"<< Средняя цена всех квартир с указанным количеством комнат
>>"<<endl;
    setcolor(12,0); // Смена цвета на красный
    cout<<setw(64)<<"<< Выход >>"<<endl;
    setcolor(14,0); // Смена цвета на жёлтый
}
```


Информация о квартирах

Текст задания:

Вывод на экран информации о квартирах.

Скриншот вводимых данных:



Apartment.txt – Блокнот

	Файл	Правка	Формат	Вид	Справка
1	Centru	Ismail-26	2	1	45000
2	Buiucani	Milano-12	3	3	74000
3	Rishkanovka	Andrei_Lupan-12	2	2	47000
4	Botanica	Sarmizegetusa-24	1	2	20000
5	Rishkanovka	Doina-3	2	9	31000
6	Botanica	Nicolae_Zelinski	4	11	89000
7	Cekani	Sargidava-4	1	7	25000
8	Centru	Bucuresti-3	2	4	38000
9	Rishkanovka	Doina-10	3	1	75000
10	Botanica	Sarmizegetusa-11	2	14	47000
11	Cekani	Industrialala-16	1	9	25000
12	Buiucani	Suciava-4	3	7	59000
13	Botanica	Burebista	2	2	46000
14	Cekani	Otovasca-16	1	15	27600
15	Centru	Bucuresti-72	2	4	41820
16	Buiucani	Ion_Creanga-11	5	7	114200
17	Buiucani	Milano-55	3	2	76200
18	Cekani	Butuceni-18	4	1	96400
19	Rishkanovka	Criuleni-1	1	8	29600
20	Botanica	Padurii-44	2	7	52000
21	Rishkanovka	Doina-45	4	6	76900
22	Centru	Tigina-25	1	5	29650
23	Buiucani	Sucevita-22	1	4	35600
24	Rishkanovka	Timisoara-18	2	11	49900
25	Centru	Ismail-47	3	12	67500
26	Buiucani	Alba_Iulia-22	4	15	89600
27	Botanica	Gradina_Botanica-19	3	8	72000
28	Centru	Columna-13	5	4	156000
29	Cekani	Otovasca-19	2	8	41600
30	Rishkanovka	Ceucari-22	1	14	29900
31	Botanica	Prigoreni-12	2	3	41900
32	Centru	Bucuresti-11	4	11	91200
33	Centru	Tigina-56	3	15	35200
34	Cekani	Maria_Dragan-8	1	3	39600
35	Buiucani	Industrialala-48	2	4	69000
36	Rishkanovka	Andrei_Lupan-72	2	8	72300
37	Centru	Alexei_Mateevici-81	1	9	31600
38	Botanica	Simeon_Murafa-12	2	7	54200
39	Cekani	Mesterul_Manole-20	4	5	99600
40	Centru	Bucuresti-17	5	4	105400
41	Buiucani	Industrialala-55	1	6	29900
42	Centru	Alexei_Sciusev-43	5	2	142000
43	Buiucani	George_Cosbuc	2	1	57600
44	Rishkanovka	Alecu_Russo-115	3	1	86200
45	Centru	Columna-54	4	2	97500
46	Cekani	Sargidava-19	1	9	36100
47	Botanica	Leuseni-4	2	13	61000
48	Cekani	Ilie_Replin-2	3	14	78900
49	Rishkanovka	Petru_Zadnipru-39	2	15	54100
50	Centru	Serghei_Lazo-14	4	11	98600

Скриншот 4

Скриншот результата на экране:

Код Квартиры	Район	Адрес	Кол-во комнат	Этаж	Цена
1	Centru	Ismail-26	2	1	45000
2	Buiucani	Milano-12	3	3	74000
3	Rishkanovka	Andrei Lupan-12	2	2	47000
4	Botanica	Sarmizegetusa-24	1	2	20000
5	Rishkanovka	Doina-3	2	9	31000
6	Botanica	Nicolae Zelinski	4	11	89000
7	Cekani	Sargidava-4	1	7	25000
8	Centru	Bucuresti-3	2	4	38000
9	Rishkanovka	Doina-10	3	1	75000
10	Botanica	Sarmizegetusa-11	2	14	47000
11	Cekani	Industrialala-16	1	9	25000
12	Buiucani	Suciava-4	3	7	59000
13	Botanica	Burebista	2	2	46000
14	Cekani	Otovasca-16	1	15	27600
15	Centru	Bucuresti-72	2	4	41820
16	Buiucani	Ion Creanga-11	5	7	114200
17	Buiucani	Milano-55	3	2	76200
18	Cekani	Butuceni-18	4	1	96400
19	Rishkanovka	Criuleni-1	1	8	29600
20	Botanica	Padurii-44	2	7	52000
21	Rishkanovka	Doina-45	4	6	76900
22	Centru	Tigina-25	1	5	29650
23	Buiucani	Sucevita-22	1	4	35600
24	Rishkanovka	Timisoara-18	2	11	49900
25	Centru	Ismail-47	3	12	67500
26	Buiucani	Alba Iulia-22	4	15	89600
27	Botanica	Gradina Botanica-19	3	8	72000
28	Centru	Columna-13	5	4	156000
29	Cekani	Otovasca-19	2	8	41600
30	Rishkanovka	Ceucari-22	1	14	29900
31	Botanica	Prigoreni-12	2	3	41900
32	Centru	Bucuresti-11	4	11	91200
33	Centru	Tigina-56	3	15	35200
34	Cekani	Maria Dragan-8	1	3	39600
35	Buiucani	Industrialala-48	2	4	69000
36	Rishkanovka	Andrei Lupan-72	2	8	72300
37	Centru	Alexei Mateevici-81	1	9	31600
38	Botanica	Simeon Murafa-12	2	7	54200
39	Cekani	Mesterul Manole-20	4	5	99600
40	Centru	Bucuresti-17	5	4	105400
41	Buiucani	Industrialala-55	1	6	29900
42	Centru	Alexei Sciushev-43	5	2	142000
43	Buiucani	George Cosbuc	2	1	57600
44	Rishkanovka	Alec Russo-115	3	1	86200
45	Centru	Columna-54	4	2	97500
46	Cekani	Sargidava-19	1	9	36100
47	Botanica	Leuseni-4	2	13	61000
48	Cekani	Ilie Replin-2	3	14	78900
49	Rishkanovka	Petru Zadnipro-39	2	15	54100
50	Centru	Serghei Lazo-14	4	11	98600

Для продолжения нажмите любую клавишу . . .

Скриншот 5

Описание: Данная функция выводит шапку таблицы и информацию из массивов структур квартир. В зависимости от i (чётное или нечётное) меняется цвет заднего фона. С помощью функции `setw` данные ровно подставляются в столбцы.

Фрагмент кода

```
void showA(){ // Функция для вывода информации о квартирах
setcolor(0,15); // смена цвета заднего фона на белый
```

```
    cout<<endl<<"    Код    " <<"    Район    " <<"    Адрес    " <<"    Кол-во    " <<"
Этаж    " <<"    Цена    " <<endl;

    cout<<"    Квартиры                " <<"    комнат                " <<endl; // Создание
шапки таблицы
```

```

for(int i=0;i<nA;i++){
    if(i%2==0) setcolor(0,8); // Проверка для смены цвета через строку. Смена цвета текста на
тёмно-серый
    else setcolor(0,7); // Смена цвета на светло-серый
    cout <<left<<"    "<<setw(11)<<a[i].Kod <<setw(19)<< a[i].Raion <<setw(28) <<a[i].Adres
<<setw(8)<<a[i].Kkomnat <<setw(8)<<a[i].Etaj <<a[i].Price<<endl; // Вывод информации
    }
    setcolor(14,0); // Смена цвета на жёлтый
}


```

Информация об аренде

Текст задания:

Вывод на экран информации об аренде.

Скриншот вводимых данных:



Файл	Правка	Формат	Вид	Справка
1	Martinez Lorenzo	30 04 2022	2	
2	Patterson Camden	13 07 2020	3	
3	Scott Quenten	04 12 2015	1	
4	Brown Vidal	25 07 2017	4	
5	Gonzalez Duke	18 11 2020	5	
3	Howard Henrik	13 05 2022	2	
7	Promise Alexander	06 11 2021	3	
8	Xenos Howard	06 08 2015	1	
6	Williams Natalie	09 04 2019	2	
9	Irving James	01 09 2020	1	
3	Davis Dean	10 03 2017	2	
11	Johnson Isaias	05 01 2018	3	
10	Ferguson Harold	19 11 2021	2	
11	Martinez Nikko	11 12 2022	4	
13	Johnson Lorenzo	15 04 2022	5	
14	Gray Marcus	01 05 2022	4	
12	Jones Karen	02 08 2020	1	
15	Davis Gordon	08 10 2018	3	
16	Allen Torin	12 01 2021	2	
17	Brooks Thaddeus	28 09 2019	1	
18	King Briggs	04 12 2018	1	
19	Hernandez Elian	14 01 2021	2	
20	Nelson Ezekiel	14 09 2021	2	
21	Hernandez Diego	09 02 2022	4	
22	Moore Malachi	09 09 2015	5	
23	Scott Liam	05 06 2020	2	
24	Henderson Zander	07 01 2015	1	
25	Young Gentry	16 09 2020	2	
26	Perez Nick	26 01 2021	4	
27	Walker Josiah	15 02 2022	4	
28	Rogers Qadry	19 08 2022	2	
29	Barnes Uziel	15 09 2022	1	
24	Thomas Jase	03 09 2018	6	
31	Harris Taj	22 04 2018	2	
32	Powell Keaton	26 12 2019	1	
33	Hall Enzo	19 10 2015	2	
30	Ross Miriam	22 12 2010	5	
34	Hernandez Trenton	11 02 2015	3	
35	Hayes Lewis	24 09 2020	3	
36	Jenkins Ruben	17 02 2015	2	
37	Wright Kenneth	29 04 2022	1	
33	Young Omar	29 01 2018	2	
38	Hill Ricky	30 05 2020	3	
39	Taylor Abel	10 12 2018	3	
40	Torres Ian	12 02 2018	4	
41	Ward Bennett	31 12 2018	5	
42	Coleman Soren	13 01 2018	5	
33	Foster Juan	30 11 2021	2	
44	Johnson Lorenzo	01 05 2021	1	
45	Gray Giovanni	06 05 2015	6	
43	Vasquez Jesus	15 01 2022	1	
33	Rodriguez Felipe	12 07 2020	1	
47	Howard Finn	06 02 2015	3	
48	Harris Urias	11 05 2018	1	
49	Adams Keith	05 11 2020	1	
50	Lopez Tripp	21 07 2022	4	

Скриншот 6

Скриншот результата на экране:

Код квартиры	Фамилия	Имя	Дата аренды	Срок аренды
1	Martinez	Lorenzo	30/4 /2022	2
2	Patterson	Camden	13/7 /2020	3
3	Scott	Quenten	4 /12/2015	1
4	Brown	Vidal	25/7 /2017	4
5	Gonzalez	Duke	18/11/2020	5
3	Howard	Henrik	13/5 /2022	2
7	Promise	Alexander	6 /11/2021	3
8	Xenos	Howard	6 /8 /2015	1
6	Williams	Natalie	9 /4 /2019	2
9	Irving	James	1 /9 /2020	1
3	Davis	Dean	10/3 /2017	2
11	Johnson	Isaias	5 /1 /2018	3
10	Ferguson	Harold	19/11/2021	2
11	Martinez	Nikko	11/12/2022	4
13	Johnson	Lorenzo	15/4 /2022	5
14	Gray	Marcus	1 /5 /2022	4
12	Jones	Karen	2 /8 /2020	1
15	Davis	Gordon	8 /10/2018	3
16	Allen	Torin	12/1 /2021	2
17	Brooks	Thaddeus	28/9 /2019	1
18	King	Briggs	4 /12/2018	1
19	Hernandez	Eliau	14/1 /2021	2
20	Nelson	Ezekiel	14/9 /2021	2
21	Hernandez	Diego	9 /2 /2022	4
22	Moore	Malachi	9 /9 /2015	5
23	Scott	Liam	5 /6 /2020	2
24	Henderson	Zander	7 /1 /2015	1
25	Young	Gentry	16/9 /2020	2
26	Perez	Nick	26/1 /2021	4
27	Walker	Josiah	15/2 /2022	4
28	Rogers	Qadry	19/8 /2022	2
29	Barnes	Uziel	15/9 /2022	1
24	Thomas	Jase	3 /9 /2018	6
31	Harris	Taj	22/4 /2018	2
32	Powell	Keaton	26/12/2019	1
33	Hall	Enzo	19/10/2015	2
30	Ross	Miriam	22/12/2010	5
34	Hernandez	Trenton	11/2 /2015	3
35	Hayes	Lewis	24/9 /2020	3
36	Jenkins	Ruben	17/2 /2015	2
37	Wright	Kenneth	29/4 /2022	1
33	Young	Omar	29/1 /2018	2
38	Hill	Ricky	30/5 /2020	3
39	Taylor	Abel	10/12/2018	3
40	Torres	Ian	12/2 /2018	4
41	Ward	Bennett	31/12/2018	5
42	Coleman	Soren	13/1 /2018	5
33	Foster	Juan	30/11/2021	2
44	Johnson	Lorenzo	1 /5 /2021	1
45	Gray	Giovanni	6 /5 /2015	6
43	Vasquez	Jesus	15/1 /2022	1
33	Rodriguez	Felipe	12/7 /2020	1
47	Howard	Finn	6 /2 /2015	3
48	Harris	Urias	11/5 /2018	1
49	Adams	Keith	5 /11/2020	1
50	Lopez	Tripp	21/7 /2022	4

Для продолжения нажмите любую клавишу . . .

Скриншот 7

Описание: Данная функция выводит шапку таблицы и информацию из массивов структур аренд. В зависимости от i (чётное или нечётное) меняется цвет заднего фона. С помощью функции `setw` данные ровно подставляются в столбцы.

Фрагмент кода

```
void showR(){ // Функция для вывода информации об арендах
    setcolor(0,19); // Смена заднего фона на бирюзовый

    cout<<endl<<" Код квартиры " <<" Фамилия " <<" Имя " <<" Дата аренды"
    <<" Срок аренды "<<endl; // Создание шапки таблицы

    for(int i=0;i<nR;i++){
```

```

        if(i%2==0) setcolor(0,7); // Проверка для смены цвета через строку. Смена цвета на
        светло-серый
        else setcolor(7,1); // Смена цвета на тёмно-синий
        cout << left << "      " << setw(12) << r[i].Kod << setw(15) << r[i].Fam << setw(16) << r[i].Name
        << setw(2) << r[i].Day << "/" << setw(2) << r[i].Month << "/" << setw(14) << r[i].Year <<
        setw(5) << r[i].SrokA << endl; // Вывод информации
    }

    setcolor(14,0); // Смена цвета на жёлтый
}

```

Добавление информации о новой квартире

Текст задания:

Добавление информации о новой квартире.

Скриншот вводимых данных:

```
Введите код квартиры:51
Введите район:Centru
Введите адрес:Columna-56
Введите количество комнат:2
Введите этаж:4
Введите цену:47500
Введите позицию(при большой позиции информация добавится в конец файла):4

Информация о новой квартире добавлена.
Для продолжения нажмите любую клавишу . . .
```

Скриншот 8

Скриншот результата на экране:

До

Код Квартиры	Район	Адрес	Кол-во комнат	Этаж	Цена
1	Centru	Ismail-26	2	1	45000
2	Buiucani	Milano-12	3	3	74000
3	Rishkanovka	Andrei_Lupan-12	2	2	47000
4	Botanica	Sarmizegetusa-24	1	2	20000
5	Rishkanovka	Doina-3	2	9	31000
6	Botanica	Nicolae_Zelinski	4	11	89000
7	Cekani	Sargidava-4	1	7	25000
8	Centru	Bucuresti-3	2	4	38000
9	Rishkanovka	Doina-10	3	1	75000
10	Botanica	Sarmizegetusa-11	2	14	47000
11	Cekani	Industriala-16	1	9	25000
12	Buiucani	Suciava-4	3	7	59000
13	Botanica	Burebista	2	2	46000
14	Cekani	Otovasca-16	1	15	27600
15	Centru	Bucuresti-72	2	4	41820
16	Buiucani	Ion_Creanga-11	5	7	114200
17	Buiucani	Milano-55	3	2	76200
18	Cekani	Butuceni-18	4	1	96400
19	Rishkanovka	Criuleni-1	1	8	29600
20	Botanica	Padurii-44	2	7	52000
21	Rishkanovka	Doina-45	4	6	76900
22	Centru	Tigina-25	1	5	29650
23	Buiucani	Sucevita-22	1	4	35600
24	Rishkanovka	Timisoara-18	2	11	49900
25	Centru	Ismail-47	3	12	67500
26	Buiucani	Alba_Iulia-22	4	15	89600
27	Botanica	Gradina_Botanica-19	3	8	72000
28	Centru	Columna-13	5	4	156000
29	Cekani	Otovasca-19	2	8	41600
30	Rishkanovka	Ceucari-22	1	14	29900
31	Botanica	Prigoreni-12	2	3	41900
32	Centru	Bucuresti-11	4	11	91200
33	Centru	Tigina-56	3	15	35200
34	Cekani	Maria_Dragan-8	1	3	39600
35	Buiucani	Industriala-48	2	4	69000
36	Rishkanovka	Andrei_Lupan-72	2	8	72300
37	Centru	Alexei_Mateevici-81	1	9	31600
38	Botanica	Simeon_Murafa-12	2	7	54200
39	Cekani	Mesterul_Manole-20	4	5	99600
40	Centru	Bucuresti-17	5	4	105400
41	Buiucani	Industriala-55	1	6	29900
42	Centru	Alexei_Sciusev-43	5	2	142000
43	Buiucani	George_Cosbuc	2	1	57600
44	Rishkanovka	Alecu_Russo-115	3	1	86200
45	Centru	Columna-54	4	2	97500
46	Cekani	Sargidava-19	1	9	36100
47	Botanica	Leuseni-4	2	13	61000
48	Cekani	Ilie_Replin-2	3	14	78900
49	Rishkanovka	Petru_Zadnipru-39	2	15	54100
50	Centru	Serghei_Lazo-14	4	11	98600

Скриншот 9

После

Код Квартиры	Район	Адрес	Кол-во комнат	Этаж	Цена
1	Centru	Ismail-26	2	1	45000
2	Buiucani	Milano-12	3	3	74000
3	Rishkanovka	Andrei_Lupan-12	2	2	47000
51	Centru	Columna-56	2	4	47500
4	Botanica	Sarmizegetusa-24	1	2	20000
5	Rishkanovka	Doina-3	2	9	31000
6	Botanica	Nicolae_Zelinski	4	11	89000
7	Cekani	Sargidava-4	1	7	25000
8	Centru	Bucuresti-3	2	4	38000
9	Rishkanovka	Doina-10	3	1	75000
10	Botanica	Sarmizegetusa-11	2	14	47000
11	Cekani	Industriala-16	1	9	25000
12	Buiucani	Suciava-4	3	7	59000
13	Botanica	Burebista	2	2	46000
14	Cekani	Otovasca-16	1	15	27600
15	Centru	Bucuresti-72	2	4	41820
16	Buiucani	Ion_Creanga-11	5	7	114200
17	Buiucani	Milano-55	3	2	76200
18	Cekani	Butuceni-18	4	1	96400
19	Rishkanovka	Criuleni-1	1	8	29600
20	Botanica	Padurii-44	2	7	52000
21	Rishkanovka	Doina-45	4	6	76900
22	Centru	Tigina-25	1	5	29650
23	Buiucani	Sucevita-22	1	4	35600
24	Rishkanovka	Timisoara-18	2	11	49900
25	Centru	Ismail-47	3	12	67500
26	Buiucani	Alba_Iulia-22	4	15	89600
27	Botanica	Gradina_Botanica-19	3	8	72000
28	Centru	Columna-13	5	4	156000
29	Cekani	Otovasca-19	2	8	41600
30	Rishkanovka	Ceucari-22	1	14	29900
31	Botanica	Prigoreni-12	2	3	41900
32	Centru	Bucuresti-11	4	11	91200
33	Centru	Tigina-56	3	15	35200
34	Cekani	Maria_Dragan-8	1	3	39600
35	Buiucani	Industriala-48	2	4	69000
36	Rishkanovka	Andrei_Lupan-72	2	8	72300
37	Centru	Alexei_Mateevici-81	1	9	31600
38	Botanica	Simeon_Murafa-12	2	7	54200
39	Cekani	Mesterul_Manole-20	4	5	99600
40	Centru	Bucuresti-17	5	4	105400
41	Buiucani	Industriala-55	1	6	29900
42	Centru	Alexei_Sciusev-43	5	2	142000
43	Buiucani	George_Cosbuc	2	1	57600
44	Rishkanovka	Alecu_Russo-115	3	1	86200
45	Centru	Columna-54	4	2	97500
46	Cekani	Sargidava-19	1	9	36100
47	Botanica	Leuseni-4	2	13	61000
48	Cekani	Ilie_Replin-2	3	14	78900
49	Rishkanovka	Petru_Zadnipru-39	2	15	54100
50	Centru	Serghei_Lazo-14	4	11	98600

Скриншот 10

Описание: эта функция добавляет информацию о новой квартире. Для начала нужно создать структуру квартир, чтобы записывать в неё данные, переменную позиции и логическую переменную. Логическая переменная нужна для того, чтобы сделать защиту от неправильного ввода, так как номер квартиры не может повторяться. Далее пользователь вводит все данные. Если пользователь введёт слишком большую позицию, то информация добавится в конец. Далее массив смещается вправо до введённой позиции и заменяется введённой информацией о квартирах. Далее увеличивается счётчик квартир. Для добавления этой информации в файл используется функция `rewriteInfoA`.

Фрагмент кода:

```
void addA(){ // Функция для добавления информации о квартирах
    apartment ap; // создание структуры квартир
    int pos; // создание переменной позиции
    bool flag; // создание логической переменной
    cout<<"Введите код квартиры:";

    do{ // Защита от неправильного ввода
        cin>>ap.Kod; // ввод кода квартиры
        flag=1; // приравнивание логической переменной к 1
        for(int i=0;i<nA;i++)
            if(ap.Kod==a[i].Kod) flag=0; // если введённый код квартиры равен одному из кодов
            // квартиры файла, то переменная равна 0

        if(flag==0){ // если введённый код квартиры равен одному из кодов квартиры файла
            setcolor(12,0); // смена цвета текста на красный
            cout<<"Ошибка! Квартира с таким номером уже существует! Попробуйте ещё раз!
            "<<endl;
            setcolor(14,0); // смена цвета текста на жёлтый
            cout<<"Введите код квартиры:";

        }else if(ap.Kod<1) {
            setcolor(12,0); // смена цвета текста на красный
            cout<<"Ошибка! Попробуйте ещё раз! "<<endl;
            setcolor(14,0); // смена цвета текста на жёлтый
            cout<<"Введите код квартиры:";
```



```

}
}while(ap.Kod<1 or flag==0); // Защита от неправильного ввода

cout<<"Введите район:";
cin>>ap.Raion; // ввод района квартиры

cout<<"Введите адрес:";
cin>>ap.Adres; // ввод адреса квартиры

cout<<"Введите количество комнат:";
do{ // Защита от неправильного ввода
cin>>ap.Kkomnat;
if(ap.Kkomnat<1){
    setcolor(12,0); // смена цвета текста на красный
    cout<<"Ошибка! Попробуйте ещё раз! "<<endl;
    setcolor(14,0); // смена цвета текста на жёлтый
    cout<<"Введите количество комнат:";
}
}while(ap.Kkomnat<1); // Защита от неправильного ввода

cout<<"Введите этаж:";
do{ // Защита от неправильного ввода
cin>>ap.Etaj;
if(ap.Etaj<1){
    setcolor(12,0); // смена цвета текста на красный
    cout<<"Ошибка! Попробуйте ещё раз! "<<endl;
    setcolor(14,0); // смена цвета текста на жёлтый
    cout<<"Введите этаж:";
}
}while(ap.Etaj<1); // Защита от неправильного ввода

cout<<"Введите цену:";
do{ // Защита от неправильного ввода
cin>>ap.Price; // ввод цены квартиры
if(ap.Price<1){ // если цена квартиры меньше 1

```

```

        setcolor(12,0); // смена цвета текста на красный
        cout<<"Ошибка! Попробуйте ещё раз! "<<endl;
        setcolor(14,0); // смена цвета текста на жёлтый
        cout<<"Введите этаж:";
    }
} while(ap.Price<1); // Защита от неправильного ввода

        cout<<"Введите позицию(при большой позиции информация добавится в конец файла):";
        do{ // Защита от неправильного ввода
            cin>>pos; // ввод желаемой позиции
            if(pos<1 ){ // если желаемая позиция меньше 1
                setcolor(12,0); // смена цвета текста на красный
                cout<<"Ошибка! Попробуйте ещё раз! "<<endl;
                setcolor(14,0); // смена цвета текста на жёлтый
            }
            cout<<"Введите позицию(при большой позиции информация добавится в конец файла): ";
        }
    } while(pos<1); // Защита от неправильного ввода

    setcolor(10,0); // смена цвета текста на зелёный
    cout <<"\nИнформация о новой квартире добавлена.\n";
    setcolor(14,0); // смена цвета текста на жёлтый

    pos--; // Уменьшение переменной на 1, так как отсчёт массива начинается с нуля
    if(pos>nA) pos=nA; // если желаемая позиция больше максимальной в файле, то позиция
    становится последней
    for(int i=nA; i>pos; i--)
        a[i]=a[i-1]; // смещение массива структур вправо
    a[pos]=ap; // замена структур
    nA++; // увеличение количества квартир
    rewriteInfoA(); // перезапись файла квартир
}

```

Добавление информации о новой аренде

Текст задания:

Добавление информации о новой аренде.

Скриншот вводимых данных:

```
Введите код квартиры:51
Введите фамилию арендатора:Richardson
Введите имя арендатора:Hugo
Введите день аренды:1
Введите месяц аренды:6
Введите год аренды:2022
Введите срок аренды:2
Введите позицию(при большой позиции информация добавится в конец файла): 1

Информация об новой аренде добавлена.
Для продолжения нажмите любую клавишу . . .
```

Скриншот 11

Скриншот результата на экране:

До

Код квартиры	Фамилия	Имя	Дата аренды	Срок аренды
1	Martinez	Lorenzo	30/4 /2022	2
2	Patterson	Camden	13/7 /2020	3
3	Scott	Quenten	4 /12/2015	1
4	Brown	Vidal	25/7 /2017	4
5	Gonzalez	Duke	18/11/2020	5
3	Howard	Henrik	13/5 /2022	2
7	Promise	Alexander	6 /11/2021	3
8	Xenos	Howard	6 /8 /2015	1
6	Williams	Natalie	9 /4 /2019	2
9	Irving	James	1 /9 /2020	1
3	Davis	Dean	10/3 /2017	2
11	Johnson	Isaias	5 /1 /2018	3
10	Ferguson	Harold	19/11/2021	2
11	Martinez	Nikko	11/12/2022	4
13	Johnson	Lorenzo	15/4 /2022	5
14	Gray	Marcus	1 /5 /2022	4
12	Jones	Karen	2 /8 /2020	1
15	Davis	Gordon	8 /10/2018	3
16	Allen	Torin	12/1 /2021	2
17	Brooks	Thaddeus	28/9 /2019	1
18	King	Briggs	4 /12/2018	1
19	Hernandez	Elian	14/1 /2021	2
20	Nelson	Ezekiel	14/9 /2021	2
21	Hernandez	Diego	9 /2 /2022	4
22	Moore	Malachi	9 /9 /2015	5
23	Scott	Liam	5 /6 /2020	2
24	Henderson	Zander	7 /1 /2015	1
25	Young	Gentry	16/9 /2020	2
26	Perez	Nick	26/1 /2021	4
27	Walker	Josiah	15/2 /2022	4
28	Rogers	Qadry	19/8 /2022	2
29	Barnes	Uziel	15/9 /2022	1
24	Thomas	Jase	3 /9 /2018	6
31	Harris	Taj	22/4 /2018	2
32	Powell	Keaton	26/12/2019	1
33	Hall	Enzo	19/10/2015	2
30	Ross	Miriam	22/12/2010	5
34	Hernandez	Trenton	11/2 /2015	3
35	Hayes	Lewis	24/9 /2020	3
36	Jenkins	Ruben	17/2 /2015	2
37	Wright	Kenneth	29/4 /2022	1
33	Young	Omar	29/1 /2018	2
38	Hill	Ricky	30/5 /2020	3
39	Taylor	Abel	10/12/2018	3
40	Torres	Ian	12/2 /2018	4
41	Ward	Bennett	31/12/2018	5
42	Coleman	Soren	13/1 /2018	5
33	Foster	Juan	30/11/2021	2
44	Johnson	Lorenzo	1 /5 /2021	1
45	Gray	Giovanni	6 /5 /2015	6
43	Vasquez	Jesus	15/1 /2022	1
33	Rodriguez	Felipe	12/7 /2020	1
47	Howard	Finn	6 /2 /2015	3
48	Harris	Urias	11/5 /2018	1
49	Adams	Keith	5 /11/2020	1
50	Lopez	Tripp	21/7 /2022	4

Для продолжения нажмите любую клавишу . . .

Скриншот 12

После

Код квартиры	Фамилия	Имя	Дата аренды	Срок аренды
51	Richardson	Hugo	1 /6 /2022	2
1	Martinez	Lorenzo	30/4 /2022	2
2	Patterson	Camden	13/7 /2020	3
3	Scott	Quenten	4 /12/2015	1
4	Brown	Vidal	25/7 /2017	4
5	Gonzalez	Duke	18/11/2020	5
3	Howard	Henrik	13/5 /2022	2
7	Promise	Alexander	6 /11/2021	3
8	Xenos	Howard	6 /8 /2015	1
6	Williams	Natalie	9 /4 /2019	2
9	Irving	James	1 /9 /2020	1
3	Davis	Dean	10/3 /2017	2
11	Johnson	Isaias	5 /1 /2018	3
10	Ferguson	Harold	19/11/2021	2
11	Martinez	Nikko	11/12/2022	4
13	Johnson	Lorenzo	15/4 /2022	5
14	Gray	Marcus	1 /5 /2022	4
12	Jones	Karen	2 /8 /2020	1
15	Davis	Gordon	8 /10/2018	3
16	Allen	Torin	12/1 /2021	2
17	Brooks	Thaddeus	28/9 /2019	1
18	King	Briggs	4 /12/2018	1
19	Hernandez	Elian	14/1 /2021	2
20	Nelson	Ezekiel	14/9 /2021	2
21	Hernandez	Diego	9 /2 /2022	4
22	Moore	Malachi	9 /9 /2015	5
23	Scott	Liam	5 /6 /2020	2
24	Henderson	Zander	7 /1 /2015	1
25	Young	Gentry	16/9 /2020	2
26	Perez	Nick	26/1 /2021	4
27	Walker	Josiah	15/2 /2022	4
28	Rogers	Qadry	19/8 /2022	2
29	Barnes	Uziel	15/9 /2022	1
24	Thomas	Jase	3 /9 /2018	6
31	Harris	Taj	22/4 /2018	2
32	Powell	Keaton	26/12/2019	1
33	Hall	Enzo	19/10/2015	2
30	Ross	Miriam	22/12/2010	5
34	Hernandez	Trenton	11/2 /2015	3
35	Hayes	Lewis	24/9 /2020	3
36	Jenkins	Ruben	17/2 /2015	2
37	Wright	Kenneth	29/4 /2022	1
33	Young	Omar	29/1 /2018	2
38	Hill	Ricky	30/5 /2020	3
39	Taylor	Abel	10/12/2018	3
40	Torres	Ian	12/2 /2018	4
41	Ward	Bennett	31/12/2018	5
42	Coleman	Soren	13/1 /2018	5
33	Foster	Juan	30/11/2021	2
44	Johnson	Lorenzo	1 /5 /2021	1
45	Gray	Giovanni	6 /5 /2015	6
43	Vasquez	Jesus	15/1 /2022	1
33	Rodriguez	Felipe	12/7 /2020	1
47	Howard	Finn	6 /2 /2015	3
48	Harris	Urias	11/5 /2018	1
49	Adams	Keith	5 /11/2020	1
50	Lopez	Tripp	21/7 /2022	4

Для продолжения нажмите любую клавишу . . .

Скриншот 13

Описание: эта функция добавляет информацию об новой аренде. Для начала нужно создать структуру квартир, чтобы записывать в неё данные и переменную позиции. Далее пользователь вводит все данные. Если пользователь введёт слишком большую позицию, то информация добавится в конец. Далее массив смещается вправо до введённой позиции и заменяется введённая информация об арендах. Далее увеличивается счётчик аренд. Для добавления этой информации в файл используется функция `rewriteInfoR`.

Фрагмент кода:

```
void addR(){ // Функция для добавления информации об арендах
    rent rt; // создание структуры аренд
    int pos; // создание переменной позиции

    cout<<"Введите код квартиры:";
    do{ // Защита от неправильного ввода
        cin>>rt.Kod; // ввод кода квартиры
        if(rt.Kod<1){ // если код квартиры меньше 1
            setcolor(12,0); // смена цвета текста на красный
            cout<<"Ошибка! Попробуйте ещё раз! "<<endl;
            setcolor(14,0); // смена цвета текста на жёлтый
            cout<<"Введите код квартиры:";
        }
    }while(rt.Kod<1); // Защита от неправильного ввода

    cout<<"Введите фамилию арендатора:";
    cin>>rt.Fam; // ввод фамилии арендатора

    cout<<"Введите имя арендатора:";
    cin>>rt.Name; // ввод имени арендатора

    cout<<"Введите день аренды:";

    do{ // Защита от неправильного ввода
        cin>>rt.Day; // ввод дня аренды
        if(rt.Day<1 or rt.Day>31){ // если день аренды меньше 1 или больше 31
```

```

        setcolor(12,0); // смена цвета текста на красный
        cout<<"Ошибка! Попробуйте ещё раз! "<<endl;
        setcolor(14,0); // смена цвета текста на жёлтый
        cout<<"Введите день аренды:";
    }
} while(rt.Day<1 or rt.Day>31); // Защита от неправильного ввода

        cout<<"Введите месяц аренды:";
        do{ // Защита от неправильного ввода
            cin>>rt.Month; // ввод месяца аренды
            if(rt.Month<1 or rt.Month>12){ // если месяц аренды меньше 1 или больше 12
                setcolor(12,0); // смена цвета текста на красный
                cout<<"Ошибка! Попробуйте ещё раз! "<<endl;
                setcolor(14,0); // смена цвета текста на жёлтый
            }
            cout<<"Введите месяц аренды:";
        }
} while(rt.Month<1 or rt.Month>12); // Защита от неправильного ввода

        cout<<"Введите год аренды:";

        do{ // Защита от неправильного ввода
            cin>>rt.Year; // ввод года аренды
            if(rt.Year<1){ // если год аренды меньше 1
                setcolor(12,0); // смена цвета текста на красный
                cout<<"Ошибка! Попробуйте ещё раз! "<<endl;
                setcolor(14,0); // смена цвета текста на жёлтый
            }
            cout<<"Введите год аренды:";
        }
} while(rt.Year<1 ); // Защита от неправильного ввода

        cout<<"Введите срок аренды:";

        do{ // Защита от неправильного ввода
            cin>>rt.SrokA; // ввод срока аренды
            if(rt.SrokA<1){ // если срок аренды меньше 1
                setcolor(12,0); // смена цвета текста на красный

```

```

cout<<"Ошибка! Попробуйте ещё раз! "<<endl;
setcolor(14,0); // смена цвета текста на жёлтый
cout<<"Введите срок аренды:";
}
}while(rt.SrokA<1 ); // Защита от неправильного ввода

cout<<"Введите позицию(при большой позиции информация добавится в конец файла):
";
do{ // Защита от неправильного ввода
cin>>pos; // ввод желаемой позиции
if(pos<1){ // если желаемая позиция меньше 1
setcolor(12,0); // смена цвета текста на красный
cout<<"Ошибка! Попробуйте ещё раз! "<<endl;
setcolor(14,0); // смена цвета текста на жёлтый
cout<<"Введите позицию(при большой позиции информация добавится в конец файла): ";
}
}while(pos<1); // Защита от неправильного ввода

setcolor(10,0); // смена цвета текста на зелёный
cout <<"\nИнформация об новой аренде добавлена.\n";
setcolor(14,0); // смена цвета текста на жёлтый
pos--; // Уменьшение переменной на 1, так как отсчёт массива начинается с нуля
if(pos>nR) pos=nR; // если желаемая позиция больше максимальной в файле, то позиция
становится последней
for(int i=nR; i>pos; i--)
r[i]=r[i-1]; // смещение массива структур вправо
r[pos]=rt; // замена структур
nR++; // увеличение количества аренд
rewriteInfoR(); // перезапись файла аренд
}

```

Удаление информации о квартире, код которой введен с клавиатуры

Текст задания:

Удаление информации о квартире, код которой введен с клавиатуры.

Скриншот вводимых данных:

Введите код квартиры для удаления:51
Информация о квартире под номером 51 удалена.
Для продолжения нажмите любую клавишу . . .

Скриншот 14

Скриншот результата на экране:

До

Код Квартиры	Район	Адрес	Кол-во комнат	Этаж	Цена
1	Centru	Ismail-26	2	1	45000
2	Buiucani	Milano-12	3	3	74000
3	Rishkanovka	Andrei_Lupan-12	2	2	47000
51	Centru	Columna-56	2	4	47500
4	Botanica	Sarmizegetusa-24	1	2	20000
5	Rishkanovka	Doina-3	2	9	31000
6	Botanica	Nicolae_Zelinski	4	11	89000
7	Cekani	Sargidava-4	1	7	25000
8	Centru	Bucuresti-3	2	4	38000
9	Rishkanovka	Doina-10	3	1	75000
10	Botanica	Sarmizegetusa-11	2	14	47000
11	Cekani	Industriala-16	1	9	25000
12	Buiucani	Suciava-4	3	7	59000
13	Botanica	Burebista	2	2	46000
14	Cekani	Otovasca-16	1	15	27600
15	Centru	Bucuresti-72	2	4	41820
16	Buiucani	Ion_Creanga-11	5	7	114200
17	Buiucani	Milano-55	3	2	76200
18	Cekani	Butuceni-18	4	1	96400
19	Rishkanovka	Criuleni-1	1	8	29600
20	Botanica	Padurii-44	2	7	52000
21	Rishkanovka	Doina-45	4	6	76900
22	Centru	Tigina-25	1	5	29650
23	Buiucani	Sucevita-22	1	4	35600
24	Rishkanovka	Timisoara-18	2	11	49900
25	Centru	Ismail-47	3	12	67500
26	Buiucani	Alba_Iulia-22	4	15	89600
27	Botanica	Gradina_Botanica-19	3	8	72000
28	Centru	Columna-13	5	4	156000
29	Cekani	Otovasca-19	2	8	41600
30	Rishkanovka	Ceucari-22	1	14	29900
31	Botanica	Prigoreni-12	2	3	41900
32	Centru	Bucuresti-11	4	11	91200
33	Centru	Tigina-56	3	15	35200
34	Cekani	Maria_Dragan-8	1	3	39600
35	Buiucani	Industriala-48	2	4	69000
36	Rishkanovka	Andrei_Lupan-72	2	8	72300
37	Centru	Alexei_Mateevici-81	1	9	31600
38	Botanica	Simeon_Murafa-12	2	7	54200
39	Cekani	Mesterul_Manole-20	4	5	99600
40	Centru	Bucuresti-17	5	4	105400
41	Buiucani	Industriala-55	1	6	29900
42	Centru	Alexei_Sciusev-43	5	2	142000
43	Buiucani	George_Cosbuc	2	1	57600
44	Rishkanovka	Alecu_Russo-115	3	1	86200
45	Centru	Columna-54	4	2	97500
46	Cekani	Sargidava-19	1	9	36100
47	Botanica	Leuseni-4	2	13	61000
48	Cekani	Ilie_Replin-2	3	14	78900
49	Rishkanovka	Petru_Zadnipro-39	2	15	54100
50	Centru	Serghei_Lazo-14	4	11	98600

Для продолжения нажмите любую клавишу . . .

Скриншот 15

Код квартиры	Фамилия	Имя	Дата аренды	Срок аренды
51	Richardson	Hugo	1 /6 /2022	2
1	Martinez	Lorenzo	30/4 /2022	2
2	Patterson	Camden	13/7 /2020	3
3	Scott	Quenten	4 /12/2015	1
4	Brown	Vidal	25/7 /2017	4
5	Gonzalez	Duke	18/11/2020	5
3	Howard	Henrik	13/5 /2022	2
7	Promise	Alexander	6 /11/2021	3
8	Xenos	Howard	6 /8 /2015	1
6	Williams	Natalie	9 /4 /2019	2
9	Irving	James	1 /9 /2020	1
3	Davis	Dean	10/3 /2017	2
11	Johnson	Isaias	5 /1 /2018	3
10	Ferguson	Harold	19/11/2021	2
11	Martinez	Nikko	11/12/2022	4
13	Johnson	Lorenzo	15/4 /2022	5
14	Gray	Marcus	1 /5 /2022	4
12	Jones	Karen	2 /8 /2020	1
15	Davis	Gordon	8 /10/2018	3
16	Allen	Torin	12/1 /2021	2
17	Brooks	Thaddeus	28/9 /2019	1
18	King	Briggs	4 /12/2018	1
19	Hernandez	Elian	14/1 /2021	2
20	Nelson	Ezekiel	14/9 /2021	2
21	Hernandez	Diego	9 /2 /2022	4
22	Moore	Malachi	9 /9 /2015	5
23	Scott	Liam	5 /6 /2020	2
24	Henderson	Zander	7 /1 /2015	1
25	Young	Gentry	16/9 /2020	2
26	Perez	Nick	26/1 /2021	4
27	Walker	Josiah	15/2 /2022	4
28	Rogers	Qadry	19/8 /2022	2
29	Barnes	Uziel	15/9 /2022	1
24	Thomas	Jase	3 /9 /2018	6
31	Harris	Taj	22/4 /2018	2
32	Powell	Keaton	26/12/2019	1
33	Hall	Enzo	19/10/2015	2
30	Ross	Miriam	22/12/2010	5
34	Hernandez	Trenton	11/2 /2015	3
35	Hayes	Lewis	24/9 /2020	3
36	Jenkins	Ruben	17/2 /2015	2
37	Wright	Kenneth	29/4 /2022	1
33	Young	Omar	29/1 /2018	2
38	Hill	Ricky	30/5 /2020	3
39	Taylor	Abel	10/12/2018	3
40	Torres	Ian	12/2 /2018	4
41	Ward	Bennett	31/12/2018	5
42	Coleman	Soren	13/1 /2018	5
33	Foster	Juan	30/11/2021	2
44	Johnson	Lorenzo	1 /5 /2021	1
45	Gray	Giovanni	6 /5 /2015	6
43	Vasquez	Jesus	15/1 /2022	1
33	Rodriguez	Felipe	12/7 /2020	1
47	Howard	Finn	6 /2 /2015	3
48	Harris	Ulias	11/5 /2018	1
49	Adams	Keith	5 /11/2020	1
50	Lopez	Tripp	21/7 /2022	4

Для продолжения нажмите любую клавишу . . .

Скриншот 16

После

Код Квартиры	Район	Адрес	Кол-во комнат	Этаж	Цена
1	Centru	Ismail-26	2	1	45000
2	Buiucani	Milano-12	3	3	74000
3	Rishkanovka	Andrei Lupan-12	2	2	47000
4	Botanica	Sarmizegetusa-24	1	2	20000
5	Rishkanovka	Doina-3	2	9	31000
6	Botanica	Nicolae Zelinski	4	11	89000
7	Cekani	Sargidava-4	1	7	25000
8	Centru	Bucuresti-3	2	4	38000
9	Rishkanovka	Doina-10	3	1	75000
10	Botanica	Sarmizegetusa-11	2	14	47000
11	Cekani	Industrial-16	1	9	25000
12	Buiucani	Suciava-4	3	7	59000
13	Botanica	Burebista	2	2	46000
14	Cekani	Otovasca-16	1	15	27600
15	Centru	Bucuresti-72	2	4	41820
16	Buiucani	Ion Creanga-11	5	7	114200
17	Buiucani	Milano-55	3	2	76200
18	Cekani	Butuceni-18	4	1	96400
19	Rishkanovka	Criuleni-1	1	8	29600
20	Botanica	Padurii-44	2	7	52000
21	Rishkanovka	Doina-45	4	6	76900
22	Centru	Tigina-25	1	5	29650
23	Buiucani	Sucevita-22	1	4	35600
24	Rishkanovka	Timisoara-18	2	11	49900
25	Centru	Ismail-47	3	12	67500
26	Buiucani	Alba Iulia-22	4	15	89600
27	Botanica	Gradina Botanica-19	3	8	72000
28	Centru	Columna-13	5	4	156000
29	Cekani	Otovasca-19	2	8	41600
30	Rishkanovka	Ceucari-22	1	14	29900
31	Botanica	Prigoreni-12	2	3	41900
32	Centru	Bucuresti-11	4	11	91200
33	Centru	Tigina-56	3	15	35200
34	Cekani	Maria Dragan-8	1	3	39600
35	Buiucani	Industrial-48	2	4	69000
36	Rishkanovka	Andrei Lupan-72	2	8	72300
37	Centru	Alexei Mateevici-81	1	9	31600
38	Botanica	Simeon Murafa-12	2	7	54200
39	Cekani	Mesterul Manole-20	4	5	99600
40	Centru	Bucuresti-17	5	4	105400
41	Buiucani	Industrial-55	1	6	29900
42	Centru	Alexei Sciuzev-43	5	2	142000
43	Buiucani	George Cosbuc	2	1	57600
44	Rishkanovka	Alecu Russo-115	3	1	86200
45	Centru	Columna-54	4	2	97500
46	Cekani	Sargidava-19	1	9	36100
47	Botanica	Leuseni-4	2	13	61000
48	Cekani	Ilie Replin-2	3	14	78900
49	Rishkanovka	Petru Zadnipro-39	2	15	54100
50	Centru	Sergei Lazo-14	4	11	98600

Для продолжения нажмите любую клавишу . . .

Скриншот 17

Описание: данная функция выполняет удаление информации о квартире по коду квартиры. Для начала создаются переменные для кода квартиры и счётчика. Далее следует поиск квартиры с введённым кодом. Если такой код найден, то массив структур начиная от найденной позиции *i* сдвигается влево. Уменьшается на 1 счётчик квартир и увеличивается счётчик для проверки. Если этот счётчик не увеличился, то выводится информация о том, что код квартиры не найден, в другом случае выводится информация, что квартира под этим кодом удалена. Далее этот же код квартиры ищется в массиве структур и аналогично удаляется, и уменьшается счётчик аренд. Для удаления информации из файлов используются функции *rewriteInfoA* и *rewriteInfoR*.

Код квартиры	Фамилия	Имя	Дата аренды	Срок аренды
1	Martinez	Lorenzo	30/4 /2022	2
2	Patterson	Camden	13/7 /2020	3
3	Scott	Quenten	4 /12/2015	1
4	Brown	Vidal	25/7 /2017	4
5	Gonzalez	Duke	18/11/2020	5
3	Howard	Henrik	13/5 /2022	2
7	Promise	Alexander	6 /11/2021	3
8	Xenos	Howard	6 /8 /2015	1
6	Williams	Natalie	9 /4 /2019	2
9	Irving	James	1 /9 /2020	1
3	Davis	Dean	10/3 /2017	2
11	Johnson	Isaias	5 /1 /2018	3
10	Ferguson	Harold	19/11/2021	2
11	Martinez	Nikko	11/12/2022	4
13	Johnson	Lorenzo	15/4 /2022	5
14	Gray	Marcus	1 /5 /2022	4
12	Jones	Karen	2 /8 /2020	1
15	Davis	Gordon	8 /10/2018	3
16	Allen	Torin	12/1 /2021	2
17	Brooks	Thaddeus	28/9 /2019	1
18	King	Briggs	4 /12/2018	1
19	Hernandez	Elian	14/1 /2021	2
20	Nelson	Ezekiel	14/9 /2021	2
21	Hernandez	Diego	9 /2 /2022	4
22	Moore	Malachi	9 /9 /2015	5
23	Scott	Liam	5 /6 /2020	2
24	Henderson	Zander	7 /1 /2015	1
25	Young	Gentry	16/9 /2020	2
26	Perez	Nick	26/1 /2021	4
27	Walker	Josiah	15/2 /2022	4
28	Rogers	Qadry	19/8 /2022	2
29	Barnes	Uziel	15/9 /2022	1
24	Thomas	Jase	3 /9 /2018	6
31	Harris	Taj	22/4 /2018	2
32	Powell	Keaton	26/12/2019	1
33	Hall	Enzo	19/10/2015	2
30	Ross	Miriam	22/12/2010	5
34	Hernandez	Trenton	11/2 /2015	3
35	Hayes	Lewis	24/9 /2020	3
36	Jenkins	Ruben	17/2 /2015	2
37	Wright	Kenneth	29/4 /2022	1
33	Young	Omar	29/1 /2018	2
38	Hill	Ricky	30/5 /2020	3
39	Taylor	Abel	10/12/2018	3
40	Torres	Ian	12/2 /2018	4
41	Ward	Bennett	31/12/2018	5
42	Coleman	Soren	13/1 /2018	5
33	Foster	Juan	30/11/2021	2
44	Johnson	Lorenzo	1 /5 /2021	1
45	Gray	Giovanni	6 /5 /2015	6
43	Vasquez	Jesus	15/1 /2022	1
33	Rodriguez	Felipe	12/7 /2020	1
47	Howard	Finn	6 /2 /2015	3
48	Harris	Urias	11/5 /2018	1
49	Adams	Keith	5 /11/2020	1
50	Lopez	Tripp	21/7 /2022	4

Для продолжения нажмите любую клавишу . . .

Скриншот 18

Фрагмент кода:

```
void deleteInfoA() { // функция для удаления информации о квартире
    int k,p; // создание переменной для кода квартиры и счётчика
    cout<<"\nВведите код квартиры для удаления:";

    do { // защита от неправильного ввода
        cin>> k; // ввод кода квартиры
        if(k<1) { // если код квартиры меньше 1
            setcolor(12,0); // смена цвета текста на красный
            cout<<"Ошибка! Попробуйте ещё раз! "<<endl;
            setcolor(14,0); // смена цвета текста на жёлтый
            cout<<"\nВведите код квартиры для удаления:";
        }
    } while(k<1); // защита от неправильного ввода

    for(int i=0; i<nA; i++)
        if(k== a[i].Kod) { // если введённый код квартиры есть в массиве структур квартир
            for(int j=i; j<nA; j++) a[j]=a[j+1]; // сдвиг массива квартир влево
            nA--; // уменьшение количества квартир
            p++; // увеличение счётчика для проверки
        }
    if(p==0) { // если переменная не увеличилась
        setcolor(12,0); // смена цвета текста на красный
        cout << "\nКод квартиры не найден.\n";
        setcolor(14,0); // смена цвета текста на жёлтый
    }

    else { // в другом случае
        setcolor(10,0); // смена цвета текста на зелёный
        cout << "Информация о квартире под номером " << k << " удалена.\n\n";
        setcolor(14,0); // смена цвета текста на жёлтый
        for(int i=0; i<nR; i++)
            if(k==r[i].Kod) { // если введённый код квартиры есть в массиве структур аренд
                for(int j=i; j<nA; j++)
                    r[j]=r[j+1]; // сдвиг массива аренд влево
```

```
        nR--; // уменьшение количества аренд
    }

    rewriteInfoA(); // перезапись файла квартир
    rewriteInfoR(); // перезапись файла аренд
}

}
```

Удаление информации об аренде, код которой введен с клавиатуры

Текст задания:

Удаление информации об аренде порядковый номер которой вводится с клавиатуры.

Скриншот вводимых данных:

Введите порядковый номер аренды для удаления:1
Информация об аренде под порядковым номером 1 удалена.
Для продолжения нажмите любую клавишу . . .

Скриншот 19

Скриншот результата на экране:

До

После

Код квартиры	Фамилия	Имя	Дата аренды	Срок аренды
1	Martinez	Lorenzo	30/4 /2022	2
2	Patterson	Camden	13/7 /2020	3
3	Scott	Quenten	4 /12/2015	1
4	Brown	Vidal	25/7 /2017	4
5	Gonzalez	Duke	18/11/2020	5
3	Howard	Henrik	13/5 /2022	2
7	Promise	Alexander	6 /11/2021	3
8	Xenos	Howard	6 /8 /2015	1
6	Williams	Natalie	9 /4 /2019	2
9	Irving	James	1 /9 /2020	1
3	Davis	Dean	10/3 /2017	2
11	Johnson	Isaias	5 /1 /2018	3
10	Ferguson	Harold	19/11/2021	2
11	Martinez	Nikko	11/12/2022	4
13	Johnson	Lorenzo	15/4 /2022	5
14	Gray	Marcus	1 /5 /2022	4
12	Jones	Karen	2 /8 /2020	1
15	Davis	Gordon	8 /10/2018	3
16	Allen	Torin	12/1 /2021	2
17	Brooks	Thaddeus	28/9 /2019	1
18	King	Briggs	4 /12/2018	1
19	Hernandez	Eliau	14/1 /2021	2
20	Nelson	Ezekiel	14/9 /2021	2
21	Hernandez	Diego	9 /2 /2022	4
22	Moore	Malachi	9 /9 /2015	5
23	Scott	Liam	5 /6 /2020	2
24	Henderson	Zander	7 /1 /2015	1
25	Young	Gentry	16/9 /2020	2
26	Perez	Nick	26/1 /2021	4
27	Walker	Josiah	15/2 /2022	4
28	Rogers	Qadry	19/8 /2022	2
29	Barnes	Uziel	15/9 /2022	1
24	Thomas	Jase	3 /9 /2018	6
31	Harris	Taj	22/4 /2018	2
32	Powell	Keaton	26/12/2019	1
33	Hall	Enzo	19/10/2015	2
30	Ross	Miriam	22/12/2010	5
34	Hernandez	Trenton	11/2 /2015	3
35	Hayes	Lewis	24/9 /2020	3
36	Jenkins	Ruben	17/2 /2015	2
37	Wright	Kenneth	29/4 /2022	1
33	Young	Omar	29/1 /2018	2
38	Hill	Ricky	30/5 /2020	3
39	Taylor	Abel	10/12/2018	3
40	Torres	Ian	12/2 /2018	4
41	Ward	Bennett	31/12/2018	5
42	Coleman	Soren	13/1 /2018	5
33	Foster	Juan	30/11/2021	2
44	Johnson	Lorenzo	1 /5 /2021	1
45	Gray	Giovanni	6 /5 /2015	6
43	Vasquez	Jesus	15/1 /2022	1
33	Rodriguez	Felipe	12/7 /2020	1
47	Howard	Finn	6 /2 /2015	3
48	Harris	Urias	11/5 /2018	1
49	Adams	Keith	5 /11/2020	1
50	Lopez	Tripp	21/7 /2022	4

Для продолжения нажмите любую клавишу . . .

Скриншот 20

Код квартиры	Фамилия	Имя	Дата аренды	Срок аренды
2	Patterson	Camden	13/7 /2020	3
3	Scott	Quenten	4 /12/2015	1
4	Brown	Vidal	25/7 /2017	4
5	Gonzalez	Duke	18/11/2020	5
3	Howard	Henrik	13/5 /2022	2
7	Promise	Alexander	6 /11/2021	3
8	Xenos	Howard	6 /8 /2015	1
6	Williams	Natalie	9 /4 /2019	2
9	Irving	James	1 /9 /2020	1
3	Davis	Dean	10/3 /2017	2
11	Johnson	Isaias	5 /1 /2018	3
10	Ferguson	Harold	19/11/2021	2
11	Martinez	Nikko	11/12/2022	4
13	Johnson	Lorenzo	15/4 /2022	5
14	Gray	Marcus	1 /5 /2022	4
12	Jones	Karen	2 /8 /2020	1
15	Davis	Gordon	8 /10/2018	3
16	Allen	Torin	12/1 /2021	2
17	Brooks	Thaddeus	28/9 /2019	1
18	King	Briggs	4 /12/2018	1
19	Hernandez	Eliau	14/1 /2021	2
20	Nelson	Ezekiel	14/9 /2021	2
21	Hernandez	Diego	9 /2 /2022	4
22	Moore	Malachi	9 /9 /2015	5
23	Scott	Liam	5 /6 /2020	2
24	Henderson	Zander	7 /1 /2015	1
25	Young	Gentry	16/9 /2020	2
26	Perez	Nick	26/1 /2021	4
27	Walker	Josiah	15/2 /2022	4
28	Rogers	Qadry	19/8 /2022	2
29	Barnes	Uziel	15/9 /2022	1
24	Thomas	Jase	3 /9 /2018	6
31	Harris	Taj	22/4 /2018	2
32	Powell	Keaton	26/12/2019	1
33	Hall	Enzo	19/10/2015	2
30	Ross	Miriam	22/12/2010	5
34	Hernandez	Trenton	11/2 /2015	3
35	Hayes	Lewis	24/9 /2020	3
36	Jenkins	Ruben	17/2 /2015	2
37	Wright	Kenneth	29/4 /2022	1
33	Young	Omar	29/1 /2018	2
38	Hill	Ricky	30/5 /2020	3
39	Taylor	Abel	10/12/2018	3
40	Torres	Ian	12/2 /2018	4
41	Ward	Bennett	31/12/2018	5
42	Coleman	Soren	13/1 /2018	5
33	Foster	Juan	30/11/2021	2
44	Johnson	Lorenzo	1 /5 /2021	1
45	Gray	Giovanni	6 /5 /2015	6
45	Gray	Giovanni	6 /5 /2015	6
43	Vasquez	Jesus	15/1 /2022	1
33	Rodriguez	Felipe	12/7 /2020	1
47	Howard	Finn	6 /2 /2015	3
48	Harris	Urias	11/5 /2018	1
49	Adams	Keith	5 /11/2020	1

Для продолжения нажмите любую клавишу . . .

Скриншот 21

Описание: данная функция выполняет удаление информации об аренде по порядковому номеру. Для начала создается переменная для порядкового номера аренды. Если введенный

порядковый номер больше количества аренд, то выводится информация о том, что порядковый номер аренды не найден. В другом случае массив структур начиная от введённого порядкового номера сдвигается влево и уменьшается на 1 количество аренд. Для удаления информации из файла используется функция `rewriteInfoR`.

Фрагмент кода:

```
void deleteInfoR() { // функция для удаления информации об аренде
    int pn; // создание переменной для порядкового номера аренды
    cout<<"\nВведите порядковый номер аренды для удаления:";

    do { // защита от неправильного ввода
        cin>> pn;
        if(pn<1) { // если порядковый номер аренды меньше 1
            setcolor(12,0); // смена цвета текста на красный
            cout<<"Ошибка! Попробуйте ещё раз! "<<endl;
            setcolor(14,0); // смена цвета текста на жёлтый
            cout<<"\nВведите порядковый номер аренды для удаления:";
        }
    } while(pn<1); // защита от неправильного ввода
    pn--; // Уменьшение переменной на 1, так как отсчёт массива начинается с нуля

    if(pn>nR) { // если порядковый номер аренды больше величины массива структур аренд
        setcolor(12,0); // смена цвета на красный
        cout << "\nПорядковый номер аренды не найден.\n";
        setcolor(14,0); // смена цвета на жёлтый
    } else { // в другом случае
        for(int i=pn; i<nR-1; i++)
            r[i]=r[i+1]; // сдвиг массива аренд влево
        nR--; // уменьшение количества аренд

        setcolor(10,0); // смена цвета текста на зелёный
        cout << "Информация об аренде под порядковым номером " << pn+1 << " удалена.\n";
        setcolor(14,0); // смена цвета текста на жёлтый
        rewriteInfoR(); // перезапись файла аренд
    }
}
```

Редактирование информации о квартире, код которой введен с клавиатуры

Текст задания:

Редактирование информации о квартире, код которой введен с клавиатуры.

Скриншот вводимых данных:

Введите код квартиры для редактирования: 50

Скриншот 22

Что вы хотите изменить?

<< Район >> << Адрес >> << Количество комнат >> << Этаж >> << Цена >> << Всё >> << Отмена >>

Введите район:Botanica
Введите адрес:Botanica_Veche-14
Введите количество комнат:3
Введите этаж:9
Введите цену:75000

Данные изменены
Для продолжения нажмите любую клавишу . . .

Скриншот 23

Скриншот результата на экране:

До

После

Код Квартиры	Район	Адрес	Кол-во комнат	Этаж	Цена
1	Centru	Ismail-26	2	1	45000
2	Buiucani	Milano-12	3	3	74000
3	Rishkanovka	Andrei_Lupan-12	2	2	47000
4	Botanica	Sarmizegetusa-24	1	2	20000
5	Rishkanovka	Doina-3	2	9	31000
6	Botanica	Nicolae_Zelinski	4	11	89000
7	Cekani	Sargidava-4	1	7	25000
8	Centru	Bucuresti-3	2	4	38000
9	Rishkanovka	Doina-10	3	1	75000
10	Botanica	Sarmizegetusa-11	2	14	47000
11	Cekani	Industrialia-16	1	9	25000
12	Buiucani	Suciava-4	3	7	59000
13	Botanica	Burebista	2	2	46000
14	Cekani	Otovasca-16	1	15	27600
15	Centru	Bucuresti-72	2	4	41820
16	Buiucani	Ion_Creanga-11	5	7	114200
17	Buiucani	Milano-55	3	2	76200
18	Cekani	Butuceni-18	4	1	96400
19	Rishkanovka	Criuleni-1	1	8	29600
20	Botanica	Padurii-44	2	7	52000
21	Rishkanovka	Doina-45	4	6	76900
22	Centru	Tigina-25	1	5	29650
23	Buiucani	Sucevita-22	1	4	35600
24	Rishkanovka	Timisoara-18	2	11	49900
25	Centru	Ismail-47	3	12	67500
26	Buiucani	Alba_Iulia-22	4	15	89600
27	Botanica	Gradina_Botanica-19	3	8	72000
28	Centru	Columna-13	5	4	156000
29	Cekani	Otovasca-19	2	8	41600
30	Rishkanovka	Ceucari-22	1	14	29900
31	Botanica	Prigoreni-12	2	3	41900
32	Centru	Bucuresti-11	4	11	91200
33	Centru	Tigina-56	3	15	35200
34	Cekani	Maria_Dragan-8	1	3	39600
35	Buiucani	Industrialia-48	2	4	69000
36	Rishkanovka	Andrei_Lupan-72	2	8	72300
37	Centru	Alexei_Mateevici-81	1	9	31600
38	Botanica	Simeon_Murafa-12	2	7	54200
39	Cekani	Mesterul_Manole-20	4	5	99600
40	Centru	Bucuresti-17	5	4	105400
41	Buiucani	Industrialia-55	1	6	29900
42	Centru	Alexei_Sciusev-43	5	2	142000
43	Buiucani	George_Cosbuc	2	1	57600
44	Rishkanovka	Alecu_Russo-115	3	1	86200
45	Centru	Columna-54	4	2	97500
46	Cekani	Sargidava-19	1	9	36100
47	Botanica	Leuseni-4	2	13	61000
48	Cekani	Ilie_Replin-2	3	14	78900
49	Rishkanovka	Petru_Zadnipru-39	2	15	54100
50	Centru	Serghei_Lazo-14	4	11	98600

Для продолжения нажмите любую клавишу . . .

Скриншот 24

Код Квартиры	Район	Адрес	Кол-во комнат	Этаж	Цена
1	Centru	Ismail-26	2	1	45000
2	Buiucani	Milano-12	3	3	74000
3	Rishkanovka	Andrei_Lupan-12	2	2	47000
4	Botanica	Sarmizegetusa-24	1	2	20000
5	Rishkanovka	Doina-3	2	9	31000
6	Botanica	Nicolae_Zelinski	4	11	89000
7	Cekani	Sargidava-4	1	7	25000
8	Centru	Bucuresti-3	2	4	38000
9	Rishkanovka	Doina-10	3	1	75000
10	Botanica	Sarmizegetusa-11	2	14	47000
11	Cekani	Industrialia-16	1	9	25000
12	Buiucani	Suciava-4	3	7	59000
13	Botanica	Burebista	2	2	46000
14	Cekani	Otovasca-16	1	15	27600
15	Centru	Bucuresti-72	2	4	41820
16	Buiucani	Ion_Creanga-11	5	7	114200
17	Buiucani	Milano-55	3	2	76200
18	Cekani	Butuceni-18	4	1	96400
19	Rishkanovka	Criuleni-1	1	8	29600
20	Botanica	Padurii-44	2	7	52000
21	Rishkanovka	Doina-45	4	6	76900
22	Centru	Tigina-25	1	5	29650
23	Buiucani	Sucevita-22	1	4	35600
24	Rishkanovka	Timisoara-18	2	11	49900
25	Centru	Ismail-47	3	12	67500
26	Buiucani	Alba_Iulia-22	4	15	89600
27	Botanica	Gradina_Botanica-19	3	8	72000
28	Centru	Columna-13	5	4	156000
29	Cekani	Otovasca-19	2	8	41600
30	Rishkanovka	Ceucari-22	1	14	29900
31	Botanica	Prigoreni-12	2	3	41900
32	Centru	Bucuresti-11	4	11	91200
33	Centru	Tigina-56	3	15	35200
34	Cekani	Maria_Dragan-8	1	3	39600
35	Buiucani	Industrialia-48	2	4	69000
36	Rishkanovka	Andrei_Lupan-72	2	8	72300
37	Centru	Alexei_Mateevici-81	1	9	31600
38	Botanica	Simeon_Murafa-12	2	7	54200
39	Cekani	Mesterul_Manole-20	4	5	99600
40	Centru	Bucuresti-17	5	4	105400
41	Buiucani	Industrialia-55	1	6	29900
42	Centru	Alexei_Sciusev-43	5	2	142000
43	Buiucani	George_Cosbuc	2	1	57600
44	Rishkanovka	Alecu_Russo-115	3	1	86200
45	Centru	Columna-54	4	2	97500
46	Cekani	Sargidava-19	1	9	36100
47	Botanica	Leuseni-4	2	13	61000
48	Cekani	Ilie_Replin-2	3	14	78900
49	Rishkanovka	Petru_Zadnipru-39	2	15	54100
50	Botanica	Botanica_Veche-14	3	9	75000

Для продолжения нажмите любую клавишу . . .

Скриншот 25

Описание: данная функция редактирует информацию о квартире по коду квартиры. Для начала создаются переменные для кода квартиры и позиции. Далее пользователь вводит код квартиры и после этого, для правильной работы мыши в меню очищается консоль. Далее в цикле ищется введённый код квартиры и если такой найден, то переменная *p* равняется *i*. Если данная переменная изменилась, то выводятся все кнопки и следует подключение мыши для управления в консоли. Далее в зависимости от решения пользователя, редактируются данные. Если переменная *v* не равна 0 (отмена), то выводится информация о том, что данные изменены, в другом случае выводится информация о том, что введённый код квартиры не найден. Для редактирования этой информации в файле с информацией о квартирах используется функция *rewriteInfoA*.

Фрагмент кода:

```
void RedactA(){ // функция для редактирования информации о квартире
    int k,p; // создание переменных для кода квартиры и позиции
    cout <<"Введите код квартиры для редактирования: ";

    do{ // защита от неправильного ввода
        cin>> k; // ввод кода квартиры

        if(k<1){ // если код квартиры меньше 1
            setcolor(12,0); // смена цвета текста на красный
            cout<<"Ошибка! Попробуйте ещё раз! "<<endl;
            setcolor(14,0); // смена цвета текста на жёлтый
            cout <<"Введите код квартиры для редактирования: ";
        }
    }while(k<1); // защита от неправильного ввода

    system("CLS"); // очистка консоли
    for(int i=0;i<nA;i++)
        if(k == a[i].Kod) p=i; // если введённый код квартиры равен коду квартиры из файла, то
        переменная p равна порядковому номеру
        if(p!=0){ // если переменная p изменилась
            setcolor(15,0); // смена цвета текста на белый
            cout <<"\nЧто вы хотите изменить?\n\n";
            setcolor(14,0); // смена цвета текста на жёлтый
```

```

cout << "    << Район >>" << " << Адрес >>" << " << Количество комнат >>"
<< " << Этаж >>" << " << Цена >>" << " << Всё >>";
setcolor(12,0); // смена цвета текста на красный
cout << " << Отмена >>" << endl;
setcolor(14,0); // смена цвета текста на жёлтый

int v=swredA(); // считывание нажатия мыши по кнопкам в консоли

switch(v){
    case 1:{
cout<<"Введите район:";
    cin>> a[p].Raion; // ввод района

    rewriteInfoA(); // перезапись файла квартир
        break;
    }
    case 2:{
        cout<<"Введите адрес:";
        cin>> a[p].Adres; // ввод адреса

        rewriteInfoA(); // перезапись файла квартир
            break;
        }
    case 3:{

        cout<<"Введите количество комнат:";
            do{ // защита от неправильного ввода
        cin>> a[p].Kkomnat; // ввод количества комнат
        if(a[p].Kkomnat<1){ // если количество комнат меньше 1
            setcolor(12,0); // смена цвета текста на красный
            cout<<"Ошибка! Попробуйте ещё раз!" << endl;
            setcolor(14,0); // смена цвета текста на жёлтый
            cout<<"Введите количество комнат:";
        }
    }
}

```

```
} while(a[p].Kkomnat<1); // защита от неправильного ввода
```

```
rewriteInfoA(); // перезапись файла квартир
```

```
break;
```

```
}
```

```
case 4:{
```

```
    cout<<"Введите этаж:";
```

```
do{ // защита от неправильного ввода
```

```
cin>> a[p].Etaj; // ввод этажа
```

```
if(a[p].Etaj<1){
```

```
    setcolor(12,0); // смена цвета текста на красный
```

```
    cout<<"Ошибка! Попробуйте ещё раз! "<<endl;
```

```
    setcolor(14,0); // смена цвета текста на жёлтый
```

```
    cout<<"Введите этаж:";
```

```
}
```

```
} while(a[p].Etaj<1); // защита от неправильного ввода
```

```
rewriteInfoA(); // перезапись файла квартир
```

```
break;
```

```
}
```

```
case 5:{
```

```
    cout<<"Введите цену:";
```

```
do{ // защита от неправильного ввода
```

```
cin>> a[p].Price; // ввод цены
```

```
if(a[p].Price<1){ // если цена меньше 1
```

```
    setcolor(12,0); // смена цвета текста на красный
```

```
    cout<<"Ошибка! Попробуйте ещё раз! "<<endl;
```

```
    setcolor(14,0); // смена цвета текста на жёлтый
```

```
    cout<<"Введите цену:";
```

```
}
```

```
} while(a[p].Price<1); // защита от неправильного ввода
```



```

rewriteInfoA(); // перезапись файла квартир
    break;
}
case 6:{

cout<<"Введите район:";
    cin>> a[p].Raion; // ввод района

    cout<<"Введите адрес:";
    cin>> a[p].Adres; // ввод адреса


    cout<<"Введите количество комнат:";
        do{ // защита от неправильного ввода
cin>> a[p].Kkomnat; // ввод количества комнат
if(a[p].Kkomnat<1){ // если количество комнат меньше 1
    setcolor(12,0); // смена цвета текста на красный
    cout<<"Ошибка! Попробуйте ещё раз! "<<endl;
    setcolor(14,0); // смена цвета текста на жёлтый
    cout<<"Введите количество комнат:";
}
}while(a[p].Kkomnat<1); // защита от неправильного ввода


    cout<<"Введите этаж:";
    do{ // защита от неправильного ввода
cin>> a[p].Etaj; // ввод этажа
if(a[p].Etaj<1){
    setcolor(12,0); // смена цвета текста на красный
    cout<<"Ошибка! Попробуйте ещё раз! "<<endl;
    setcolor(14,0); // смена цвета текста на жёлтый
    cout<<"Введите этаж:";
}
}while(a[p].Etaj<1); // защита от неправильного ввода

```

```

    cout<<"Введите цену:";
    do{ // защита от неправильного ввода
    cin>> a[p].Price; // ввод цены
    if(a[p].Price<1){
        setcolor(12,0); // смена цвета текста на красный
        cout<<"Ошибка! Попробуйте ещё раз! "<<endl;
        setcolor(14,0); // смена цвета текста на жёлтый
        cout<<"Введите цену:";
    }
    }while(a[p].Price<1); // защита от неправильного ввода

rewriteInfoA(); // перезапись файла квартир
break;
}

case 0:{
    setcolor(12,0); // смена цвета текста на красный
    cout << "\nОтмена.\n";
    setcolor(14,0); // смена цвета текста на жёлтый
    break;
}

}

if(v!=0){ // если счётчик изменился
    setcolor(10,0); // смена цвета текста на зелёный
    cout << "\nДанные изменены\n";
    setcolor(14,0); // смена цвета текста на жёлтый
}

}

}else{ // в другом случае
    setcolor(12,0); // смена цвета текста на красный
    cout<<"Код квартиры не найден.\n";
    setcolor(14,0); // смена цвета текста на жёлтый
}

}

}

```

Редактирование информации об аренде, код которой введен с клавиатуры

Текст задания:

Редактирование информации об аренде порядковый номер которой вводится с клавиатуры.

Скриншот вводимых данных:

Введите порядковый номер аренды для редактирования: 10

Скриншот 26

Что вы хотите изменить?

<< Фамилия >> << Имя >> << День >> << Месяц >> << Год >> << Срок аренды >> << Всё >> << Отмена >>

Введите фамилию арендатора:Rodgers

Введите имя арендатора:John

Введите день аренды:9

Введите месяц аренды:11

Введите год аренды:2021

Введите срок аренды:1

Данные изменены

Для продолжения нажмите любую клавишу . . .

Скриншот 27

Скриншот результата на экране:

До

Код квартиры	Фамилия	Имя	Дата аренды	Срок аренды
1	Martinez	Lorenzo	30/4 /2022	2
2	Patterson	Camden	13/7 /2020	3
3	Scott	Quenten	4 /12/2015	1
4	Brown	Vidal	25/7 /2017	4
5	Gonzalez	Duke	18/11/2020	5
3	Howard	Henrik	13/5 /2022	2
7	Promise	Alexander	6 /11/2021	3
8	Xenos	Howard	6 /8 /2015	1
6	Williams	Natalie	9 /4 /2019	2
9	Irving	James	1 /9 /2020	1
3	Davis	Dean	10/3 /2017	2
11	Johnson	Isaias	5 /1 /2018	3
10	Ferguson	Harold	19/11/2021	2
11	Martinez	Nikko	11/12/2022	4
13	Johnson	Lorenzo	15/4 /2022	5
14	Gray	Marcus	1 /5 /2022	4
12	Jones	Karen	2 /8 /2020	1
15	Davis	Gordon	8 /10/2018	3
16	Allen	Torin	12/1 /2021	2
17	Brooks	Thaddeus	28/9 /2019	1
18	King	Briggs	4 /12/2018	1
19	Hernandez	Elian	14/1 /2021	2
20	Nelson	Ezekiel	14/9 /2021	2
21	Hernandez	Diego	9 /2 /2022	4
22	Moore	Malachi	9 /9 /2015	5
23	Scott	Liam	5 /6 /2020	2
24	Henderson	Zander	7 /1 /2015	1
25	Young	Gentry	16/9 /2020	2
26	Perez	Nick	26/1 /2021	4
27	Walker	Josiah	15/2 /2022	4
28	Rogers	Qadry	19/8 /2022	2
29	Barnes	Uziel	15/9 /2022	1
24	Thomas	Jase	3 /9 /2018	6
31	Harris	Taj	22/4 /2018	2
32	Powell	Keaton	26/12/2019	1
33	Hall	Enzo	19/10/2015	2
30	Ross	Miriam	22/12/2010	5
34	Hernandez	Trenton	11/2 /2015	3
35	Hayes	Lewis	24/9 /2020	3
36	Jenkins	Ruben	17/2 /2015	2
37	Wright	Kenneth	29/4 /2022	1
33	Young	Omar	29/1 /2018	2
38	Hill	Ricky	30/5 /2020	3
39	Taylor	Abel	10/12/2018	3
40	Torres	Ian	12/2 /2018	4
41	Ward	Bennett	31/12/2018	5
42	Coleman	Soren	13/1 /2018	5
33	Foster	Juan	30/11/2021	2
44	Johnson	Lorenzo	1 /5 /2021	1
45	Gray	Giovanni	6 /5 /2015	6
43	Vasquez	Jesus	15/1 /2022	1
33	Rodriguez	Felipe	12/7 /2020	1
47	Howard	Finn	6 /2 /2015	3
48	Harris	Urias	11/5 /2018	1
49	Adams	Keith	5 /11/2020	1
50	Lopez	Tripp	21/7 /2022	4

Для продолжения нажмите любую клавишу . . .

Скриншот 28

После

Код квартиры	Фамилия	Имя	Дата аренды	Срок аренды
1	Martinez	Lorenzo	30/4 /2022	2
2	Patterson	Camden	13/7 /2020	3
3	Scott	Quenten	4 /12/2015	1
4	Brown	Vidal	25/7 /2017	4
5	Gonzalez	Duke	18/11/2020	5
3	Howard	Henrik	13/5 /2022	2
7	Promise	Alexander	6 /11/2021	3
8	Xenos	Howard	6 /8 /2015	1
6	Williams	Natalie	9 /4 /2019	2
9	Rodgers	John	9 /11/2021	1
3	Davis	Dean	10/3 /2017	2
11	Johnson	Isaias	5 /1 /2018	3
10	Ferguson	Harold	19/11/2021	2
11	Martinez	Nikko	11/12/2022	4
13	Johnson	Lorenzo	15/4 /2022	5
14	Gray	Marcus	1 /5 /2022	4
12	Jones	Karen	2 /8 /2020	1
15	Davis	Gordon	8 /10/2018	3
16	Allen	Torin	12/1 /2021	2
17	Brooks	Thaddeus	28/9 /2019	1
18	King	Briggs	4 /12/2018	1
19	Hernandez	Elian	14/1 /2021	2
20	Nelson	Ezekiel	14/9 /2021	2
21	Hernandez	Diego	9 /2 /2022	4
22	Moore	Malachi	9 /9 /2015	5
23	Scott	Liam	5 /6 /2020	2
24	Henderson	Zander	7 /1 /2015	1
25	Young	Gentry	16/9 /2020	2
26	Perez	Nick	26/1 /2021	4
27	Walker	Josiah	15/2 /2022	4
28	Rogers	Qadry	19/8 /2022	2
29	Barnes	Uziel	15/9 /2022	1
24	Thomas	Jase	3 /9 /2018	6
31	Harris	Taj	22/4 /2018	2
32	Powell	Keaton	26/12/2019	1
33	Hall	Enzo	19/10/2015	2
30	Ross	Miriam	22/12/2010	5
34	Hernandez	Trenton	11/2 /2015	3
35	Hayes	Lewis	24/9 /2020	3
36	Jenkins	Ruben	17/2 /2015	2
37	Wright	Kenneth	29/4 /2022	1
33	Young	Omar	29/1 /2018	2
38	Hill	Ricky	30/5 /2020	3
39	Taylor	Abel	10/12/2018	3
40	Torres	Ian	12/2 /2018	4
41	Ward	Bennett	31/12/2018	5
42	Coleman	Soren	13/1 /2018	5
33	Foster	Juan	30/11/2021	2
44	Johnson	Lorenzo	1 /5 /2021	1
45	Gray	Giovanni	6 /5 /2015	6
43	Vasquez	Jesus	15/1 /2022	1
33	Rodriguez	Felipe	12/7 /2020	1
47	Howard	Finn	6 /2 /2015	3
48	Harris	Urias	11/5 /2018	1
49	Adams	Keith	5 /11/2020	1
50	Lopez	Tripp	21/7 /2022	4

Для продолжения нажмите любую клавишу . . .

Скриншот 29

Описание: данная функция редактирует информацию об аренде по порядковому номеру. Для начала создается переменная для порядкового номера аренды. После этого эта переменная уменьшается на 1, так как отсчёт в массиве начинается с нуля. Далее для правильной работы мыши в меню очищается консоль. Далее выводятся все кнопки и следует подключение мыши для управления в консоли. Далее в зависимости от решения пользователя, редактируются данные. Если переменная *v* не равна 0(отмена), то выводится информация о том, что данные изменены. Для редактирования этой информации в файле с информацией об арендах используется функция *rewriteInfoR*.

Фрагмент кода:

```
void RedactR(){ // функция для редактирования информации об аренде
    int k; // создание переменной для порядкового номера аренды
    cout <<"Введите порядковый номер аренды для редактирования: ";
    do{ // защита от неправильного ввода
        cin>> k; // ввод порядкового номера аренды
        if(k<1 or k>nR){
            setcolor(12,0); // смена цвета текста на красный
            cout<<"Ошибка! Попробуйте ещё раз! "<<endl;
            setcolor(14,0); // смена цвета текста на жёлтый
            cout <<"Введите порядковый номер аренды для редактирования: ";
        }
    }while(k<1 or k>nR); // защита от неправильного ввода

    k--; // уменьшение переменной на 1, так как отсчёт массива начинается с нуля
    system("CLS"); // очистка консоли
    setcolor(15,0); // смена цвета текста на белый
    cout <<"\nЧто вы хотите изменить?\n\n";
    setcolor(14,0); // смена цвета текста на жёлтый

    cout <<"    << Фамилия >>" <<" << Имя >>" <<" << День >>" <<" << Месяц >>"
    <<" << Год >>" <<" << Срок аренды >>"<<" << Всё >>";
    setcolor(12,0); // смена цвета текста на красный
    cout<<" << Отмена >>"<<endl;
    setcolor(14,0); // смена цвета текста на жёлтый
```

```
int v=swredR(); // считывание нажатия мыши по кнопкам в консоли
```

```
switch(v){  
    case 1:{  
        cout<<"Введите фамилию арендатора:";  
        cin>>r[k].Fam; // ввод фамилии арендатора  
        rewriteInfoR(); // перезапись файла аренд  
        break;  
    }  
    case 2:{  
        cout<<"Введите имя арендатора:";  
        cin>>r[k].Name; // ввод имени арендатора  
        rewriteInfoR(); // перезапись файла аренд  
        break;  
    }  
    case 3:{  
        cout<<"Введите день аренды:";  
        do{ // защита от неправильного ввода  
            cin>>r[k].Day; // ввод дня аренды  
            if(r[k].Day<1 or r[k].Day>31){ // если день аренды меньше 1 или больше 31  
                setcolor(12,0); // смена цвета текста на красный  
                cout<<"Ошибка! Попробуйте ещё раз! "<<endl;  
                setcolor(14,0); // смена цвета текста на жёлтый  
                cout<<"Введите день аренды:";  
            }  
        } while(r[k].Day<1 or r[k].Day>31); // защита от неправильного ввода  
        rewriteInfoR(); // перезапись файла аренд  
        break;  
    }  
    case 4:{  
        cout<<"Введите месяц аренды:";  
        do{ // защита от неправильного ввода  
            cin>>r[k].Month; // ввод месяца аренды  
            if(r[k].Month<1 or r[k].Month>12){ // если месяц аренды меньше 1 или больше 12  
                setcolor(12,0); // смена цвета текста на красный
```

```

    cout<<"Ошибка! Попробуйте ещё раз! "<<endl;
    setcolor(14,0); // смена цвета текста на жёлтый
    cout<<"Введите месяц аренды:";
}
}while(r[k].Month<1 or r[k].Month>12); // защита от неправильного ввода

rewriteInfoR(); // перезапись файла аренд
    break;
}
case 5:{
    cout<<"Введите год аренды:";
    do{ // защита от неправильного ввода
        cin>>r[k].Year; // ввод года аренды
        if(r[k].Year<1){ // если год аренды меньше 1
            setcolor(12,0); // смена цвета текста на красный
            cout<<"Ошибка! Попробуйте ещё раз! "<<endl;
            setcolor(14,0); // смена цвета текста на жёлтый
            cout<<"Введите год аренды:";
        }
    }while(r[k].Year<1); // защита от неправильного ввода

    rewriteInfoR(); // перезапись файла аренд
    break;
}
case 6:{
    cout<<"Введите срок аренды:";
    do{ // защита от неправильного ввода
        cin>>r[k].SrokA; // ввод срока аренды
        if(r[k].SrokA<1){ // если срок аренды меньше 1
            setcolor(12,0); // смена цвета текста на красный
            cout<<"Ошибка! Попробуйте ещё раз! "<<endl;
            setcolor(14,0); // смена цвета текста на жёлтый
            cout<<"Введите срок аренды:";
        }
    }while(r[k].SrokA<1); // защита от неправильного ввода
}

```

```

rewriteInfoR(); // перезапись файла аренд
    break;
}
case 7:{
    cout<<"Введите фамилию арендатора:";
    cin>>r[k].Fam; // ввод фамилии арендатора

    cout<<"Введите имя арендатора:";
    cin>>r[k].Name; // ввод имени арендатора

    cout<<"Введите день аренды:";
    do{ // защита от неправильного ввода
    cin>>r[k].Day; // ввод дня аренды
    if(r[k].Day<1 or r[k].Day>31){ // если день аренды меньше 1 или больше 31
        setcolor(12,0); // смена цвета текста на красный
        cout<<"Ошибка! Попробуйте ещё раз! "<<endl;
        setcolor(14,0); // смена цвета текста на жёлтый
        cout<<"Введите день аренды:";
    }
} while(r[k].Day<1 or r[k].Day>31); // защита от неправильного ввода

    cout<<"Введите месяц аренды:";
    do{ // защита от неправильного ввода
    cin>>r[k].Month; // ввод месяца аренды
    if(r[k].Month<1 or r[k].Month>12){ // если месяц аренды меньше 1 или больше 12
        setcolor(12,0); // смена цвета текста на красный
        cout<<"Ошибка! Попробуйте ещё раз! "<<endl;
        setcolor(14,0); // смена цвета текста на жёлтый
        cout<<"Введите месяц аренды:";
    }
} while(r[k].Month<1 or r[k].Month>12); // защита от неправильного ввода

    cout<<"Введите год аренды:";
    do{ // защита от неправильного ввода

```

```

cin>>r[k].Year; // ввод года аренды
if(r[k].Year<1){ // если год аренды меньше 1
    setcolor(12,0); // смена цвета текста на красный
    cout<<"Ошибка! Попробуйте ещё раз! "<<endl;
    setcolor(14,0); // смена цвета текста на жёлтый
    cout<<"Введите год аренды:";
}
}while(r[k].Year<1); // защита от неправильного ввода

    cout<<"Введите срок аренды:";
do{ // защита от неправильного ввода
    cin>>r[k].SrokA; // ввод срока аренды
    if(r[k].SrokA<1){ // если срок аренды меньше 1
        setcolor(12,0); // смена цвета текста на красный
        cout<<"Ошибка! Попробуйте ещё раз! "<<endl;
        setcolor(14,0); // смена цвета текста на жёлтый
        cout<<"Введите срок аренды:";
    }
}while(r[k].SrokA<1); // защита от неправильного ввода

    rewriteInfoR(); // перезапись файла аренд
    break;
}
case 0:{
setcolor(12,0); // смена цвета текста на красный
    cout << "\nОтмена.\n";
    setcolor(14,0); // смена цвета текста на жёлтый
    }
    }
    if(v!=0){ // если переменная v изменилась
        setcolor(10,0); // смена цвета текста на зелёный
        cout << "\nДанные изменены\n";
        setcolor(14,0); // смена цвета текста на жёлтый
    }
}
}

```


Создание текстового файла, содержащего информацию об арендах, срок которых истек

Текст задания:

Создание текстового файла Expire.txt, содержащего информацию об арендах, срок которых истек на дату, введенную с клавиатуры.

Скриншот вводимых данных:

```
Введите день(1-31): 12
Введите месяц(1-12): 6
Введите год(>1): 2022
```

Скриншот 30

```
Желаете открыть файл? << Да >> << Нет >>
Открытие файла...
Для продолжения нажмите любую клавишу . . .
```

Скриншот 31

Скриншот результата на экране:

```
Желаете открыть файл? << Да >> << Нет >>
Открытие файла...
Для продолжения нажмите любую клавишу . . .
```

Файл	Правка	Формат	Вид	Справка
3	Scott	Quenten	4	12 2015 1
4	Brown	Vidal	25	7 2017 4
8	Xenos	Howard	6	8 2015 1
6	Williams	Natalie	9	4 2019 2
9	Irving	James	1	9 2020 1
3	Davis	Dean	10	3 2017 2
11	Johnson	Isaias	5	1 2018 3
12	Jones	Karen	2	8 2020 1
15	Davis	Gordon	8	10 2018 3
17	Brooks	Thaddeus	28	9 2019 1
18	King	Briggs	4	12 2018 1
22	Moore	Malachi	9	9 2015 5
23	Scott	Liam	5	6 2020 2
24	Henderson	Zander	7	1 2015 1
31	Harris	Taj	22	4 2018 2
32	Powell	Keaton	26	12 2019 1
33	Hall	Enzo	19	10 2015 2
30	Ross	Miriam	22	12 2010 5
34	Hernandez	Trenton	11	2 2015 3
36	Jenkins	Ruben	17	2 2015 2
33	Young	Omar	29	1 2018 2
39	Taylor	Abel	10	12 2018 3
40	Torres	Ian	12	2 2018 4
44	Johnson	Lorenzo	1	5 2021 1
45	Gray	Giovanni	6	5 2015 6
33	Rodriguez	Felipe	12	7 2020 1
47	Howard	Finn	6	2 2015 3
48	Harris	Urias	11	5 2018 1
49	Adams	Keith	5	11 2020 1

Стр 100% Windows (CRLF) UTF-8

Скриншот 32

Описание: данная функция создаёт текстовый файл *Expire.txt*, содержащий информацию об арендах, срок которых истек на дату, введенную с клавиатуры. Для начала создаётся массив из 3 элементов для дня, месяца и года. Далее пользователь вводит желаемый день, месяц и год. Сначала программа сравнивает введенный год и год+ срок аренды из файла, если введенный год больше года+ срока аренды из файла, то данные записываются в новый файл. В другом случае если введенный год и год+ срок аренды из файла равны, то сравниваются месяца, если введенный месяц больше месяца из файла, то данные записываются в файл. В другом случае если введенный год и год+ срок аренды из файла равны и введенный месяц равен месяцу из файла, то сравниваются введенный день и день из файла, если введенный день больше дня из файла, то данные записываются в файл. Далее очищается консоль для правильной работы мыши, выводится сообщение о том, что файл создан и выводятся кнопки. Далее пользователю предлагается выбор открытия файла. При нажатии на кнопку << Да >> откроется текстовый файл с информацией об истекивших арендах и в консоль выведется сообщение об открытии файла, в другом случае в консоль выведется сообщение о закрытии файла.

Фрагмент кода:

```
void Srok() { // функция для поиска истёкших аренд
int date[3]; // создание массива для желаемой даты
cout << "Введите день(1-31): ";
do { // защита от неправильного ввода
cin >> date[0]; // ввод дня

if (date[0] < 1 or date[0] > 31) { // если день меньше 1 или больше 31
    setcolor(12, 0); // смена цвета текста на красный
    cout << "Ошибка! Попробуйте ещё раз! " << endl;
    setcolor(14, 0); // смена цвета текста на жёлтый
    cout << "Введите день(1-31): ";
}
} while (date[0] < 1 or date[0] > 31); // защита от неправильного ввода

cout << "Введите месяц(1-12): ";
do { // защита от неправильного ввода
cin >> date[1]; // ввод месяца
```

```

if(date[1]<1 or date[1]>12){ // если месяц меньше 1 или больше 12
    setcolor(12,0); // смена цвета текста на красный
    cout<<"Ошибка! Попробуйте ещё раз! "<<endl;
    setcolor(14,0); // смена цвета текста на жёлтый
    cout <<"Введите месяц(1-31): ";
}
}while(date[1]<1 or date[1]>12); // защита от неправильного ввода

cout <<"Введите год(>1): ";
do{ // защита от неправильного ввода

cin >> date[2]; // ввод года

if(date[2]<1){ // если год меньше 1
    setcolor(12,0); // смена цвета текста на красный
    cout<<"Ошибка! Попробуйте ещё раз! "<<endl;
    setcolor(14,0); // смена цвета текста на жёлтый
    cout <<"Введите год(>1): ";
}
}while(date[2]<1); // защита от неправильного ввода

ofstream f("Expire.txt"); // открытие файла для записи

for(int i=0;i<nR;i++){
    if(date[2] > r[i].Year + r[i].SrokA) // если введённый год больше года+срока аренды
f <<r[i].Kod <<" "<< r[i].Fam <<" "<<r[i].Name <<" "<<r[i].Day<<" "<<r[i].Month<<"
"<<r[i].Year<<" "<<r[i].SrokA<<endl; // запись информации об истекшей аренде в файл

else if(date[2] == r[i].Year + r[i].SrokA and date[1] > r[i].Month) // если введённый год равен
году+сроку аренды и введённый месяц больше месяца аренды

f <<r[i].Kod <<" "<< r[i].Fam <<" "<<r[i].Name <<" "<<r[i].Day<<" "<<r[i].Month<<"
"<<r[i].Year<<" "<<r[i].SrokA<<endl; // запись информации об истекшей аренде в файл

```

```

else if(date[2] == r[i].Year + r[i].SrokA and date[1] == r[i].Month and date[0] > r[i].Day )// если
введённый год равен году+сроку аренды и введённый месяц равен месяцу аренды и
введённый день больше дня аренды
f << r[i].Kod << " " << r[i].Fam << " " << r[i].Name << " " << r[i].Day << " " << r[i].Month << "
" << r[i].Year << " " << r[i].SrokA << endl; // запись информации об истекшей аренде в файл
}
system("CLS"); // очистка консоли
setcolor(10,0); // смена цвета текста на зелёный
cout << "Файл создан." << endl;
setcolor(14,0); // смена цвета текста на жёлтый
cout << internal << setw(175) << "Желаете открыть файл?";
setcolor(10,0); // смена цвета текста на зелёный
cout << "    << Да >>";
setcolor(12,0); // смена цвета текста на красный
cout << "    << Нет >>" << endl;
setcolor(14,0); // смена цвета текста на жёлтый

int ch=swopenFile(); // считывание нажатия мыши по кнопкам в консоли
if(ch==1){ // если была нажата кнопка << Да >>
    ShellExecute(NULL, "open", "Expire.txt", NULL, NULL, SW_SHOWNORMAL); // открытие
созданного файла
    setcolor(10,0); // смена цвета текста на зелёный
    cout << "Открытие файла... " << endl;
    setcolor(14,0); // смена цвета текста на жёлтый
}else{ // в другом случае
    setcolor(12,0); // смена цвета текста на красный
    cout << "Закрытие файла..." << endl;
    setcolor(14,0); // смена цвета текста на жёлтый
}
f.close(); // закрытие файла
}

```

Список свободных квартир в порядке возрастания количества комнат

Текст задания:

Вывод на экран списка свободных квартир в порядке возрастания количества комнат.

Скриншот результата на экране:

Код Квартиры	Район	Адрес	Кол-во комнат	Этаж	Цена
4	Botanica	Sarmizegetusa-24	1	2	20000
11	Cekani	Industriala-16	1	9	25000
22	Centru	Tigina-25	1	5	29650
30	Rishkanovka	Ceucari-22	1	14	29900
23	Buiucani	Sucevita-22	1	4	35600
34	Cekani	Maria_Dragan-8	1	3	39600
8	Centru	Bucuresti-3	2	4	38000
15	Centru	Bucuresti-72	2	4	41820
31	Botanica	Prigoreni-12	2	3	41900
24	Rishkanovka	Timisoara-18	2	11	49900
49	Rishkanovka	Petru_Zadniru-39	2	15	54100
47	Botanica	Leuseni-4	2	13	61000
36	Rishkanovka	Andrei_Lupan-72	2	8	72300
33	Centru	Tigina-56	3	15	35200
12	Buiucani	Suciava-4	3	7	59000
9	Rishkanovka	Doina-10	3	1	75000
17	Buiucani	Milano-55	3	2	76200
48	Cekani	Ilie_Replin-2	3	14	78900
44	Rishkanovka	Alecu_Russo-115	3	1	86200
6	Botanica	Nicolae_Zelinski	4	11	89000
32	Centru	Bucuresti-11	4	11	91200
18	Cekani	Butuceni-18	4	1	96400
45	Centru	Columna-54	4	2	97500
39	Cekani	Mesterul_Manole-20	4	5	99600
40	Centru	Bucuresti-17	5	4	105400

Для продолжения нажмите любую клавишу . . .

Скриншот 33

Описание: данная функция выводит на экран свободные квартиры в порядке возрастания количества комнат. Сначала я проверил то, существует ли файл и не пустой ли он. Если файл не существует или пустой, то выводится информация об этом. Далее я создал новый массив структур аренд, в который считал всю информацию об истекших арендах. Далее я создал новый массив структур квартир и присвоил в него номера квартир для дальнейшего поиска. Далее я сравнивал коды истекших квартир с кодами квартир и, если коды были равны, считывал данные о квартирах в новый массив структур. Далее я отсортировал методом вставок квартиры сначала по количеству комнат. А позже если количество комнат было равно, сортировал методом вставок по возрастанию цены и вывел на экран.

Фрагмент кода:

```
void Sapartments(){ // функция для поиска свободных квартир
```

```

rent *m=new rent[nR]; // создание массива структур аренд

ifstream g("Expire.txt"); // открытие файла с истекшими арендами для чтения
int nS=0; // создание переменной для подсчёта количества истекших аренд
if(g){ // Проверка на существование файла
if(g.peek()!=EOF){ // Проверка на то, не пустой ли файл
while(!g.eof()){ // Пока не конец файла
g>>m[nS].Kod>>m[nS].Fam >> m[nS].Name >> m[nS].Day>>m[nS].Month >> m[nS].Year
>> m[nS].SrokA; // считывание информации из файла
nS++; // увеличение счётчика
}
nS--; // уменьшение переменной количества истекших аренд на 1 для того, чтобы не
учитывать последний пробел в файле
g.close(); // закрытие файла

apartment *s= new apartment[nA]; // создание массива структур квартир

for(int i=0;i<nS;i++) s[i].Kod = m[i].Kod; // присваивание кодов истекших аренд в новый
массив

for(int i=0;i<nS;i++)
for(int j=0;j<nA;j++)
if(s[i].Kod == a[j].Kod) s[i]=a[j]; // если нашли код истекшей квартиры в массиве квартир,
считываем данные о квартире
for(int i=1; i<nS; i++){ // сортировка вставками
int j = i;
while (s[j].Kkomnat<s[j-1].Kkomnat and j>0){ // сортировка по возрастанию количества
комнат
swap(s[j],s[j-1]);
j--;
}
}

for(int i=1; i<nS; i++){ // сортировка вставками
int j = i;

```

```

while (s[j].Kkomnat==s[j-1].Kkomnat and s[j].Price< s[j-1].Price and j>0){ // если
количество комнат равно, сортируем по возрастанию цены
    swap(s[j],s[j-1]);
    j--;
}
}

setcolor(0,15); // смена цвета заднего фона на белый


cout<<endl<<"  Код  " <<"  Район  " <<"  Адрес  " <<"  Кол-во " <<"
Этаж " <<"  Цена  " <<endl;

cout<<" Квартиры                комнат                " <<endl; // Создание
шапки таблицы


for(int i=0;i<nS;i++){
    if(i%2==0) setcolor(0,8); // Проверка для смены цвета через строку. Смена цвета на тёмно-
серый
    else setcolor(0,7); // Смена цвета на светло-серый
    if(s[i].Kod==s[i+1].Kod)i++; // если код квартиры равен следующему, то увеличиваем
счётчик цикла на 1 и переходим на следующую строку
    else cout <<left<<"  " <<setw(11)<<s[i].Kod <<setw(19)<< s[i].Raion <<setw(28) <<s[i].Adres
<<setw(8)<<s[i].Kkomnat <<setw(8)<<s[i].Etaj <<s[i].Price<<endl; // в другом случае выводим
информацию
}

setcolor(14,0); // смена цвета текста на жёлтый
} else { // в другом случае
    setcolor(12,0); // смена цвета текста на красный
    cout <<"Нет свободных квартир." << endl;
setcolor(14,0); // смена цвета текста на жёлтый
}

} else { // в другом случае
    setcolor(12,0); // смена цвета текста на красный
    cout <<"Нет файла с свободными квартирами." << endl;
setcolor(14,0); // смена цвета текста на жёлтый
}
}
}

```

Квартира, пользующаяся наибольшим спросом

Текст задания:

Вывод на экран информации квартире, пользующейся наибольшим спросом.

Скриншот результата на экране:

Код Квартиры	Район	Адрес	Кол-во комнат	Этаж	Цена
33	Centru	Tigina-56	3	15	35200
Для продолжения нажмите любую клавишу . . .					

Скриншот 34

Описание: данная функция выводит информацию о квартире, которая пользуется наибольшим спросом. Для начала я создал и скопировал структуру аренд в новую структуру. Далее отсортировал по возрастанию кода квартиры. Далее создал переменную максимума и позиции. И в цикле если коды квартиры были равны, увеличивал счётчик. Если максимум был меньше счётчика, присваивал переменным максимум и позицию счётчика. Далее я выводил шапку таблицы и одну квартиру. Далее я искал повторения(если есть квартиры с одинаковым спросом). Для этого я создал массив и счётчик для удаления повторений. Далее в цикле, если коды квартир были равны увеличивал счётчик. После этого я сравнивал максимум с счётчиком и с кодом первой найденной квартиры. Если данные подходили, записывал в массив позиции элементов. Далее, если в массиве коды квартир были равны, то удалял повторения и уменьшал счётчик количества элементов этого массива. Далее я в цикле сравнивал коды квартир и, если находил такой код, выводил информацию на экран.

Фрагмент кода:

```
void Spros(){ // функция для поиска квартиры, пользующейся наибольшим спросом
rent *m=r; // создаём и копируем структуру аренды в новую структуру
int k; // создание счётчика
    for(int i=1; i<nR; i++){ // сортировка вставками
int j = i;
    while (m[j].Kod<m[j-1].Kod and j>0){ // сортировка по возрастанию кода квартиры
        swap(m[j],m[j-1]);
        j--;
    }
}
```



```

int max=0,pos; // создание переменных для максимума и позиции
for(int i=0;i<nR;i++){
    k=0; // обнуляем счётчик
    for(int j=0;j<nR;j++)
        if(m[i].Kod == m[j].Kod) k++; // если коды квартир равны, то увеличиваем счётчик

    if(max<k){ // если максимум меньше счётчика
        max=k; // то максимум равен счётчику
        pos=i; // позиция равна i(счётчик цикла)
    }
}

for(int i=0;i<nA;i++)
    if(m[pos].Kod==a[i].Kod){ // если код квартиры структуры с позицией pos равен коду
        // квартиры структуры квартир с позицией i
        setcolor(0,15); // смена цвета заднего фона на белый

        cout<<endl<<"    Код " <<"    Район    " <<"    Адрес    " <<"    Кол-во " <<"
Этаж " <<"    Цена    " <<endl;

        cout<<"    Квартиры                комнат                " <<endl; // Создание
шапки таблицы

        setcolor(0,8); // смена цвета текста на тёмно-серый
        cout <<left<<"    " <<setw(11)<<a[i].Kod <<setw(19)<<a[i].Raion <<setw(28)<<a[i].Adres
<<setw(8)<<a[i].Kkomnat <<setw(8)<<a[i].Etaj <<a[i].Price<<endl; // Вывод информации о
квартирах

    }

int *pov=new int[nA]; // создаём новый массив для удаления повторений
int g=0; // создаём счётчик для удаления повторений

for(int i=0;i<nR;i++){

```

```

    k=0; // обнуляем счётчик
    for(int j=0;j<nR;j++)
        if(m[i].Kod == m[j].Kod) k++; // если коды квартир равны, то увеличиваем счётчик

    if(max==k and m[i].Kod!=m[pos].Kod) // если максимум равен счётчику и код не равен
        коду сравнения
        pov[g++]=i; // записываем в массив позиции

    }
    if(g>0){
    for(int i=0;i<g+1;i++)
        if(m[pov[i]].Kod == m[pov[i+1]].Kod){ // если коды квартиры равны
            for(int j=i; j<nA; j++)
                m[j]=m[j+1]; // сдвиг массива квартир влево(удаление)
            g--; //уменьшение счётчика
        }

    for(int i=0;i<g;i++)
        for(int c=0;c<nA;c++){
            if(i%2==0) setcolor(0,7); // Проверка для смены цвета через строку. Смена цвета
            текста на тёмно-серый
            else setcolor(0,8); // Смена цвета на светло-серый

            if(m[pov[i]].Kod==a[c].Kod) // если код квартиры структуры с позицией pov[i] равен коду
            квартиры структуры квартир с позицией c
                cout <<left<<"    "<<setw(11)<<a[c].Kod <<setw(19)<< a[c].Raion <<setw(28) <<a[c].Adres
                <<setw(8)<<a[c].Kkomnat <<setw(8)<<a[c].Etaaj <<a[c].Price<<endl; // Вывод информации о
                квартирах
            }
        }
    cout<<endl;
    setcolor(14,0); // смена цвета текста на жёлтый
    }

```

Средняя цена всех квартир с указанным количеством комнат

Текст задания:

Вывод на экран средней цены всех квартир с указанным количеством комнат.

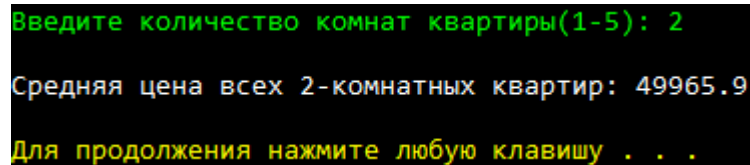
Скриншот вводимых данных:



```
Введите количество комнат квартиры(1-5): 2
```

Скриншот 35

Скриншот результата на экране:



```
Введите количество комнат квартиры(1-5): 2
Средняя цена всех 2-комнатных квартир: 49965.9
Для продолжения нажмите любую клавишу . . .
```

Скриншот 36

Описание: данная функция выводит на экран среднюю цену всех квартир с указанным количеством комнат. Для начала я создал переменные для количества комнат, счётчика квартир и средней цены. Далее я искал максимальную комнату, для защиты от неверно введённых данных. Далее пользователь вводил количество комнат. И в цикле если количество комнат квартиры равно введённому количеству комнат, то суммируем цены квартир и увеличиваем счётчик квартир. После цикла делим сумму цен квартир на количество квартир и выводим информацию.

Фрагмент кода:

```
void Srprice() { // функция для поиска средней цены всех квартир с указанным количеством комнат
```

```
    int kkomnat,k; // создание переменной для количества комнат и счётчика
    float Srsum; // создание переменной для средней цены
    int max=0; // создание переменной для вычисления максимальной комнаты
    for(int i=0;i<nA;i++)
        if(max<a[i].Kkomnat)max=a[i].Kkomnat; // поиск максимальной комнаты
    setcolor(10,0); // смена цвета текста на зелёный
    cout<<"\nВведите количество комнат квартиры(1-"<<max<<"): ";
    do{ // защита от неправильного ввода
        cin>> kkomnat; // ввод количества комнат
        if(kkomnat<1 or kkomnat>max){
```

```

        setcolor(12,0); // смена цвета текста на красный
cout<<"Ошибка! Попробуйте ещё раз! "<<endl;
        setcolor(10,0); // смена цвета текста на зелёный
        cout<<"\nВведите количество комнат квартиры(1-"<<max<<"): ";
    }
} while(kkomnat<1 or kkomnat>max); // защита от неправильного ввода


for(int i=0;i<nA;i++)
    if(a[i].Kkomnat== kkomnat){ // если количество комнат квартиры равно введённому
количеству комнат
        Srsum+=a[i].Price; // то суммируем цены квартир
        k++; // увеличиваем счётчик
    }
Srsum/=k; // делим сумму цен квартир на количество
        setcolor(15,0); // смена цвета текста на белый
        cout<<"\nСредняя цена всех " << kkomnat <<"-комнатных квартир: " <<Srsum<<
endl<<endl;
        setcolor(14,0); // смена цвета текста на жёлтый
    }
}

```

Вспомогательные функции

Смена цвета

Описание: данная функция нужна для того, чтобы менять цвет текста и заднего фона. Благодаря этой функции, не нужно писать длинную строку для смены цвета.

Фрагмент кода

```
void setcolor(int text, int backG=0){ // функция для изменения цвета
HANDLE color = GetStdHandle(STD_OUTPUT_HANDLE); // дескриптор устройства
стандартного вывода.
SetConsoleTextAttribute(color,(WORD)((backG << 4)| text)); // задаем цвет фона и цвет текста
}
```

Управление мышкой в главном меню

Описание: данная функция нужна для того, чтобы была возможность пользоваться мышкой в главном меню. С помощью координат задаются кнопки и при нажатии на них функция возвращает одно из значений.

Фрагмент кода

```
int mouse(){ // Функция для мыши в главном меню

HANDLE hin = GetStdHandle(STD_INPUT_HANDLE); // получаем дескриптор
INPUT_RECORD InputRecord; // используется для возвращения информации о входных
сообщениях в консольном входном буфере
DWORD Events; // unsigned long
COORD; // для координат X, Y

/*Запрещение выделения консоли*/
DWORD prev_mode;
GetConsoleMode(hin, &prev_mode);
SetConsoleMode(GetStdHandle(STD_INPUT_HANDLE), ENABLE_EXTENDED_FLAGS |
(prev_mode & ~ENABLE_QUICK_EDIT_MODE));
/*Запрещение выделения консоли*/

SetConsoleMode(hin, ENABLE_MOUSE_INPUT); // разрешаем обработку мыши
```

```

while (true)
{
    ReadConsoleInput(hin, &InputRecord, 1, &Events); // считывание
    if (InputRecord.Event.MouseEvent.dwButtonState ==
FROM_LEFT_1ST_BUTTON_PRESSED) // если нажата левая кнопка мыши
    {
        coord.X = InputRecord.Event.MouseEvent.dwMousePosition.X; // считывание координат
X
        coord.Y = InputRecord.Event.MouseEvent.dwMousePosition.Y; // считывание координат
Y
        //cout << "left - X" << coord.X << ", Y = " << coord.Y << endl;

        if(coord.X>=45 and coord.X<=72 and coord.Y==2){ // задаём координаты кнопок для
нажатия
            prev_mode = 487; // присваиваем стандартный код консоли(для работы с
клавиатурой и т.д)
            SetConsoleMode(hin,prev_mode); // возвращение стандартного режима консоли
            return 1; // возвращаем нужное значение

        }else if(coord.X>=46 and coord.X<=71 and coord.Y==3){ // задаём координаты кнопок
для нажатия
            prev_mode = 487; // присваиваем стандартный код консоли(для работы с
клавиатурой и т.д)
            SetConsoleMode(hin,prev_mode); // возвращение стандартного режима консоли
            return 2; // возвращаем нужное значение

        }else if(coord.X>=37 and coord.X<=80 and coord.Y==4){ // задаём координаты кнопок
для нажатия
            prev_mode = 487; // присваиваем стандартный код консоли(для работы с
клавиатурой и т.д)
            SetConsoleMode(hin,prev_mode); // возвращение стандартного режима консоли
            return 3; // возвращаем нужное значение

```

```

} else if(coord.X>=38 and coord.X<=79 and coord.Y==5){ // задаём координаты кнопок
для нажатия

    prev_mode = 487; // присваиваем стандартный код консоли(для работы с
клавиатурой и т.д)

    SetConsoleMode(hin,prev_mode); // возвращение стандартного режима консоли
    return 4; // возвращаем нужное значение

} else if(coord.X>=25 and coord.X<=93 and coord.Y==6){ // задаём координаты кнопок
для нажатия

    prev_mode = 487; // присваиваем стандартный код консоли(для работы с
клавиатурой и т.д)

    SetConsoleMode(hin,prev_mode); // возвращение стандартного режима консоли
    return 5; // возвращаем нужное значение

} else if(coord.X>=25 and coord.X<=93 and coord.Y==7){ // задаём координаты кнопок
для нажатия

    prev_mode = 487; // присваиваем стандартный код консоли(для работы с
клавиатурой и т.д)

    SetConsoleMode(hin,prev_mode); // возвращение стандартного режима консоли
    return 6; // возвращаем нужное значение

} else if(coord.X>=22 and coord.X<=96 and coord.Y==8){ // задаём координаты кнопок
для нажатия

    prev_mode = 487; // присваиваем стандартный код консоли(для работы с
клавиатурой и т.д)

    SetConsoleMode(hin,prev_mode); // возвращение стандартного режима консоли
    return 7; // возвращаем нужное значение

} else if(coord.X>=22 and coord.X<=96 and coord.Y==9){ // задаём координаты кнопок
для нажатия

    prev_mode = 487; // присваиваем стандартный код консоли(для работы с
клавиатурой и т.д)

    SetConsoleMode(hin,prev_mode); // возвращение стандартного режима консоли
    return 8; // возвращаем нужное значение

```

```
}else if(coord.X>=17 and coord.X<=101 and coord.Y==10){ // задаём координаты кнопок  
для нажатия
```

```
    prev_mode = 487; // присваиваем стандартный код консоли(для работы с  
клавиатурой и т.д)
```

```
    SetConsoleMode(hin,prev_mode); // возвращение стандартного режима консоли
```

```
    return 9; // возвращаем нужное значение
```

```
}else if(coord.X>=24 and coord.X<=93 and coord.Y==11){ // задаём координаты кнопок  
для нажатия
```

```
    prev_mode = 487; // присваиваем стандартный код консоли(для работы с  
клавиатурой и т.д)
```

```
    SetConsoleMode(hin,prev_mode); // возвращение стандартного режима консоли
```

```
    return 10; // возвращаем нужное значение
```

```
}else if(coord.X>=35 and coord.X<=81 and coord.Y==12){ // задаём координаты кнопок  
для нажатия
```

```
    prev_mode = 487; // присваиваем стандартный код консоли(для работы с  
клавиатурой и т.д)
```

```
    SetConsoleMode(hin,prev_mode); // возвращение стандартного режима консоли
```

```
    return 11; // возвращаем нужное значение
```

```
}else if(coord.X>=28 and coord.X<=89 and coord.Y==13){ // задаём координаты кнопок  
для нажатия
```

```
    prev_mode = 487; // присваиваем стандартный код консоли(для работы с  
клавиатурой и т.д)
```

```
    SetConsoleMode(hin,prev_mode); // возвращение стандартного режима консоли
```

```
    return 12; // возвращаем нужное значение
```

```
}else if(coord.X>=53 and coord.X<=63 and coord.Y==14){ // задаём координаты кнопок  
для нажатия
```

```
    prev_mode = 487; // присваиваем стандартный код консоли(для работы с  
клавиатурой и т.д)
```

```
    SetConsoleMode(hin,prev_mode); // возвращение стандартного режима консоли
```



```

        return 13; // возвращаем нужное значение

    }

    } else if (InputRecord.Event.MouseEvent.dwButtonState ==
RIGHTMOST_BUTTON_PRESSED) // если нажата правая кнопка мыши
return 0; // возвращаем нужное значение

}

return 0; // возвращаем нужное значение

}

```

Управление мышкой в редактировании квартир

Описание: данная функция нужна для того, чтобы была возможность пользоваться мышкой при выборе того, что нужно редактировать в структуре квартир. С помощью координат задаются кнопки и при нажатии на них функция возвращает одно из значений.

Фрагмент кода

```

int swredA() { // Функция для мыши в редактировании квартир

    HANDLE hin = GetStdHandle(STD_INPUT_HANDLE); // получаем дескриптор
    INPUT_RECORD InputRecord; // используется для возвращения информации о входных
сообщениях в консольном входном буфере
    DWORD Events; // unsigned long
    COORD coord; // для координат X, Y

    /*Запрещение выделения консоли*/
    DWORD prev_mode;
    GetConsoleMode(hin, &prev_mode);
    SetConsoleMode(GetStdHandle(STD_INPUT_HANDLE), ENABLE_EXTENDED_FLAGS |
(prev_mode & ~ENABLE_QUICK_EDIT_MODE));

```

```
/*Запрещение выделения консоли*/
```

```
SetConsoleMode(hin, ENABLE_MOUSE_INPUT); // разрешаем обработку мыши
```

```
while (true)
```

```
{
```

```
    ReadConsoleInput(hin, &InputRecord, 1, &Events); // считывание
```

```
    if (InputRecord.Event.MouseEvent.dwButtonState ==  
FROM_LEFT_1ST_BUTTON_PRESSED){ // если нажата левая кнопка мыши  
        coord.X = InputRecord.Event.MouseEvent.dwMousePosition.X; // считывание координат  
X  
        coord.Y = InputRecord.Event.MouseEvent.dwMousePosition.Y; // считывание координат  
Y  
        // cout << "left - X" << coord.X << ", Y = " << coord.Y << endl;
```

```
        if(coord.X>=7 and coord.X<=17 and coord.Y==3){ // задаём координаты кнопок для  
нажатия
```

```
            prev_mode = 487; // присваиваем стандартный код консоли(для работы с  
клавиатурой и т.д)
```

```
            SetConsoleMode(hin,prev_mode); // возвращение стандартного режима консоли
```

```
            return 1; // возвращаем нужное значение
```

```
        }else if(coord.X>=20 and coord.X<=30 and coord.Y==3){ // задаём координаты кнопок  
для нажатия
```

```
            prev_mode = 487;
```

```
            SetConsoleMode(hin,prev_mode); // возвращение стандартного режима консоли
```

```
            return 2; // возвращаем нужное значение
```

```
        }else if(coord.X>=33 and coord.X<=55 and coord.Y==3){ // задаём координаты кнопок  
для нажатия
```

```
            prev_mode = 487; // присваиваем стандартный код консоли(для работы с  
клавиатурой и т.д)
```

```
            SetConsoleMode(hin,prev_mode); // возвращение стандартного режима консоли
```

```

    return 3; // возвращаем нужное значение

} else if(coord.X>=58 and coord.X<=67 and coord.Y==3){ // задаём координаты кнопок
для нажатия

    prev_mode = 487; // присваиваем стандартный код консоли(для работы с
клавиатурой и т.д)

    SetConsoleMode(hin,prev_mode); // возвращение стандартного режима консоли
    return 4; // возвращаем нужное значение

} else if(coord.X>=70 and coord.X<=79 and coord.Y==3){ // задаём координаты кнопок
для нажатия

    prev_mode = 487; // присваиваем стандартный код консоли(для работы с
клавиатурой и т.д)

    SetConsoleMode(hin,prev_mode); // возвращение стандартного режима консоли
    return 5; // возвращаем нужное значение

} else if(coord.X>=79 and coord.X<=88 and coord.Y==3){ // задаём координаты кнопок
для нажатия

    prev_mode = 487; // присваиваем стандартный код консоли(для работы с
клавиатурой и т.д)

    SetConsoleMode(hin,prev_mode); // возвращение стандартного режима консоли
    return 6; // возвращаем нужное значение

} else if(coord.X>=82 and coord.X<=90 and coord.Y==3){ // задаём координаты кнопок
для нажатия

    prev_mode = 487; // присваиваем стандартный код консоли(для работы с
клавиатурой и т.д)

    SetConsoleMode(hin,prev_mode); // возвращение стандартного режима консоли
    return 7; // возвращаем нужное значение

} else if(coord.X>=93 and coord.X<=104 and coord.Y==3){ // задаём координаты кнопок
для нажатия

    prev_mode = 487; // присваиваем стандартный код консоли(для работы с
клавиатурой и т.д)

```

```

        SetConsoleMode(hin, prev_mode); // возвращение стандартного режима консоли
        return 0; // возвращаем нужное значение
    }

    }else if (InputRecord.Event.MouseEvent.dwButtonState ==
RIGHTMOST_BUTTON_PRESSED) // если нажата правая кнопка мыши
return 0; // возвращаем нужное значение

    }

return 0; // возвращаем нужное значение

}

```

Управление мышкой в редактировании аренд

Описание: данная функция нужна для того, чтобы была возможность пользоваться мышкой при выборе того, что нужно редактировать в структуре аренд. С помощью координат задаются кнопки и при нажатии на них функция возвращает одно из значений.

Фрагмент кода

```

int swredR() { // Функция для мыши в редактировании аренд

    HANDLE hin = GetStdHandle(STD_INPUT_HANDLE); // получаем дескриптор
    INPUT_RECORD InputRecord; // используется для возвращения информации о входных
сообщениях в консольном входном буфере
    DWORD Events; // unsigned long
    COORD coord; // для координат X, Y

    /*Запрещение выделения консоли*/
    DWORD prev_mode;
    GetConsoleMode(hin, &prev_mode);

```

```

    SetConsoleMode(GetStdHandle(STD_INPUT_HANDLE), ENABLE_EXTENDED_FLAGS |
(prev_mode & ~ENABLE_QUICK_EDIT_MODE));
/*Запрещение выделения консоли*/

    SetConsoleMode(hin, ENABLE_MOUSE_INPUT); // разрешаем обработку мыши

    while (true)
    {
        ReadConsoleInput(hin, &InputRecord, 1, &Events); // считывание

        if (InputRecord.Event.MouseEvent.dwButtonState ==
FROM_LEFT_1ST_BUTTON_PRESSED) // если нажата левая кнопка мыши
        {
            coord.X = InputRecord.Event.MouseEvent.dwMousePosition.X; // считывание координат
X
            coord.Y = InputRecord.Event.MouseEvent.dwMousePosition.Y; // считывание координат
Y
            //cout << "left - X" << coord.X << ", Y = " << coord.Y << endl;

            if(coord.X>5 and coord.X<=17 and coord.Y==3){ // задаём координаты кнопок для
нажатия
                prev_mode = 487; // присваиваем стандартный код консоли(для работы с
клавиатурой и т.д)
                SetConsoleMode(hin,prev_mode); // возвращение стандартного режима консоли
                return 1; // возвращаем нужное значение

            }else if(coord.X>=20 and coord.X<=28 and coord.Y==3){ // задаём координаты кнопок
для нажатия
                prev_mode = 487; // присваиваем стандартный код консоли(для работы с
клавиатурой и т.д)
                SetConsoleMode(hin,prev_mode); // возвращение стандартного режима консоли
                return 2; // возвращаем нужное значение

```

```

}else if(coord.X>=31 and coord.X<=40 and coord.Y==3){ // задаём координаты кнопок
для нажатия

    prev_mode = 487; // присваиваем стандартный код консоли(для работы с
клавиатурой и т.д)

    SetConsoleMode(hin,prev_mode); // возвращение стандартного режима консоли
    return 3; // возвращаем нужное значение

}else if(coord.X>=43 and coord.X<=53 and coord.Y==3){ // задаём координаты кнопок
для нажатия

    prev_mode = 487; // присваиваем стандартный код консоли(для работы с
клавиатурой и т.д)

    SetConsoleMode(hin,prev_mode); // возвращение стандартного режима консоли
    return 4; // возвращаем нужное значение

}else if(coord.X>=56 and coord.X<=64 and coord.Y==3){ // задаём координаты кнопок
для нажатия

    prev_mode = 487; // присваиваем стандартный код консоли(для работы с
клавиатурой и т.д)

    SetConsoleMode(hin,prev_mode); // возвращение стандартного режима консоли
    return 5; // возвращаем нужное значение

}else if(coord.X>=67 and coord.X<=83 and coord.Y==3){ // задаём координаты кнопок
для нажатия

    prev_mode = 487; // присваиваем стандартный код консоли(для работы с
клавиатурой и т.д)

    SetConsoleMode(hin,prev_mode); // возвращение стандартного режима консоли
    return 6; // возвращаем нужное значение

}else if(coord.X>=86 and coord.X<=94 and coord.Y==3){ // задаём координаты кнопок
для нажатия

    prev_mode = 487; // присваиваем стандартный код консоли(для работы с
клавиатурой и т.д)

    SetConsoleMode(hin,prev_mode); // возвращение стандартного режима консоли
    return 7; // возвращаем нужное значение

```

```

        }else if(coord.X>=97 and coord.X<=108 and coord.Y==3){ // задаём координаты кнопок
для нажатия

        prev_mode = 487; // присваиваем стандартный код консоли(для работы с
клавиатурой и т.д)

        SetConsoleMode(hin,prev_mode); // возвращение стандартного режима консоли
        return 0; // возвращаем нужное значение

    }

    }else if (InputRecord.Event.MouseEvent.dwButtonState ==
RIGHTMOST_BUTTON_PRESSED)
return 0; // возвращаем нужное значение

    }

return 0; // возвращаем нужное значение

}

```

Управление мышкой в главном меню

Описание: данная функция нужна для того, чтобы была возможность пользоваться мышкой при выборе открывать или не открывать текстовый файл с истекшими арендами. С помощью координат задаются кнопки и при нажатии на них функция возвращает одно из значений.

Фрагмент кода

```
int swopenFile(){ // Функция для мыши в задании, для открытия файла
```

```

HANDLE hin = GetStdHandle(STD_INPUT_HANDLE); // получаем дескриптор
INPUT_RECORD InputRecord; // используется для возвращения информации о входных
сообщениях в консольном входном буфере
DWORD Events; // unsigned long
COORD coord; // для координат X, Y

```

```

/*Запрещение выделения консоли*/
    DWORD prev_mode;
    GetConsoleMode(hin, &prev_mode);
    SetConsoleMode(GetStdHandle(STD_INPUT_HANDLE), ENABLE_EXTENDED_FLAGS |
    (prev_mode & ~ENABLE_QUICK_EDIT_MODE));
/*Запрещение выделения консоли*/

    SetConsoleMode(hin, ENABLE_MOUSE_INPUT); // разрешаем обработку мыши

    while (true)
    {
        ReadConsoleInput(hin, &InputRecord, 1, &Events); // считывание

        if (InputRecord.Event.MouseEvent.dwButtonState ==
        FROM_LEFT_1ST_BUTTON_PRESSED) // если нажата левая кнопка мыши
        {
            coord.X = InputRecord.Event.MouseEvent.dwMousePosition.X; // считывание координат
            X
            coord.Y = InputRecord.Event.MouseEvent.dwMousePosition.Y; // считывание координат
            Y

            //cout << "left - X" << coord.X << ", Y = " << coord.Y << endl;

            if(coord.X>60 and coord.X<=67 and coord.Y==2){ // задаём координаты кнопок для
            нажатия
                prev_mode = 487; // присваиваем стандартный код консоли(для работы с
                клавиатурой и т.д)
                SetConsoleMode(hin,prev_mode); // возвращение стандартного режима консоли
                return 1; // возвращаем нужное значение

            }else if(coord.X>=71 and coord.X<=79 and coord.Y==2){ // задаём координаты кнопок
            для нажатия
                prev_mode = 487; // присваиваем стандартный код консоли(для работы с
                клавиатурой и т.д)

```



```

        SetConsoleMode(hin,prev_mode); // возвращение стандартного режима консоли
        return 0; // возвращаем нужное значение
    }

    }else if (InputRecord.Event.MouseEvent.dwButtonState ==
RIGHTMOST_BUTTON_PRESSED)
return 0; // возвращаем нужное значение

    }

return 0; // возвращаем нужное значение

}

```

Считывание информации о квартирах

Описание: эта функция играет важную роль в работе программы, так как она изначально считывает информацию из файла с информацией о квартирах и записывает её в структуру квартир для дальнейшего использования.

Фрагмент кода

```

void readA(){ // Функция для считывания информации о квартирах с файла
    ifstream f("Apartment.txt"); // Открытие файла для чтения
    if(f){ // Проверка на существование файла
        if(f.peek()!=EOF){ // Проверка на то, не пустой ли файл
            int i=0; // Счётчик для количества квартир
            while(!f.eof()){ // Пока не конец файла
                f>>a[i].Kod>>a[i].Raion >> a[i].Adres >> a[i].Kkomnat>>a[i].Etaj >> a[i].Price; //

```

Считывание данных с файла

```

        i++; // Увеличение счётчика
    }

    nA=i-1; // Присваивание переменной количества квартир(-1 для того, чтобы не учитывать
последний пробел в файле)
    f.close(); // Закрытие файла
}else{ // в другом случае

```

```

        setcolor(12,0); // Смена цвета на красный
cout <<"Файл с информацией о квартирах пустой."<< endl;
}
}else{ // в другом случае
        setcolor(12,0); // Смена цвета на красный
cout <<"Файл с информацией о квартирах не найден."<< endl;
}
setcolor(14,0); // Смена цвета на жёлтый

}

```

Считывание информации об арендах

Описание: эта функция играет важную роль в работе программы, так как она изначально считывает информацию из файла с информацией о арендах и записывает её в структуру аренд для дальнейшего использования.

Фрагмент кода

```

void readR(){ // Функция для считывания информации об арендах с файла
    ifstream g("Rent.txt"); // Открытие файла для чтения
    int i=0; // Счётчик для количества аренд
    if(g){ // Проверка на существование файла
        if(g.peek()!=EOF){ // Проверка на то, не пустой ли файл
            while(!g.eof()){ // Пока не конец файла
                g>>r[i].Kod>>r[i].Fam >> r[i].Name >> r[i].Day>>r[i].Month >> r[i].Year >> r[i].SrokA; //

```

Считывание данных с файла

```

        i++; // Увеличение счётчика
    }
    nR=i-1; // Присваивание переменной количества аренд(-1 для того, чтобы не учитывать
последний пробел в файле)
    g.close(); // Закрытие файла
}else{ // в другом случае
        setcolor(12,0); // Смена цвета на красный
        cout <<"Файл с информацией об аренде пустой."<< endl;
    }
}else{ // в другом случае

```

```

        setcolor(12,0); // Смена цвета на красный
        cout <<"Файл с информацией об аренде не найден."<< endl;
    }
    setcolor(14,0); // Смена цвета на жёлтый

}

```

Перезапись файла квартир

Описание: данная функция нужна для того, чтобы при каких-либо изменениях, удалениях и так далее, записывать изменённую информацию о квартирах обратно в файл. Эта функция является одной из самых важных, так как без неё данные бы не записывались обратно в файл и текстовый файл с квартирами оставался бы неизменным.

Фрагмент кода

```

void rewriteInfoA(){ // Функция для перезаписи файла квартир
    ofstream f("Apartment.txt"); // открытие файла квартир для записи
    for(int i=0; i<nA; i++)
        f << a[i].Kod <<" " << a[i].Raion <<" " << a[i].Adres<<" " << a[i].Kkomnat<<" " <<a[i].Etaj<<
" " << a[i].Price<< endl; // перезапись всех квартир в файл
    f.close(); // закрытие файла
}

```

Перезапись файла аренд

Описание: данная функция нужна для того, чтобы при каких-либо изменениях, удалениях и так далее, записывать изменённую информацию об арендах обратно в файл. Эта функция является одной из самых важных, так как без неё данные бы не записывались обратно в файл и текстовый файл с арендами оставался бы неизменным.

Фрагмент кода

```

void rewriteInfoR(){ // Функция для перезаписи файла аренд
    ofstream f("Rent.txt"); // открытие файла аренд для записи
    for(int i=0; i<nR; i++)
        f << r[i].Kod <<" " << r[i].Fam <<" " << r[i].Name<<" " << r[i].Day <<" " << r[i].Month <<"
" << r[i].Year <<" " << r[i].SrokA<< endl; // перезапись всех аренд в файл
    f.close(); // закрытие файла
}

```

Microsoft Excel:


В моем проекте книга Microsoft Excel содержит заставку, которая появляется при открытии книги.



Моя книга содержит 2 взаимосвязанные таблицы с информацией об квартирах и арендах, которые находятся на листах «Apartment» и «Rent», при изменении данных в одной таблице, они будут влиять на данные в другой.

Apartment

Таблица №1 Квартиры

Код квар тиры	Район	Адрес	Количе ство комнат	Этаж	Цена	
1	Centru	Ismail-26	2	1	45 000 €	
2	Buiucani	Milano-12	3	3	74 000 €	

Код квар тиры	Район	Адрес	Количе ство комнат	Этаж	Цена	
3	Rishkanovka	Andrei_Lupan-12	2	2	47 000 €	
4	Botanica	Sarmizegetusa-24	1	2	20 000 €	
5	Rishkanovka	Doina-3	2	9	31 000 €	
6	Botanica	Nicolae_Zelinski	4	11	89 000 €	
7	Cekani	Sargidava-4	1	7	25 000 €	
8	Centru	Bucuresti-3	2	4	38 000 €	

Код квар тиры	Район	Адрес	Количе ство комнат	Этаж	Цена	
9	Rishkanovka	Doina-10	3	1	75 000 €	
10	Botanica	Sarmizegetusa-11	2	14	47 000 €	
11	Cekani	Industrialia-16	1	9	25 000 €	
12	Buiucani	Suciava-4	3	7	59 000 €	
13	Botanica	Burebista	2	2	46 000 €	
14	Cekani	Otovasca-16	1	15	27 600 €	

Код квар тиры	Район	Адрес	Количе ство комнат	Этаж	Цена	
15	Centru	Bucuresti-72	2	4	41 820 €	
16	Buiucani	Ion_Creanga-11	5	7	114 200 €	
17	Buiucani	Milano-55	3	2	76 200 €	
18	Cekani	Butuceni-18	4	1	96 400 €	
19	Rishkanovka	Criuleni-1	1	8	29 600 €	
20	Botanica	Padurii-44	2	7	52 000 €	

Код квар тиры	Район	Адрес	Количе ство комнат	Этаж	Цена	
21	Rishkanovka	Doina-45	4	6	76 900 €	
22	Centru	Tigina-25	1	5	29 650 €	
23	Buiucani	Sucevita-22	1	4	35 600 €	
24	Rishkanovka	Timisoara-18	2	11	49 900 €	
25	Centru	Ismail-47	3	12	67 500 €	
26	Buiucani	Alba_Iulia-22	4	15	89 600 €	

Код квар тиры	Район	Адрес	Количе ство комнат	Этаж	Цена	
27	Botanica	Gradina_Botanica-19	3	8	72 000 €	
28	Centru	Columna-13	5	4	156 000 €	
29	Cekani	Otovasca-19	2	8	41 600 €	
30	Rishkanovka	Ceucari-22	1	14	29 900 €	
31	Botanica	Prigoreni-12	2	3	41 900 €	
32	Centru	Bucuresti-11	4	11	91 200 €	

Код квар тиры	Район	Адрес	Количе ство комнат	Этаж	Цена	
33	Centru	Tigina-56	3	15	35 200 €	
34	Cekani	Maria_Dragan-8	1	3	39 600 €	
35	Buiucani	Industrialala-48	2	4	69 000 €	
36	Rishkanovka	Andrei_Lupan-72	2	8	72 300 €	
37	Centru	Alexei_Mateevici-81	1	9	31 600 €	
38	Botanica	Simeon_Murafa-12	2	7	54 200 €	

Код квар тиры	Район	Адрес	Количе ство комнат	Этаж	Цена	
39	Cekani	Mesterul_Manole- 20	4	5	99 600 €	
40	Centru	Bucuresti-17	5	4	105 400 €	
41	Buiucani	Industrialala-55	1	6	29 900 €	
42	Centru	Alexei_Sciusev-43	5	2	142 000 €	
43	Buiucani	George_Cosbuc	2	1	57 600 €	
44	Rishkanovka	Alecu_Russo-115	3	1	86 200 €	

Код квар тиры	Район	Адрес	Количе ство комнат	Этаж	Цена	
45	Centru	Columna-54	4	2	97 500 €	
46	Cekani	Sargidava-19	1	9	36 100 €	
47	Botanica	Leuseni-4	2	13	61 000 €	
48	Cekani	Ilie_Replin-2	3	14	78 900 €	
49	Rishkanovka	Petru_Zadnipru-39	2	15	54 100 €	
50	Centru	Serghei_Lazo-14	4	11	98 600 €	

Таблица №2 Аренды

Код квартиры	Фамилия арендатора	Имя арендатора	Дата аренды	Срок аренды	Сумма к оплате
1	Martinez	Lorenzo	30.04.2022	2	45 000 €
2	Patterson	Camden	13.07.2020	3	74 000 €
3	Scott	Quenten	04.12.2015	1	47 000 €
4	Brown	Vidal	25.07.2017	4	20 000 €
5	Gonzalez	Duke	18.11.2020	5	31 000 €
3	Howard	Henrik	13.05.2022	2	47 000 €
7	Promise	Alexander	06.11.2021	3	25 000 €
8	Xenos	Howard	06.08.2015	1	38 000 €
6	Williams	Natalie	09.04.2019	2	89 000 €
9	Irving	James	01.09.2020	1	75 000 €
3	Davis	Dean	10.03.2017	2	47 000 €
11	Johnson	Isaias	05.01.2018	3	25 000 €
10	Ferguson	Harold	19.11.2021	2	47 000 €
11	Martinez	Nikko	11.12.2022	4	25 000 €
13	Johnson	Lorenzo	15.04.2022	5	46 000 €
14	Gray	Marcus	01.05.2022	4	27 600 €
12	Jones	Karen	02.08.2020	1	59 000 €
15	Davis	Gordon	08.10.2018	3	41 820 €
16	Allen	Torin	12.01.2021	2	114 200 €
17	Brooks	Thaddeus	28.09.2019	1	76 200 €
18	King	Briggs	04.12.2018	1	96 400 €
19	Hernandez	Elian	14.01.2021	2	29 600 €
20	Nelson	Ezekiel	14.09.2021	2	52 000 €
21	Hernandez	Diego	09.02.2022	4	76 900 €
22	Moore	Malachi	09.09.2015	5	29 650 €
23	Scott	Liam	05.06.2020	2	35 600 €
24	Henderson	Zander	07.01.2015	1	49 900 €
25	Young	Gentry	16.09.2020	2	67 500 €
26	Perez	Nick	26.01.2021	4	89 600 €
27	Walker	Josiah	15.02.2022	4	72 000 €
28	Rogers	Qadry	19.08.2022	2	156 000 €
29	Barnes	Uziel	15.09.2022	1	41 600 €

Код квартиры	Фамилия арендатора	Имя арендатора	Дата аренды	Срок аренды	Сумма к оплате
24	Thomas	Jase	03.09.2018	6	49 900 €
31	Harris	Taj	22.04.2018	2	41 900 €
32	Powell	Keaton	26.12.2019	1	91 200 €
33	Hall	Enzo	19.10.2015	2	35 200 €
30	Ross	Miriam	22.12.2010	5	29 900 €
34	Hernandez	Trenton	11.02.2015	3	39 600 €
35	Hayes	Lewis	24.09.2020	3	69 000 €
36	Jenkins	Ruben	17.02.2015	2	72 300 €
37	Wright	Kenneth	29.04.2022	1	31 600 €
33	Young	Omar	29.01.2018	2	35 200 €
38	Hill	Ricky	30.05.2020	3	54 200 €
39	Taylor	Abel	10.12.2018	3	99 600 €
40	Torres	Ian	12.02.2018	4	105 400 €
41	Ward	Bennett	31.12.2018	5	29 900 €
42	Coleman	Soren	13.01.2018	5	142 000 €
33	Foster	Juan	30.11.2021	2	35 200 €
44	Johnson	Lorenzo	01.05.2021	1	86 200 €
45	Gray	Giovanni	06.05.2015	6	97 500 €
43	Vasquez	Jesus	15.01.2022	1	57 600 €
33	Rodriguez	Felipe	12.07.2020	1	35 200 €
47	Howard	Finn	06.02.2015	3	61 000 €
48	Harris	Urias	11.05.2018	1	78 900 €
49	Adams	Keith	05.11.2020	1	54 100 €
50	Lopez	Tripp	21.07.2022	4	98 600 €
Общая сумма всех аренд:					3 327 770 €

Таблица №3

Сектор	Количество квартир	Средняя цена
Centru	13	75 343,85 €
Buiucani	9	67 233,33 €
Botanica	9	53 677,78 €
Rishkanovka	10	55 190,00 €
Cekani	9	52 200,00 €

Данная таблица находится на листе «Статистика». В ней показаны количество квартир определенного сектора и их средняя цена. При изменении данных в таблицах с данными, данные также будут изменяться и здесь.

Таблица №4

Количество комнат	Количество квартир
1-комнатные	12
2-комнатные	17
3-комнатные	9
4-комнатные	8
5-комнатные	4

Данная таблица и гистограмма находятся на листе «Графическое представление». В данной таблице показаны количество квартир с определенным количеством комнат. При изменении данных в таблицах с данными, данные также будут изменяться и здесь. Также ниже представлена диаграмма на основе этих данных.



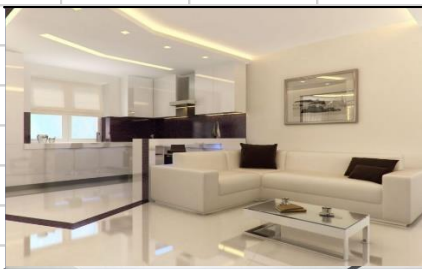
ComboBox

Таблица №5

33					
Код квартиры	Фамилия арендатора	Имя арендатора	Дата аренды	Срок аренды	Сумма к оплате
33	Hall	Enzo	19.10.2015	2	35 200 €
33	Young	Omar	29.01.2018	2	35 200 €
33	Foster	Juan	30.11.2021	2	35 200 €
33	Rodriguez	Felipe	12.07.2020	1	35 200 €

Данный динамический фильтр находится на листе «ComboBox». Этот динамический фильтр выводит данные об аренде по коду квартиры, который выбирается из раскрывающегося списка.

1				



Данный список с картинками находится на листе «Список с картинками». Он выводит фото квартиры по коду, который выбирается из раскрывающегося списка

Вывод

Мне очень понравилась практика в колледже, мне было интересно справляться с задачами, узнавать что-то новое. Во время практики мы получаем новые знания, можем дорабатывать программы на своё усмотрение.

Мне понравились предложенные задания, возможность использовать всё, что позволяет фантазия и нет никаких ограничений. Также мне понравилось работать со всеми вместе в одном компьютерном классе и узнавать что-то новое друг от друга.

Во время практики отрицательных моментов не возникало.

Были некоторые трудности с Excel из-за незнания некоторых формул.

В реализации некоторых заданий мне помогли интернет-ресурсы и одноклассники.

Было бы интересно выделить 1 день практики для того, чтобы посмотреть проекты других групп, посмотреть какие-то идеи и может взять что-то для себя.

Библиография

Брайан У. Керниган, Д. М. (2017). *Язык программирования С. Вильямс.*

Выпадающий список с картинками RU. (б.д.). Получено из

<https://www.youtube.com/watch?v=itrkg9o9pAc>

Дэвис, С. Р. (2015). *С++ для чайников.* Москва - Санкт-Петербург- Киев : Диалектика.

Заставка при открытии книги Excel RU. (б.д.). Получено из

www.planetaexcel.ru/techniques/3/168/

Лафоре, Р. (2013). *Объектно-ориентированное программирование в С++.* ПИТЕР.

Страуструп, Б. (2015). *Язык программирования С++.* БИНОМ.

Приложение

1) Apartments.h

```
#ifndef APARTMENTS_H_INCLUDED
#define APARTMENTS_H_INCLUDED

struct apartment{
    int Kod;
    char Raion[20];
    char Adres[20];
    int Kkomnat;
    int Etaj;
    int Price;
};

extern int nA;
extern apartment a[100];

struct rent{
    int Kod;
    char Fam[20];
    char Name[20];
    int Day;
    int Month;
    int Year;
    int SrokA;
};

extern int nR;
extern rent r[100];

void menu();
void readA();
void readR();
void showA();
void showR();
```

```

void addA();
void rewriteInfoA();
void addR();
void rewriteInfoR();
void deleteInfoA();
void deleteInfoR();
void RedactA();
void RedactR();
void Srok();
void Spros();
void Sapartments();
int mouse();

void Srprice();

#endif // APARTMENTS_H_INCLUDED

```

2) Apartments.cpp

```

#include <iostream>
#include <fstream>
#include <iomanip>
#include <cstring>
#include <windows.h>
#include "Apartments.h"

int nR,nA; // Создание глобальных переменных для количества квартир и аренд
apartment a[100]; // Создание глобальной структуры квартир
rent r[100]; // Создание глобальной структуры аренд
using namespace std;

void setcolor(int text, int backG=0){ // функция для изменения цвета
HANDLE color = GetStdHandle(STD_OUTPUT_HANDLE); // дескриптор устройства
стандартного вывода.
SetConsoleTextAttribute(color,(WORD)((backG << 4)| text)); // задаем цвет фона и цвет текста

```

```
}
```

```
int mouse(){ // Функция для мыши в главном меню
```

```
    HANDLE hin = GetStdHandle(STD_INPUT_HANDLE); // получаем дескриптор  
    INPUT_RECORD InputRecord; // используется для возвращения информации о входных  
сообщениях в консольном входном буфере
```

```
    DWORD Events; // unsigned long
```

```
    COORD coord; // для координат X, Y
```

```
/*Запрещение выделения консоли*/
```

```
    DWORD prev_mode;
```

```
    GetConsoleMode(hin, &prev_mode);
```

```
    SetConsoleMode(GetStdHandle(STD_INPUT_HANDLE), ENABLE_EXTENDED_FLAGS |  
(prev_mode & ~ENABLE_QUICK_EDIT_MODE));
```

```
/*Запрещение выделения консоли*/
```

```
    SetConsoleMode(hin, ENABLE_MOUSE_INPUT); // разрешаем обработку мыши
```

```
while (true)
```

```
{
```

```
    ReadConsoleInput(hin, &InputRecord, 1, &Events); // считывание
```

```
    if (InputRecord.Event.MouseEvent.dwButtonState ==  
FROM_LEFT_1ST_BUTTON_PRESSED) // если нажата левая кнопка мыши
```

```
    {
```

```
        coord.X = InputRecord.Event.MouseEvent.dwMousePosition.X; // считывание координат  
X
```

```
        coord.Y = InputRecord.Event.MouseEvent.dwMousePosition.Y; // считывание координат  
Y
```

```
        //cout << "left - X" << coord.X << ", Y = " << coord.Y << endl;
```

```
        if(coord.X>=45 and coord.X<=72 and coord.Y==2){ // задаём координаты кнопок для  
нажатия
```

```

prev_mode = 487; // присваиваем стандартный код консоли(для работы с
клавиатурой и т.д)
SetConsoleMode(hin,prev_mode); // возвращение стандартного режима консоли
return 1; // возвращаем нужное значение

}else if(coord.X>=46 and coord.X<=71 and coord.Y==3){ // задаём координаты кнопок
для нажатия
prev_mode = 487; // присваиваем стандартный код консоли(для работы с
клавиатурой и т.д)
SetConsoleMode(hin,prev_mode); // возвращение стандартного режима консоли
return 2; // возвращаем нужное значение

}else if(coord.X>=37 and coord.X<=80 and coord.Y==4){ // задаём координаты кнопок
для нажатия
prev_mode = 487; // присваиваем стандартный код консоли(для работы с
клавиатурой и т.д)
SetConsoleMode(hin,prev_mode); // возвращение стандартного режима консоли
return 3; // возвращаем нужное значение

}else if(coord.X>=38 and coord.X<=79 and coord.Y==5){ // задаём координаты кнопок
для нажатия
prev_mode = 487; // присваиваем стандартный код консоли(для работы с
клавиатурой и т.д)
SetConsoleMode(hin,prev_mode); // возвращение стандартного режима консоли
return 4; // возвращаем нужное значение

}else if(coord.X>=25 and coord.X<=93 and coord.Y==6){ // задаём координаты кнопок
для нажатия
prev_mode = 487; // присваиваем стандартный код консоли(для работы с
клавиатурой и т.д)
SetConsoleMode(hin,prev_mode); // возвращение стандартного режима консоли
return 5; // возвращаем нужное значение

```

```

} else if(coord.X>=25 and coord.X<=93 and coord.Y==7){ // задаём координаты кнопок
для нажатия
    prev_mode = 487; // присваиваем стандартный код консоли(для работы с
клавиатурой и т.д)
    SetConsoleMode(hin,prev_mode); // возвращение стандартного режима консоли
    return 6; // возвращаем нужное значение

} else if(coord.X>=22 and coord.X<=96 and coord.Y==8){ // задаём координаты кнопок
для нажатия
    prev_mode = 487; // присваиваем стандартный код консоли(для работы с
клавиатурой и т.д)
    SetConsoleMode(hin,prev_mode); // возвращение стандартного режима консоли
    return 7; // возвращаем нужное значение

} else if(coord.X>=22 and coord.X<=96 and coord.Y==9){ // задаём координаты кнопок
для нажатия
    prev_mode = 487; // присваиваем стандартный код консоли(для работы с
клавиатурой и т.д)
    SetConsoleMode(hin,prev_mode); // возвращение стандартного режима консоли
    return 8; // возвращаем нужное значение

} else if(coord.X>=17 and coord.X<=101 and coord.Y==10){ // задаём координаты кнопок
для нажатия
    prev_mode = 487; // присваиваем стандартный код консоли(для работы с
клавиатурой и т.д)
    SetConsoleMode(hin,prev_mode); // возвращение стандартного режима консоли
    return 9; // возвращаем нужное значение

} else if(coord.X>=24 and coord.X<=93 and coord.Y==11){ // задаём координаты кнопок
для нажатия
    prev_mode = 487; // присваиваем стандартный код консоли(для работы с
клавиатурой и т.д)
    SetConsoleMode(hin,prev_mode); // возвращение стандартного режима консоли
    return 10; // возвращаем нужное значение

```



```

    }else if(coord.X>=35 and coord.X<=81 and coord.Y==12){ // задаём координаты кнопок
для нажатия

        prev_mode = 487; // присваиваем стандартный код консоли(для работы с
клавиатурой и т.д)

        SetConsoleMode(hin,prev_mode); // возвращение стандартного режима консоли
        return 11; // возвращаем нужное значение

    }else if(coord.X>=28 and coord.X<=89 and coord.Y==13){ // задаём координаты кнопок
для нажатия

        prev_mode = 487; // присваиваем стандартный код консоли(для работы с
клавиатурой и т.д)

        SetConsoleMode(hin,prev_mode); // возвращение стандартного режима консоли
        return 12; // возвращаем нужное значение

    }else if(coord.X>=53 and coord.X<=63 and coord.Y==14){ // задаём координаты кнопок
для нажатия

        prev_mode = 487; // присваиваем стандартный код консоли(для работы с
клавиатурой и т.д)

        SetConsoleMode(hin,prev_mode); // возвращение стандартного режима консоли
        return 13; // возвращаем нужное значение

    }

}

    } else if (InputRecord.Event.MouseEvent.dwButtonState ==
RIGHTMOST_BUTTON_PRESSED) // если нажата правая кнопка мыши
return 0; // возвращаем нужное значение

}

return 0; // возвращаем нужное значение

}

```

```

int swredA(){ // Функция для мыши в редактировании квартир

    HANDLE hin = GetStdHandle(STD_INPUT_HANDLE); // получаем дескриптор
    INPUT_RECORD InputRecord; // используется для возвращения информации о входных
    сообщениях в консольном входном буфере
    DWORD Events; // unsigned long
    COORD coord; // для координат X, Y

    /*Запрещение выделения консоли*/
    DWORD prev_mode;
    GetConsoleMode(hin, &prev_mode);
    SetConsoleMode(GetStdHandle(STD_INPUT_HANDLE), ENABLE_EXTENDED_FLAGS |
    (prev_mode & ~ENABLE_QUICK_EDIT_MODE));
    /*Запрещение выделения консоли*/

    SetConsoleMode(hin, ENABLE_MOUSE_INPUT); // разрешаем обработку мыши

    while (true)
    {
        ReadConsoleInput(hin, &InputRecord, 1, &Events); // считывание

        if (InputRecord.Event.MouseEvent.dwButtonState ==
        FROM_LEFT_1ST_BUTTON_PRESSED){ // если нажата левая кнопка мыши
            coord.X = InputRecord.Event.MouseEvent.dwMousePosition.X; // считывание координат
            X
            coord.Y = InputRecord.Event.MouseEvent.dwMousePosition.Y; // считывание координат
            Y
            // cout << "left - X" << coord.X << ", Y = " << coord.Y << endl;

            if(coord.X>=7 and coord.X<=17 and coord.Y==3){ // задаём координаты кнопок для
            нажатия
                prev_mode = 487; // присваиваем стандартный код консоли(для работы с
                клавиатурой и т.д)
                SetConsoleMode(hin,prev_mode); // возвращение стандартного режима консоли

```

```

return 1; // возвращаем нужное значение

} else if(coord.X>=20 and coord.X<=30 and coord.Y==3){ // задаём координаты кнопок
для нажатия
    prev_mode = 487;
    SetConsoleMode(hin,prev_mode); // возвращение стандартного режима консоли
    return 2; // возвращаем нужное значение

} else if(coord.X>=33 and coord.X<=55 and coord.Y==3){ // задаём координаты кнопок
для нажатия
    prev_mode = 487; // присваиваем стандартный код консоли(для работы с
клавиатурой и т.д)
    SetConsoleMode(hin,prev_mode); // возвращение стандартного режима консоли
    return 3; // возвращаем нужное значение

} else if(coord.X>=58 and coord.X<=67 and coord.Y==3){ // задаём координаты кнопок
для нажатия
    prev_mode = 487; // присваиваем стандартный код консоли(для работы с
клавиатурой и т.д)
    SetConsoleMode(hin,prev_mode); // возвращение стандартного режима консоли
    return 4; // возвращаем нужное значение

} else if(coord.X>=70 and coord.X<=79 and coord.Y==3){ // задаём координаты кнопок
для нажатия
    prev_mode = 487; // присваиваем стандартный код консоли(для работы с
клавиатурой и т.д)
    SetConsoleMode(hin,prev_mode); // возвращение стандартного режима консоли
    return 5; // возвращаем нужное значение

} else if(coord.X>=79 and coord.X<=88 and coord.Y==3){ // задаём координаты кнопок
для нажатия
    prev_mode = 487; // присваиваем стандартный код консоли(для работы с
клавиатурой и т.д)

```

```

        SetConsoleMode(hin,prev_mode); // возвращение стандартного режима консоли
        return 6; // возвращаем нужное значение

    }else if(coord.X>=82 and coord.X<=90 and coord.Y==3){ // задаём координаты кнопок
для нажатия

        prev_mode = 487; // присваиваем стандартный код консоли(для работы с
клавиатурой и т.д)

        SetConsoleMode(hin,prev_mode); // возвращение стандартного режима консоли
        return 7; // возвращаем нужное значение

    }else if(coord.X>=93 and coord.X<=104 and coord.Y==3){ // задаём координаты кнопок
для нажатия

        prev_mode = 487; // присваиваем стандартный код консоли(для работы с
клавиатурой и т.д)

        SetConsoleMode(hin,prev_mode); // возвращение стандартного режима консоли
        return 0; // возвращаем нужное значение

    }

    }else if (InputRecord.Event.MouseEvent.dwButtonState ==
RIGHTMOST_BUTTON_PRESSED) // если нажата правая кнопка мыши
return 0; // возвращаем нужное значение

    }

return 0; // возвращаем нужное значение

}

int swredR(){ // Функция для мыши в редактировании аренд

HANDLE hin = GetStdHandle(STD_INPUT_HANDLE); // получаем дескриптор
INPUT_RECORD InputRecord; // используется для возвращения информации о входных
сообщениях в консольном входном буфере

```

```

DWORD Events; // unsigned long
COORD coord; // для координат X, Y

/*Запрещение выделения консоли*/
DWORD prev_mode;
GetConsoleMode(hin, &prev_mode);
SetConsoleMode(GetStdHandle(STD_INPUT_HANDLE), ENABLE_EXTENDED_FLAGS |
(prev_mode & ~ENABLE_QUICK_EDIT_MODE));
/*Запрещение выделения консоли*/

SetConsoleMode(hin, ENABLE_MOUSE_INPUT); // разрешаем обработку мыши

while (true)
{
    ReadConsoleInput(hin, &InputRecord, 1, &Events); // считывание

    if (InputRecord.Event.MouseEvent.dwButtonState ==
FROM_LEFT_1ST_BUTTON_PRESSED) // если нажата левая кнопка мыши
    {
        coord.X = InputRecord.Event.MouseEvent.dwMousePosition.X; // считывание координат
X
        coord.Y = InputRecord.Event.MouseEvent.dwMousePosition.Y; // считывание координат
Y
        //cout << "left - X" << coord.X << ", Y = " << coord.Y << endl;

        if(coord.X>5 and coord.X<=17 and coord.Y==3){ // задаём координаты кнопок для
нажатия
            prev_mode = 487; // присваиваем стандартный код консоли(для работы с
клавиатурой и т.д)
            SetConsoleMode(hin,prev_mode); // возвращение стандартного режима консоли
            return 1; // возвращаем нужное значение

        }else if(coord.X>=20 and coord.X<=28 and coord.Y==3){ // задаём координаты кнопок
для нажатия

```

```

prev_mode = 487; // присваиваем стандартный код консоли(для работы с
клавиатурой и т.д)
SetConsoleMode(hin,prev_mode); // возвращение стандартного режима консоли
return 2; // возвращаем нужное значение

}else if(coord.X>=31 and coord.X<=40 and coord.Y==3){ // задаём координаты кнопок
для нажатия
prev_mode = 487; // присваиваем стандартный код консоли(для работы с
клавиатурой и т.д)
SetConsoleMode(hin,prev_mode); // возвращение стандартного режима консоли
return 3; // возвращаем нужное значение

}else if(coord.X>=43 and coord.X<=53 and coord.Y==3){ // задаём координаты кнопок
для нажатия
prev_mode = 487; // присваиваем стандартный код консоли(для работы с
клавиатурой и т.д)
SetConsoleMode(hin,prev_mode); // возвращение стандартного режима консоли
return 4; // возвращаем нужное значение

}else if(coord.X>=56 and coord.X<=64 and coord.Y==3){ // задаём координаты кнопок
для нажатия
prev_mode = 487; // присваиваем стандартный код консоли(для работы с
клавиатурой и т.д)
SetConsoleMode(hin,prev_mode); // возвращение стандартного режима консоли
return 5; // возвращаем нужное значение

}else if(coord.X>=67 and coord.X<=83 and coord.Y==3){ // задаём координаты кнопок
для нажатия
prev_mode = 487; // присваиваем стандартный код консоли(для работы с
клавиатурой и т.д)
SetConsoleMode(hin,prev_mode); // возвращение стандартного режима консоли
return 6; // возвращаем нужное значение

```

```

    }else if(coord.X>=86 and coord.X<=94 and coord.Y==3){ // задаём координаты кнопок
для нажатия

        prev_mode = 487; // присваиваем стандартный код консоли(для работы с
клавиатурой и т.д)

        SetConsoleMode(hin,prev_mode); // возвращение стандартного режима консоли
        return 7; // возвращаем нужное значение

    }else if(coord.X>=97 and coord.X<=108 and coord.Y==3){ // задаём координаты кнопок
для нажатия

        prev_mode = 487; // присваиваем стандартный код консоли(для работы с
клавиатурой и т.д)

        SetConsoleMode(hin,prev_mode); // возвращение стандартного режима консоли
        return 0; // возвращаем нужное значение

    }

}

}else if (InputRecord.Event.MouseEvent.dwButtonState ==
RIGHTMOST_BUTTON_PRESSED)
return 0; // возвращаем нужное значение

}

return 0; // возвращаем нужное значение

}

int swopenFile(){ // Функция для мыши в задании, для открытия файла

HANDLE hin = GetStdHandle(STD_INPUT_HANDLE); // получаем дескриптор
INPUT_RECORD InputRecord; // используется для возвращения информации о входных
сообщениях в консольном входном буфере
DWORD Events; // unsigned long
COORD coord; // для координат X, Y

```

```

/*Запрещение выделения консоли*/
    DWORD prev_mode;
    GetConsoleMode(hin, &prev_mode);
    SetConsoleMode(GetStdHandle(STD_INPUT_HANDLE), ENABLE_EXTENDED_FLAGS |
    (prev_mode & ~ENABLE_QUICK_EDIT_MODE));
/*Запрещение выделения консоли*/

    SetConsoleMode(hin, ENABLE_MOUSE_INPUT); // разрешаем обработку мыши

    while (true)
    {
        ReadConsoleInput(hin, &InputRecord, 1, &Events); // считывание

        if (InputRecord.Event.MouseEvent.dwButtonState ==
        FROM_LEFT_1ST_BUTTON_PRESSED) // если нажата левая кнопка мыши
        {
            coord.X = InputRecord.Event.MouseEvent.dwMousePosition.X; // считывание координат
            X
            coord.Y = InputRecord.Event.MouseEvent.dwMousePosition.Y; // считывание координат
            Y
            //cout << "left - X" << coord.X << ", Y = " << coord.Y << endl;

            if(coord.X>60 and coord.X<=67 and coord.Y==2){ // задаём координаты кнопок для
            нажатия
                prev_mode = 487; // присваиваем стандартный код консоли(для работы с
                клавиатурой и т.д)
                SetConsoleMode(hin,prev_mode); // возвращение стандартного режима консоли
                return 1; // возвращаем нужное значение

            }else if(coord.X>=71 and coord.X<=79 and coord.Y==2){ // задаём координаты кнопок
            для нажатия
                prev_mode = 487; // присваиваем стандартный код консоли(для работы с
                клавиатурой и т.д)
                SetConsoleMode(hin,prev_mode); // возвращение стандартного режима консоли

```



```

        return 0; // возвращаем нужное значение

    }

    }else if (InputRecord.Event.MouseEvent.dwButtonState ==
RIGHTMOST_BUTTON_PRESSED)
return 0; // возвращаем нужное значение

    }

return 0; // возвращаем нужное значение

}

void menu(){ // Функция для вывода меню
    setlocale(LC_ALL, "Russian"); // выбор русского языка в консоли
    setcolor(14,0); // Смена цвета на жёлтый
    cout<<"\n\n";
    cout<<internal<<setw(73)<<"<< Информация о квартирах >>"<<endl;
    cout<<setw(72)<<"<< Информация об аренде >>"<<endl;
    cout<<setw(81)<<"<< Добавление информации о новой квартире >>"<<endl;
    cout<<setw(80)<<"<< Добавление информации о новой аренде >>"<<endl;
    cout<<setw(94)<<"<< Удаление информации о квартире, код которой введен с клавиатуры
>>"<<endl;
    cout<<setw(94)<<"<< Удаление информации об аренде, код которой введен с клавиатуры
>>"<<endl;
    cout<<setw(97)<<"<< Редактирование информации о квартире, код которой введен с
клавиатуры >>"<<endl;
    cout<<setw(97)<<"<< Редактирование информации об аренде, код которой введен с
клавиатуры >>"<<endl;
    cout<<setw(102)<<"<< Создание текстового файла, содержащего информацию об
арендах,срок которых истек >>"<<endl;
    cout<<setw(94)<<"<< Список свободных квартир в порядке возрастания количества
комнат >>"<<endl;
    cout<<setw(82)<<"<< Квартира, пользующаяся наибольшим спросом >>"<<endl;

```

```

    cout<<setw(90)<<"<< Средняя цена всех квартир с указанным количеством комнат
>>"<<endl;

    setcolor(12,0); // Смена цвета на красный
    cout<<setw(64)<<"<< Выход >>"<<endl;
    setcolor(14,0); // Смена цвета на жёлтый
}

void readA(){ // Функция для считывания информации о квартирах с файла
    ifstream f("Apartment.txt"); // Открытие файла для чтения
    if(f){ // Проверка на существование файла
        if(f.peek()!=EOF){ // Проверка на то, не пустой ли файл
            int i=0; // Счётчик для количества квартир
            while(!f.eof()){ // Пока не конец файла
                f>>a[i].Kod>>a[i].Raion >> a[i].Adres >> a[i].Kkomnat>>a[i].Etaj >> a[i].Price; //
Считывание данных с файла
                i++; // Увеличение счётчика
            }
            nA=i-1; // Присваивание переменной количества квартир(-1 для того, чтобы не учитывать
последний пробел в файле)
            f.close(); // Закрытие файла
        }else{ // в другом случае
            setcolor(12,0); // Смена цвета на красный
            cout <<"Файл с информацией о квартирах пустой."<< endl;
        }
        }else{ // в другом случае
            setcolor(12,0); // Смена цвета на красный
            cout <<"Файл с информацией о квартирах не найден."<< endl;
        }
        setcolor(14,0); // Смена цвета на жёлтый
    }

}

void readR(){ // Функция для считывания информации об арендах с файла
    ifstream g("Rent.txt"); // Открытие файла для чтения
    int i=0; // Счётчик для количества аренд

```

```

if(g){ // Проверка на существование файла
if(g.peek()!=EOF){ // Проверка на то, не пустой ли файл
while(!g.eof()){ // Пока не конец файла
    g>>r[i].Kod>>r[i].Fam >> r[i].Name >> r[i].Day>>r[i].Month >> r[i].Year >> r[i].SrokA; //
Считывание данных с файла
    i++; // Увеличение счётчика
}
    nR=i-1; // Присваивание переменной количества аренд(-1 для того, чтобы не учитывать
последний пробел в файле)
    g.close(); // Закрытие файла
}else{ // в другом случае
    setcolor(12,0); // Смена цвета на красный
    cout <<"Файл с информацией об аренде пустой."<< endl;
}
}else{ // в другом случае
    setcolor(12,0); // Смена цвета на красный
    cout <<"Файл с информацией об аренде не найден."<< endl;
}
setcolor(14,0); // Смена цвета на жёлтый

}

void showA(){ // Функция для вывода информации о квартирах
setcolor(0,15); // смена цвета заднего фона на белый

    cout<<endl<<"    Код " <<"    Район    " <<"    Адрес    " <<"    Кол-во " <<"
Этаж " <<"    Цена    " <<endl;
    cout<<"    Квартиры                комнат                " <<endl; // Создание
шапки таблицы

    for(int i=0;i<nA;i++){
        if(i%2==0) setcolor(0,8); // Проверка для смены цвета через строку. Смена цвета текста на
тёмно-серый
        else setcolor(0,7); // Смена цвета на светло-серый

```

```

cout << left << " " << setw(11) << a[i].Kod << setw(19) << a[i].Raion << setw(28) << a[i].Adres
<< setw(8) << a[i].Kkomnat << setw(8) << a[i].Etaj << a[i].Price << endl; // Вывод информации
}

setcolor(14,0); // Смена цвета на жёлтый
}

void showR(){ // Функция для вывода информации об арендах
    setcolor(0,19); // Смена заднего фона на бирюзовый
    cout << endl << " Код квартиры " << " Фамилия " << " Имя " << " Дата аренды"
    << " Срок аренды " << endl; // Создание шапки таблицы

    for(int i=0; i<nR; i++){
        if(i%2==0) setcolor(0,7); // Проверка для смены цвета через строку. Смена цвета на
        светло-серый
        else setcolor(7,1); // Смена цвета на тёмно-синий
        cout << left << " " << setw(12) << r[i].Kod << setw(15) << r[i].Fam << setw(16) << r[i].Name
        << setw(2) << r[i].Day << "/" << setw(2) << r[i].Month << "/" << setw(14) << r[i].Year <<
        setw(5) << r[i].SrokA << endl; // Вывод информации
    }

    setcolor(14,0); // Смена цвета на жёлтый
}

void addA(){ // Функция для добавления информации о квартирах
    apartment ap; // создание структуры квартир
    int pos; // создание переменной позиции
    bool flag; // создание логической переменной
    cout << "Введите код квартиры:";

    do{ // Защита от неправильного ввода
        cin >> ap.Kod; // ввод кода квартиры
        flag = 1; // приравнивание логической переменной к 1
        for(int i=0; i<nA; i++){
            if(ap.Kod == a[i].Kod) flag = 0; // если введённый код квартиры равен одному из кодов
            квартиры файла, то переменная равна 0

```

```

if(flag==0){ // если введенный код квартиры равен одному из кодов квартиры файла
    setcolor(12,0); // смена цвета текста на красный
    cout<<"Ошибка! Квартира с таким номером уже существует! Попробуйте ещё раз!
"<<endl;
    setcolor(14,0); // смена цвета текста на жёлтый
    cout<<"Введите код квартиры:";

} else if(ap.Kod<1) {
    setcolor(12,0); // смена цвета текста на красный
    cout<<"Ошибка! Попробуйте ещё раз! "<<endl;
    setcolor(14,0); // смена цвета текста на жёлтый
    cout<<"Введите код квартиры:";
}
}while(ap.Kod<1 or flag==0); // Защита от неправильного ввода

cout<<"Введите район:";
cin>>ap.Raion; // ввод района квартиры

cout<<"Введите адрес:";
cin>>ap.Adres; // ввод адреса квартиры

cout<<"Введите количество комнат:";
do{ // Защита от неправильного ввода
    cin>>ap.Kkomnat;
    if(ap.Kkomnat<1){
        setcolor(12,0); // смена цвета текста на красный
        cout<<"Ошибка! Попробуйте ещё раз! "<<endl;
        setcolor(14,0); // смена цвета текста на жёлтый
        cout<<"Введите количество комнат:";
    }
}while(ap.Kkomnat<1); // Защита от неправильного ввода

```

```

cout<<"Введите этаж:";

    do{ // Защита от неправильного ввода
        cin>>ap.Etaj;
        if(ap.Etaj<1){
            setcolor(12,0); // смена цвета текста на красный
            cout<<"Ошибка! Попробуйте ещё раз! "<<endl;
            setcolor(14,0); // смена цвета текста на жёлтый
        }
        cout<<"Введите этаж:";
    }
}while(ap.Etaj<1); // Защита от неправильного ввода


cout<<"Введите цену:";

    do{ // Защита от неправильного ввода
        cin>>ap.Price; // ввод цены квартиры
        if(ap.Price<1){ // если цена квартиры меньше 1
            setcolor(12,0); // смена цвета текста на красный
            cout<<"Ошибка! Попробуйте ещё раз! "<<endl;
            setcolor(14,0); // смена цвета текста на жёлтый
        }
        cout<<"Введите этаж:";
    }
}while(ap.Price<1); // Защита от неправильного ввода


cout<<"Введите позицию(при большой позиции информация добавится в конец файла):";

    do{ // Защита от неправильного ввода
        cin>>pos; // ввод желаемой позиции
        if(pos<1 ){ // если желаемая позиция меньше 1
            setcolor(12,0); // смена цвета текста на красный
            cout<<"Ошибка! Попробуйте ещё раз! "<<endl;
            setcolor(14,0); // смена цвета текста на жёлтый
        }
        cout<<"Введите позицию(при большой позиции информация добавится в конец файла): ";
    }
}while(pos<1); // Защита от неправильного ввода


setcolor(10,0); // смена цвета текста на зелёный

```

```

cout << "\nИнформация о новой квартире добавлена.\n";
setcolor(14,0); // смена цвета текста на жёлтый

pos--; // Уменьшение переменной на 1, так как отсчёт массива начинается с нуля
if(pos>nA) pos=nA; // если желаемая позиция больше максимальной в файле, то позиция
становится последней
for(int i=nA; i>pos; i--)
    a[i]=a[i-1]; // смещение массива структур вправо
a[pos]=ap; // замена структур
nA++; // увеличение количества квартир
rewriteInfoA(); // перезапись файла квартир
}

void rewriteInfoA(){ // Функция для перезаписи файла квартир
    ofstream f("Apartment.txt"); // открытие файла квартир для записи
    for(int i=0; i<nA; i++)
        f << a[i].Kod << " " << a[i].Raion << " " << a[i].Adres << " " << a[i].Kkomnat << " " << a[i].Etaj <<
        " " << a[i].Price << endl; // перезапись всех квартир в файл
    f.close(); // закрытие файла
}

void addR(){ // Функция для добавления информации об арендах
    rent rt; // создание структуры аренд
    int pos; // создание переменной позиции

    cout << "Введите код квартиры:";
    do{ // Защита от неправильного ввода
        cin >> rt.Kod; // ввод кода квартиры
        if(rt.Kod < 1){ // если код квартиры меньше 1
            setcolor(12,0); // смена цвета текста на красный
            cout << "Ошибка! Попробуйте ещё раз! " << endl;
            setcolor(14,0); // смена цвета текста на жёлтый
            cout << "Введите код квартиры:";
        }
    } while(rt.Kod < 1); // Защита от неправильного ввода
}

```

```

cout<<"Введите фамилию арендатора:";
cin>>rt.Fam; // ввод фамилии арендатора

cout<<"Введите имя арендатора:";
cin>>rt.Name; // ввод имени арендатора

cout<<"Введите день аренды:";

    do{ // Защита от неправильного ввода
cin>>rt.Day; // ввод дня аренды
if(rt.Day<1 or rt.Day>31){ // если день аренды меньше 1 или больше 31
    setcolor(12,0); // смена цвета текста на красный
cout<<"Ошибка! Попробуйте ещё раз! "<<endl;
    setcolor(14,0); // смена цвета текста на жёлтый
cout<<"Введите день аренды:";
}
}while(rt.Day<1 or rt.Day>31); // Защита от неправильного ввода

cout<<"Введите месяц аренды:";
    do{ // Защита от неправильного ввода
cin>>rt.Month; // ввод месяца аренды
if(rt.Month<1 or rt.Month>12){ // если месяц аренды меньше 1 или больше 12
    setcolor(12,0); // смена цвета текста на красный
cout<<"Ошибка! Попробуйте ещё раз! "<<endl;
    setcolor(14,0); // смена цвета текста на жёлтый
cout<<"Введите месяц аренды:";
}
}while(rt.Month<1 or rt.Month>12); // Защита от неправильного ввода

cout<<"Введите год аренды:";

```



```

    do{ // Защита от неправильного ввода
cin>>rt.Year; // ввод года аренды
if(rt.Year<1){ // если год аренды меньше 1
    setcolor(12,0); // смена цвета текста на красный
    cout<<"Ошибка! Попробуйте ещё раз! "<<endl;
    setcolor(14,0); // смена цвета текста на жёлтый
    cout<<"Введите год аренды:";
}
}while(rt.Year<1 ); // Защита от неправильного ввода

```

```

    cout<<"Введите срок аренды:";
    do{ // Защита от неправильного ввода
cin>>rt.SrokA; // ввод срока аренды
if(rt.SrokA<1){ // если срок аренды меньше 1
    setcolor(12,0); // смена цвета текста на красный
    cout<<"Ошибка! Попробуйте ещё раз! "<<endl;
    setcolor(14,0); // смена цвета текста на жёлтый
    cout<<"Введите срок аренды:";
}
}while(rt.SrokA<1 ); // Защита от неправильного ввода

```

```

    cout<<"Введите позицию(при большой позиции информация добавится в конец файла): ";
    do{ // Защита от неправильного ввода
cin>>pos; // ввод желаемой позиции
if(pos<1){ // если желаемая позиция меньше 1
    setcolor(12,0); // смена цвета текста на красный
    cout<<"Ошибка! Попробуйте ещё раз! "<<endl;
    setcolor(14,0); // смена цвета текста на жёлтый
    cout<<"Введите позицию(при большой позиции информация добавится в конец файла): ";
}
}while(pos<1); // Защита от неправильного ввода

```

```

setcolor(10,0); // смена цвета текста на зелёный
cout << "\nИнформация об новой аренде добавлена.\n";
setcolor(14,0); // смена цвета текста на жёлтый
    pos--; // Уменьшение переменной на 1, так как отсчёт массива начинается с нуля
    if(pos>nR) pos=nR; // если желаемая позиция больше максимальной в файле, то позиция
становится последней
    for(int i=nR; i>pos; i--)
        r[i]=r[i-1]; // смещение массива структур вправо
    r[pos]=rt; // замена структур
    nR++; // увеличение количества аренд
    rewriteInfoR(); // перезапись файла аренд
}

void rewriteInfoR(){ // Функция для перезаписи файла аренд
    ofstream f("Rent.txt"); // открытие файла аренд для записи
    for(int i=0; i<nR; i++)
        f << r[i].Kod << " " << r[i].Fam << " " << r[i].Name << " " << r[i].Day << " " << r[i].Month << "
" << r[i].Year << " " << r[i].SrokA << endl; // перезапись всех аренд в файл
    f.close(); // закрытие файла
}

void deleteInfoA(){ // функция для удаления информации о квартире
    int k,p; // создание переменной для кода квартиры и счётчика
    cout<< "\nВведите код квартиры для удаления:";

    do{ // защита от неправильного ввода
        cin>> k; // ввод кода квартиры
        if(k<1){ // если код квартиры меньше 1
            setcolor(12,0); // смена цвета текста на красный
            cout<< "Ошибка! Попробуйте ещё раз! " << endl;
            setcolor(14,0); // смена цвета текста на жёлтый
            cout<< "\nВведите код квартиры для удаления:";
        }
    }
}

```

```

} while(k<1); // защита от неправильного ввода

for(int i=0; i<nA; i++)
if(k== a[i].Kod) { // если введённый код квартиры есть в массиве структур квартир
    for(int j=i; j<nA; j++) a[j]=a[j+1]; // сдвиг массива квартир влево
    nA--; // уменьшение количества квартир
    p++; // увеличение счётчика для проверки
}
if(p==0){ // если переменная не увеличилась
    setcolor(12,0); // смена цвета текста на красный
    cout << "\nКод квартиры не найден.\n";
    setcolor(14,0); // смена цвета текста на жёлтый
}

else { // в другом случае
    setcolor(10,0); // смена цвета текста на зелёный
    cout << "Информация о квартире под номером " << k << " удалена.\n\n";
    setcolor(14,0); // смена цвета текста на жёлтый
    for(int i=0; i<nR; i++)
        if(k==r[i].Kod){ // если введённый код квартиры есть в массиве структур аренд
            for(int j=i; j<nA; j++)
                r[j]=r[j+1]; // сдвиг массива аренд влево
            nR--; // уменьшение количества аренд
        }

    rewriteInfoA(); // перезапись файла квартир
    rewriteInfoR(); // перезапись файла аренд
}

}

void deleteInfoR(){ // функция для удаления информации об аренде
    int pn; // создание переменной для порядкового номера аренды
    cout<<"\nВведите порядковый номер аренды для удаления:";

```

```

        do{ // защита от неправильного ввода
cin>> pn;
if(pn<1){ // если порядковый номер аренды меньше 1
    setcolor(12,0); // смена цвета текста на красный
    cout<<"Ошибка! Попробуйте ещё раз! "<<endl;
    setcolor(14,0); // смена цвета текста на жёлтый
    cout<<"\nВведите порядковый номер аренды для удаления:";
}
}while(pn<1); // защита от неправильного ввода
pn--; // Уменьшение переменной на 1, так как отсчёт массива начинается с нуля

if(pn>nR) { // если порядковый номер аренды больше величины массива структур аренд
    setcolor(12,0); // смена цвета на красный
    cout << "\nПорядковый номер аренды не найден.\n";
    setcolor(14,0); // смена цвета на жёлтый
} else{ // в другом случае
    for(int i=pn; i<nR-1; i++)
        r[i]=r[i+1]; // сдвиг массива аренд влево
    nR--; // уменьшение количества аренд

    setcolor(10,0); // смена цвета текста на зелёный
    cout << "Информация об аренде под порядковым номером " << pn+1 << " удалена.\n";
    setcolor(14,0); // смена цвета текста на жёлтый
    rewriteInfoR(); // перезапись файла аренд
}

}

void RedactA(){ // функция для редактирования информации о квартире
    int k,p; // создание переменных для кода квартиры и позиции
    cout <<"Введите код квартиры для редактирования: ";

    do{ // защита от неправильного ввода

```

```

cin>> k; // ввод кода квартиры

if(k<1){ // если код квартиры меньше 1
    setcolor(12,0); // смена цвета текста на красный
    cout<<"Ошибка! Попробуйте ещё раз! "<<endl;
    setcolor(14,0); // смена цвета текста на жёлтый
    cout <<"Введите код квартиры для редактирования: ";
}
}while(k<1); // защита от неправильного ввода

system("CLS"); // очистка консоли
for(int i=0;i<nA;i++)
    if(k == a[i].Kod) p=i; // если введённый код квартиры равен коду квартиры из файла, то
// переменная p равен порядковому номеру
if(p!=0){ // если переменная p изменилась
    setcolor(15,0); // смена цвета текста на белый
    cout <<"\nЧто вы хотите изменить?\n\n";
    setcolor(14,0); // смена цвета текста на жёлтый

    cout <<"    << Район >>" << " << Адрес >>" << " << Количество комнат >>"
    << " << Этаж >>" << " << Цена >>"<< " << Всё >>";
    setcolor(12,0); // смена цвета текста на красный
    cout<< " << Отмена >>"<<endl;
    setcolor(14,0); // смена цвета текста на жёлтый

    int v=swredA(); // считывание нажатия мыши по кнопкам в консоли

    switch(v){

        case 1:{
            cout<<"Введите район:";
            cin>> a[p].Raion; // ввод района

```

```

rewriteInfoA(); // перезапись файла квартир
    break;
}

case 2:{
    cout<<"Введите адрес:";
    cin>> a[p].Adres; // ввод адреса

    rewriteInfoA(); // перезапись файла квартир
    break;
}
case 3:{

    cout<<"Введите количество комнат:";
        do{ // защита от неправильного ввода
            cin>> a[p].Kkomnat; // ввод количества комнат
            if(a[p].Kkomnat<1){ // если количество комнат меньше 1
                setcolor(12,0); // смена цвета текста на красный
                cout<<"Ошибка! Попробуйте ещё раз! "<<endl;
                setcolor(14,0); // смена цвета текста на жёлтый
                cout<<"Введите количество комнат:";
            }
        }while(a[p].Kkomnat<1); // защита от неправильного ввода

    rewriteInfoA(); // перезапись файла квартир
    break;
}

case 4:{
    cout<<"Введите этаж:";
    do{ // защита от неправильного ввода
        cin>> a[p].Etaj; // ввод этажа
        if(a[p].Etaj<1){
            setcolor(12,0); // смена цвета текста на красный

```

```

    cout<<"Ошибка! Попробуйте ещё раз! "<<endl;
    setcolor(14,0); // смена цвета текста на жёлтый
    cout<<"Введите этаж:";
}
} while(a[p].Etaj<1); // защита от неправильного ввода

rewriteInfoA(); // перезапись файла квартир
    break;
}

case 5:{
    cout<<"Введите цену:";
    do{ // защита от неправильного ввода
    cin>> a[p].Price; // ввод цены
    if(a[p].Price<1){ // если цена меньше 1
        setcolor(12,0); // смена цвета текста на красный
        cout<<"Ошибка! Попробуйте ещё раз! "<<endl;
        setcolor(14,0); // смена цвета текста на жёлтый
        cout<<"Введите цену:";
    }
    } while(a[p].Price<1); // защита от неправильного ввода

    rewriteInfoA(); // перезапись файла квартир
    break;
}

case 6:{

cout<<"Введите район:";
    cin>> a[p].Raion; // ввод района

    cout<<"Введите адрес:";
    cin>> a[p].Adres; // ввод адреса

```

```

    cout<<"Введите количество комнат:";

    do{ // защита от неправильного ввода
cin>> a[p].Kkomnat; // ввод количества комнат
    if(a[p].Kkomnat<1){ // если количество комнат меньше 1
        setcolor(12,0); // смена цвета текста на красный
        cout<<"Ошибка! Попробуйте ещё раз! "<<endl;
        setcolor(14,0); // смена цвета текста на жёлтый
        cout<<"Введите количество комнат:";
    }
} while(a[p].Kkomnat<1); // защита от неправильного ввода

```

```

    cout<<"Введите этаж:";

    do{ // защита от неправильного ввода
cin>> a[p].Etaj; // ввод этажа
    if(a[p].Etaj<1){
        setcolor(12,0); // смена цвета текста на красный
        cout<<"Ошибка! Попробуйте ещё раз! "<<endl;
        setcolor(14,0); // смена цвета текста на жёлтый
        cout<<"Введите этаж:";
    }
} while(a[p].Etaj<1); // защита от неправильного ввода

```

```

    cout<<"Введите цену:";

    do{ // защита от неправильного ввода
cin>> a[p].Price; // ввод цены
    if(a[p].Price<1){
        setcolor(12,0); // смена цвета текста на красный
        cout<<"Ошибка! Попробуйте ещё раз! "<<endl;
        setcolor(14,0); // смена цвета текста на жёлтый
        cout<<"Введите цену:";
    }
} while(a[p].Price<1); // защита от неправильного ввода

```



```

rewriteInfoA(); // перезапись файла квартир
break;
}
    case 0:{
        setcolor(12,0); // смена цвета текста на красный
        cout << "\nОтмена.\n";
        setcolor(14,0); // смена цвета текста на жёлтый
        break;
    }
}
if(v!=0){ // если счётчик изменился
    setcolor(10,0); // смена цвета текста на зелёный
    cout << "\nДанные изменены\n";
    setcolor(14,0); // смена цвета текста на жёлтый
}
} else{ // в другом случае
    setcolor(12,0); // смена цвета текста на красный
    cout<<"Код квартиры не найден.\n";
    setcolor(14,0); // смена цвета текста на жёлтый
}
}

void RedactR(){ // функция для редактирования информации об аренде
    int k; // создание переменной для порядкового номера аренды
    cout << "Введите порядковый номер аренды для редактирования: ";
    do{ // защита от неправильного ввода
        cin>> k; // ввод порядкового номера аренды
        if(k<1 or k>nR){
            setcolor(12,0); // смена цвета текста на красный
            cout<<"Ошибка! Попробуйте ещё раз! "<<endl;
            setcolor(14,0); // смена цвета текста на жёлтый
        }
        cout << "Введите порядковый номер аренды для редактирования: ";
    }
} while(k<1 or k>nR); // защита от неправильного ввода

```

```

k--; // уменьшение переменной на 1, так как отсчёт массива начинается с нуля
system("CLS"); // очистка консоли
setcolor(15,0); // смена цвета текста на белый
cout << "\nЧто вы хотите изменить?\n\n";
setcolor(14,0); // смена цвета текста на жёлтый

cout << "    << Фамилия >>" << " << Имя >>" << " << День >>" << " << Месяц >>"
<< " << Год >>" << " << Срок аренды >>" << " << Всё >>";
setcolor(12,0); // смена цвета текста на красный
cout << " << Отмена >>" << endl;
setcolor(14,0); // смена цвета текста на жёлтый

```

```

int v=swredR(); // считывание нажатия мыши по кнопкам в консоли

```

```

switch(v){
    case 1:{
        cout<<"Введите фамилию арендатора:";
        cin>>r[k].Fam; // ввод фамилии арендатора
        rewriteInfoR(); // перезапись файла аренд
        break;
    }
    case 2:{
        cout<<"Введите имя арендатора:";
        cin>>r[k].Name; // ввод имени арендатора
        rewriteInfoR(); // перезапись файла аренд
        break;
    }
    case 3:{
        cout<<"Введите день аренды:";
        do{ // защита от неправильного ввода
            cin>>r[k].Day; // ввод дня аренды
            if(r[k].Day<1 or r[k].Day>31){ // если день аренды меньше 1 или больше 31

```

```

        setcolor(12,0); // смена цвета текста на красный
        cout<<"Ошибка! Попробуйте ещё раз! "<<endl;
        setcolor(14,0); // смена цвета текста на жёлтый
        cout<<"Введите день аренды:";
    }
} while(r[k].Day<1 or r[k].Day>31); // защита от неправильного ввода

    rewriteInfoR(); // перезапись файла аренд
    break;
}

case 4:{
    cout<<"Введите месяц аренды:";
    do{ // защита от неправильного ввода
        cin>>r[k].Month; // ввод месяца аренды
        if(r[k].Month<1 or r[k].Month>12){ // если месяц аренды меньше 1 или больше 12
            setcolor(12,0); // смена цвета текста на красный
            cout<<"Ошибка! Попробуйте ещё раз! "<<endl;
            setcolor(14,0); // смена цвета текста на жёлтый
            cout<<"Введите месяц аренды:";
        }
    } while(r[k].Month<1 or r[k].Month>12); // защита от неправильного ввода

    rewriteInfoR(); // перезапись файла аренд
    break;
}

case 5:{
    cout<<"Введите год аренды:";
    do{ // защита от неправильного ввода
        cin>>r[k].Year; // ввод года аренды
        if(r[k].Year<1){ // если год аренды меньше 1
            setcolor(12,0); // смена цвета текста на красный
            cout<<"Ошибка! Попробуйте ещё раз! "<<endl;
            setcolor(14,0); // смена цвета текста на жёлтый
            cout<<"Введите год аренды:";
        }
    }
}

```

```
} while(r[k].Year<1); // защита от неправильного ввода
```

```
rewriteInfoR(); // перезапись файла аренд
```

```
break;
```

```
}
```

```
case 6:{
```

```
cout<<"Введите срок аренды:";
```

```
do{ // защита от неправильного ввода
```

```
cin>>r[k].SrokA; // ввод срока аренды
```

```
if(r[k].SrokA<1){ // если срок аренды меньше 1
```

```
setcolor(12,0); // смена цвета текста на красный
```

```
cout<<"Ошибка! Попробуйте ещё раз! "<<endl;
```

```
setcolor(14,0); // смена цвета текста на жёлтый
```

```
cout<<"Введите срок аренды:";
```

```
}
```

```
} while(r[k].SrokA<1); // защита от неправильного ввода
```

```
rewriteInfoR(); // перезапись файла аренд
```

```
break;
```

```
}
```

```
case 7:{
```

```
cout<<"Введите фамилию арендатора:";
```

```
cin>>r[k].Fam; // ввод фамилии арендатора
```

```
cout<<"Введите имя арендатора:";
```

```
cin>>r[k].Name; // ввод имени арендатора
```

```
cout<<"Введите день аренды:";
```

```
do{ // защита от неправильного ввода
```

```
cin>>r[k].Day; // ввод дня аренды
```

```
if(r[k].Day<1 or r[k].Day>31){ // если день аренды меньше 1 или больше 31
```

```
setcolor(12,0); // смена цвета текста на красный
```

```
cout<<"Ошибка! Попробуйте ещё раз! "<<endl;
```

```
setcolor(14,0); // смена цвета текста на жёлтый
```

```
cout<<"Введите день аренды:";
```

```

}
}while(r[k].Day<1 or r[k].Day>31); // защита от неправильного ввода

    cout<<"Введите месяц аренды:";
    do{ // защита от неправильного ввода
        cin>>r[k].Month; // ввод месяца аренды
        if(r[k].Month<1 or r[k].Month>12){ // если месяц аренды меньше 1 или больше 12
            setcolor(12,0); // смена цвета текста на красный
            cout<<"Ошибка! Попробуйте ещё раз! "<<endl;
            setcolor(14,0); // смена цвета текста на жёлтый
            cout<<"Введите месяц аренды:";
        }
    }while(r[k].Month<1 or r[k].Month>12); // защита от неправильного ввода

    cout<<"Введите год аренды:";
    do{ // защита от неправильного ввода
        cin>>r[k].Year; // ввод года аренды
        if(r[k].Year<1){ // если год аренды меньше 1
            setcolor(12,0); // смена цвета текста на красный
            cout<<"Ошибка! Попробуйте ещё раз! "<<endl;
            setcolor(14,0); // смена цвета текста на жёлтый
            cout<<"Введите год аренды:";
        }
    }while(r[k].Year<1); // защита от неправильного ввода

    cout<<"Введите срок аренды:";
    do{ // защита от неправильного ввода
        cin>>r[k].SrokA; // ввод срока аренды
        if(r[k].SrokA<1){ // если срок аренды меньше 1
            setcolor(12,0); // смена цвета текста на красный
            cout<<"Ошибка! Попробуйте ещё раз! "<<endl;
            setcolor(14,0); // смена цвета текста на жёлтый
            cout<<"Введите срок аренды:";
        }
    }while(r[k].SrokA<1); // защита от неправильного ввода

```

```

rewriteInfoR(); // перезапись файла аренд
    break;
}

case 0:{
setcolor(12,0); // смена цвета текста на красный
    cout << "\nОтмена.\n";
    setcolor(14,0); // смена цвета текста на жёлтый
}
}
if(v!=0){ // если переменная v изменилась
    setcolor(10,0); // смена цвета текста на зелёный
    cout << "\nДанные изменены\n";
    setcolor(14,0); // смена цвета текста на жёлтый
}
}

void Srok(){ // функция для поиска истёкших аренд
int date[3]; // создание массива для желаемой даты
cout << "Введите день(1-31): ";
do{ // защита от неправильного ввода
cin >> date[0]; // ввод дня

if(date[0]<1 or date[0]>31){ // если день меньше 1 или больше 31
    setcolor(12,0); // смена цвета текста на красный
    cout<<"Ошибка! Попробуйте ещё раз! "<<endl;
    setcolor(14,0); // смена цвета текста на жёлтый
    cout << "Введите день(1-31): ";
}
}while(date[0]<1 or date[0]>31); // защита от неправильного ввода

cout << "Введите месяц(1-12): ";
do{ // защита от неправильного ввода
cin >> date[1]; // ввод месяца

```

```

if(date[1]<1 or date[1]>12){ // если месяц меньше 1 или больше 12
    setcolor(12,0); // смена цвета текста на красный
    cout<<"Ошибка! Попробуйте ещё раз! "<<endl;
    setcolor(14,0); // смена цвета текста на жёлтый
    cout <<"Введите месяц(1-31): ";
}
}while(date[1]<1 or date[1]>12); // защита от неправильного ввода

cout <<"Введите год(>1): ";
do{ // защита от неправильного ввода

    cin >> date[2]; // ввод года

    if(date[2]<1){ // если год меньше 1
        setcolor(12,0); // смена цвета текста на красный
        cout<<"Ошибка! Попробуйте ещё раз! "<<endl;
        setcolor(14,0); // смена цвета текста на жёлтый
        cout <<"Введите год(>1): ";
    }
}while(date[2]<1); // защита от неправильного ввода

ofstream f("Expire.txt"); // открытие файла для записи

for(int i=0;i<nR;i++){
    if(date[2] > r[i].Year + r[i].SrokA) // если введённый год больше года+срока аренды
    f <<r[i].Kod <<" "<< r[i].Fam <<" "<<r[i].Name <<" "<<r[i].Day<<" "<<r[i].Month<<"
    "<<r[i].Year<<" "<<r[i].SrokA<<endl; // запись информации об истекшей аренде в файл
    else if(date[2] == r[i].Year + r[i].SrokA and date[1] > r[i].Month) // если введённый год равен
    году+сроку аренды и введённый месяц больше месяца аренды
    f <<r[i].Kod <<" "<< r[i].Fam <<" "<<r[i].Name <<" "<<r[i].Day<<" "<<r[i].Month<<"
    "<<r[i].Year<<" "<<r[i].SrokA<<endl; // запись информации об истекшей аренде в файл
}

```

```

else if(date[2] == r[i].Year + r[i].SrokA and date[1] == r[i].Month and date[0] > r[i].Day )// если
введённый год равен году+сроку аренды и введённый месяц равен месяцу аренды и
введённый день больше дня аренды
f << r[i].Kod << " " << r[i].Fam << " " << r[i].Name << " " << r[i].Day << " " << r[i].Month << "
" << r[i].Year << " " << r[i].SrokA << endl; // запись информации об истекшей аренде в файл
}
system("CLS"); // очистка консоли
setcolor(10,0); // смена цвета текста на зелёный
cout << "Файл создан." << endl;
setcolor(14,0); // смена цвета текста на жёлтый
cout << internal << setw(175) << "Желаете открыть файл?";
setcolor(10,0); // смена цвета текста на зелёный
cout << "    << Да >>";
setcolor(12,0); // смена цвета текста на красный
cout << "    << Нет >>" << endl;
setcolor(14,0); // смена цвета текста на жёлтый

int ch=swopenFile(); // считывание нажатия мыши по кнопкам в консоли

if(ch==1){ // если была нажата кнопка << Да >>
    ShellExecute(NULL, "open", "Expire.txt", NULL, NULL, SW_SHOWNORMAL); // открытие
созданного файла
setcolor(10,0); // смена цвета текста на зелёный
cout << "Открытие файла..." << endl;
setcolor(14,0); // смена цвета текста на жёлтый
}else{ // в другом случае
    setcolor(12,0); // смена цвета текста на красный
    cout << "Закрытие файла..." << endl;
    setcolor(14,0); // смена цвета текста на жёлтый
}

f.close(); // закрытие файла
}

void Sapartments(){ // функция для поиска свободных квартир

```



```

rent *m=new rent[nR]; // создание массива структур аренд

ifstream g("Expire.txt"); // открытие файла с истекшими арендами для чтения
int nS=0; // создание переменной для подсчёта количества истекших аренд
if(g){ // Проверка на существование файла
if(g.peek()!=EOF){ // Проверка на то, не пустой ли файл
while(!g.eof()){ // Пока не конец файла
g>>m[nS].Kod>>m[nS].Fam >> m[nS].Name >> m[nS].Day>>m[nS].Month >> m[nS].Year
>> m[nS].SrokA; // считывание информации из файла
nS++; // увеличение счётчика
}
nS--; // уменьшение переменной количества истекших аренд на 1 для того, чтобы не
учитывать последний пробел в файле
g.close(); // закрытие файла

apartment *s= new apartment[nA]; // создание массива структур квартир

for(int i=0;i<nS;i++) s[i].Kod = m[i].Kod; // присваивание кодов истекших аренд в новый
массив

for(int i=0;i<nS;i++)
for(int j=0;j<nA;j++)
if(s[i].Kod == a[j].Kod) s[i]=a[j]; // если нашли код истекшей квартиры в массиве квартир,
считываем данные о квартире

for(int i=1; i<nS; i++){ // сортировка вставками
int j = i;
while (s[j].Kkomnat<s[j-1].Kkomnat and j>0){ // сортировка по возрастанию количества
комнат
swap(s[j],s[j-1]);
j--;
}
}
}

```

```

for(int i=1; i<nS; i++){ // сортировка вставками
    int j = i;
    while (s[j].Kkomnat==s[j-1].Kkomnat and s[j].Price<s[j-1].Price and j>0){ // если
количество комнат равно, сортируем по возрастанию цены
        swap(s[j],s[j-1]);
        j--;
    }
}

setcolor(0,15); // смена цвета заднего фона на белый

cout<<endl<<"    Код    " <<"    Район    " <<"    Адрес    " <<"    Кол-во    " <<"
Этаж    " <<"    Цена    " <<endl;

cout<<" Квартиры                комнат                " <<endl; // Создание
шапки таблицы

for(int i=0; i<nS; i++){
    if(i%2==0) setcolor(0,8); // Проверка для смены цвета через строку. Смена цвета на тёмно-
серый
    else setcolor(0,7); // Смена цвета на светло-серый
    if(s[i].Kod==s[i+1].Kod)i++; // если код квартиры равен следующему, то увеличиваем
счётчик цикла на 1 и переходим на следующую строку
    else cout <<left<<"    " <<setw(11)<<s[i].Kod <<setw(19)<<s[i].Raion <<setw(28)<<s[i].Adres
<<setw(8)<<s[i].Kkomnat <<setw(8)<<s[i].Etaj <<s[i].Price<<endl; // в другом случае выводим
информацию
}
setcolor(14,0); // смена цвета текста на жёлтый
} else { // в другом случае
    setcolor(12,0); // смена цвета текста на красный
    cout <<"Нет свободных квартир." << endl;
setcolor(14,0); // смена цвета текста на жёлтый
}
} else { // в другом случае
    setcolor(12,0); // смена цвета текста на красный
    cout <<"Нет файла с свободными квартирами." << endl;
}

```

```

setcolor(14,0); // смена цвета текста на жёлтый
}

}

void Spros(){ // функция для поиска квартиры, пользующейся наибольшим спросом!!!
rent *m=r; // создаём и копируем структуру аренды в новую структуру
int k; // создание счётчика
    for(int i=1; i<nR; i++){ // сортировка вставками
int j = i;
    while (m[j].Kod<m[j-1].Kod and j>0){ // сортировка по возрастанию кода квартиры
        swap(m[j],m[j-1]);
        j--;
    }
}

int max=0,pos; // создание переменных для максимума и позиции
for(int i=0;i<nR;i++){
    k=0; // обнуляем счётчик
    for(int j=0;j<nR;j++)
        if(m[i].Kod == m[j].Kod) k++; // если коды квартир равны, то увеличиваем счётчик

    if(max<k){ // если максимум меньше счётчика
        max=k; // то максимум равен счётчику
        pos=i; // позиция равна i(счётчик цикла)
    }
}

for(int i=0;i<nA;i++)
    if(m[pos].Kod==a[i].Kod){ // если код квартиры структуры с позицией pos равен коду
квартиры структуры квартир с позицией i
        setcolor(0,15); // смена цвета заднего фона на белый

        cout<<endl<<"    Код    " <<"    Район    " <<"    Адрес    " <<"    Кол-во    " <<"
Этаж " <<"    Цена    " <<endl;

```

```
cout<<" Квартиры                                комнат                                "<<endl; // Создание шапки таблицы
```

```
    setcolor(0,8); // смена цвета текста на тёмно-серый  
    cout <<left<<"      "<<setw(11)<<a[i].Kod <<setw(19)<< a[i].Raion <<setw(28) <<a[i].Adres  
    <<setw(8)<<a[i].Kkomnat <<setw(8)<<a[i].Etaj <<a[i].Price<<endl; // Вывод информации о квартирах
```

```
    }
```

```
int *pov=new int[nA]; // создаём новый массив для удаления повторений
```

```
int g=0; // создаём счётчик для удаления повторений
```

```
    for(int i=0;i<nR;i++){
```

```
        k=0; // обнуляем счётчик
```

```
        for(int j=0;j<nR;j++)
```

```
            if(m[i].Kod == m[j].Kod) k++; // если коды квартир равны, то увеличиваем счётчик
```

```
        if(max==k and m[i].Kod!=m[pos].Kod) // если максимум равен счётчику и код не равен коду сравнения
```

```
            pov[g++]=i; // записываем в массив позиции
```

```
    }
```

```
    if(g>0){
```

```
    for(int i=0;i<g+1;i++)
```

```
        if(m[pov[i]].Kod == m[pov[i+1]].Kod){ // если коды квартиры равны
```

```
            for(int j=i; j<nA; j++)
```

```
                m[j]=m[j+1]; // сдвиг массива квартир влево(удаление)
```

```
            g--; //уменьшение счётчика
```

```
        }
```

```
    for(int i=0;i<g;i++)
```

```

    for(int c=0;c<nA;c++){
        if(i%2==0) setcolor(0,7); // Проверка для смены цвета через строку. Смена цвета
        текста на тёмно-серый
        else setcolor(0,8); // Смена цвета на светло-серый

        if(m[pov[i]].Kod==a[c].Kod) // если код квартиры структуры с позицией pov[i] равен коду
        квартиры структуры квартир с позицией c
            cout<<left<<"    "<<setw(11)<<a[c].Kod<<setw(19)<<a[c].Raion<<setw(28)<<a[c].Adres
            <<setw(8)<<a[c].Kkomnat<<setw(8)<<a[c].Etaj<<a[c].Price<<endl; // Вывод информации о
            квартирах
        }
    }
    cout<<endl;
    setcolor(14,0); // смена цвета текста на жёлтый
}

void Srprice() { // функция для поиска средней цены всех квартир с указанным количеством
комнат

    int kkomnat,k; // создание переменной для количества комнат и счётчика
    float Srsum; // создание переменной для средней цены
    int max=0; // создание переменной для вычисления максимальной комнаты
    for(int i=0;i<nA;i++)
        if(max<a[i].Kkomnat)max=a[i].Kkomnat; // поиск максимальной комнаты
        setcolor(10,0); // смена цвета текста на зелёный
        cout<<"\nВведите количество комнат квартиры(1-<<max<<): ";
    do{ // защита от неправильного ввода
        cin>> kkomnat; // ввод количества комнат
        if(kkomnat<1 or kkomnat>max){
            setcolor(12,0); // смена цвета текста на красный
            cout<<"Ошибка! Попробуйте ещё раз! "<<endl;
            setcolor(10,0); // смена цвета текста на зелёный
            cout<<"\nВведите количество комнат квартиры(1-<<max<<): ";
        }
    } while(kkomnat<1 or kkomnat>max); // защита от неправильного ввода
}

```

```

for(int i=0;i<nA;i++)
    if(a[i].Kkomnat== kkomnat){ // если количество комнат квартиры равно введённому
количеству комнат
        Srsum+=a[i].Price; // то суммируем цены квартир
        k++; // увеличиваем счётчик
    }
    Srsum/=k; // делим сумму цен квартир на количество
    setcolor(15,0); // смена цвета текста на белый
    cout<< "\nСредняя цена всех " << kkomnat << "-комнатных квартир: " << Srsum<<
endl<<endl;
    setcolor(14,0); // смена цвета текста на жёлтый
}

```

3) *main.cpp*

```
#include <iostream>
#include <conio.h>
#include "Apartments.h"
#include <windows.h>
using namespace std;

void setcolor(int text, int backG=0);

int main()
{
    int ch;
    do{
        menu();

        readA();
        readR();
        ch = mouse();
        switch(ch){
            case 1:{
                showA();
                system("pause");
                system("CLS");
                break;
            }
            case 2:{
                showR();
                system("pause");
                system("CLS");
                break;
            }
            case 3:{
                addA();
                system("pause");
                system("CLS");
```

```
        break;
    }
    case 4:{
        addR();
        system("pause");
        system("CLS");
        break;
    }
    case 5:{
        deleteInfoA();
        system("pause");
        system("CLS");
        break;
    }
    case 6:{
        deleteInfoR();
        system("pause");
        system("CLS");
        break;
    }
    case 7:{
        RedactA();
        system("pause");
        system("CLS");
        break;
    }
    case 8:{
        RedactR();
        system("pause");
        system("CLS");
        break;
    }
    case 9:{
        Srok();
        system("pause");
```



```

        system("CLS");
        break;
    }
    case 10:{
        Sapartments();
        system("pause");
        system("CLS");
        break;
    }
    case 11:{
        Spros();
        system("pause");
        system("CLS");
        break;
    }
    case 12:{
        Srprice();
        system("pause");
        system("CLS");
        break;
    }
    case 13: {
        setcolor(12,0);
        cout << "Выход...";
        setcolor(14,0);
        break;
    } default: system("CLS");
    }
    } while(ch!=13);
    return 0;
}

```