Министерство образования Республики Беларусь

Учреждение образования «Полоцкий государственный университет»

Факультет информационных технологий

Кафедра технологий программирования

**Лабораторная работа №4**

**По дисциплине «Основы алгоритмизации и программирования»**

**Тема: «Структуры и перечисления»**

Выполнил: Шиковец Е.А.

Группа 21-ИТ-1

Проверила: Пантелейко А.Ф

Преподаватель-стажер кафедры ТП

Полоцк 2022

**Цель работы:** научиться работать с векторными данными языка С++: структурами и перечислениями. Изучить правила объявления структур и перечислений, обращение к их полям, построение вложенных векторных типов данных.

**Вариант № 4**

29. Структура «УЧЕТ ЖИЛЬЦОВ». В домоуправлении ведется учет жильцов данного района, который включает следующие сведения: улица, номер дома, номер квартиры, метраж, число комнат, для каждого проживающего указывается: ФИО, дата рождения, дата прописки и выписки, отношение к владельцу квартиры. Определить число проживающих в однокомнатных квартирах, используя созданное меню.

1. Создать структуру для описания предметной области вашей курсовой работы.

2. В функции main() создать указатель на структуру из задания 1. Выделить динамическую память под эту структуру.

3. Написать функцию, которая будет считывать данные с клавиатуры и помещать их в структуру. Выделить дополнительную динамическую память в указателе на структуру под новую запись.

4. Написать функцию, которая будет печатать данные из структуры в формате "поле: значение\n".

5. Написать функцию, которая будет печатать данные из структуры в табличном формате.

#include <iostream>

#include <stdlib.h>

#include <iomanip>

#include <string.h>

using namespace std;

struct T\_inf // tenant info

{

// данные квартир

char \*street;

int house\_num;

int apart\_num;

int area;

int rooms;

// данные жильцов

char \*name;

char \*birth\_date;

char \*reg\_date; // дата прописки

char \*disch\_date; // дата выписки

char \*rel; // отношение к владельцу квартиры

};

T\_inf \*add\_tenant(struct T\_inf \*tenants, int pos)

{

system("CLS");

cout << "Size: " << pos << endl;

if (pos == 0)

{

tenants = (struct T\_inf \*)malloc(sizeof(struct T\_inf \*)); // выделяем память

}

else

{

tenants = (struct T\_inf \*)realloc(tenants, sizeof(struct T\_inf \*)); // перераспределяем память

}

cout << "-- Adding a tenant --" << endl;

// информация о квартире

cout << "Street: ";

tenants[pos].street = new char[20];

cin >> tenants[pos].street;

cout << "House number: ";

cin >> tenants[pos].house\_num;

cout << "Apartment number: ";

cin >> tenants[pos].apart\_num;

cout << "Apartment area: ";

cin >> tenants[pos].area;

cout << "Rooms: ";

cin >> tenants[pos].rooms;

// информация о жителе

cout << "Name: ";

tenants[pos].name = new char[20];

cin >> tenants[pos].name;

cout << "Date of birth: ";

tenants[pos].birth\_date = new char[20];

cin >> tenants[pos].birth\_date;

cout << "Regestration date: ";

tenants[pos].reg\_date = new char[20];

cin >> tenants[pos].reg\_date;

cout << "Discharge date: "; // дата выписки

tenants[pos].disch\_date = new char[20];

cin >> tenants[pos].disch\_date;

cout << "Relations: "; // отношение к владельцу квартиры

tenants[pos].rel = new char[20];

cin >> tenants[pos].rel;

system("CLS");

cout << tenants[pos].name << " was succesfuly added!\n\n";

return tenants; // возвращаем новый массив

}

void get\_data(struct T\_inf \*tenants, int pos)

{

int count = 0; // количество проживающих в однокомнатных квартирах

for (int i = 0; i <= pos; i++)

{

if (tenants[i].rooms == 1)

{

count++;

}

}

cout << "Number of tenants in 1 room apartments: " << count << endl;

}

void output(struct T\_inf \*tenants, int pos)

{

for (int i = 0; i < pos; i++)

{

cout << " Street: " << tenants[i].street << " House num: " << tenants[i].house\_num << " Apart num: " << tenants[i].apart\_num << " Area: " << tenants[i].area;

cout << " Rooms: " << tenants[i].rooms << " Name: " << tenants[i].name << " Birth date: " << tenants[i].birth\_date << " Regestration: " << tenants[i].reg\_date;

cout << " Disch date: " << tenants[i].disch\_date << " Relations: " << tenants[i].rel;

cout << endl;

}

}

void table(struct T\_inf \*tenants, int pos)

{

cout << setw(20) << "Street" << setw(20) << "House num" << setw(20) << "Apart num" << setw(20) << "Area" << setw(20) << "Rooms" << setw(20) << "Name" << setw(20) << "Birth date" << setw(20) << "Reg" << setw(20) << "Disch" << setw(20) << "Rel" << endl;

for (int i = 0; i < pos; i++)

{

cout << setw(20) << tenants[i].street << setw(20) << tenants[i].house\_num << setw(20) << tenants[i].apart\_num << setw(20) << tenants[i].area << setw(20) << tenants[i].rooms << setw(20) << tenants[i].name << setw(20) << tenants[i].birth\_date << setw(20) << tenants[i].reg\_date << setw(20) << tenants[i].disch\_date << setw(20) << tenants[i].rel;

cout << endl;

}

}

int main()

{

struct T\_inf \*tenants;

int n = 0, pos = 0;

while (n != 5)

{

cout << "-- Main menu --\n";

cout << "To add a tenant enter 1:\n";

cout << "To get data enter 2:\n";

cout << "Output 3:\n";

cout << "Table 4:\n";

cout << "To exit enter 5:\n";

cin >> n;

switch (n)

{

case 1:

{

system("CLS");

tenants = add\_tenant(tenants, pos);

pos++;

break;

}

case 2:

{

system("CLS");

get\_data(tenants, pos);

break;

}

case 3:

{

system("CLS");

output(tenants, pos);

break;

}

case 4:

{

system("CLS");

table(tenants, pos);

break;

}

}

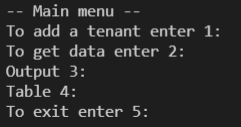
}

free(tenants); // освобождаем память

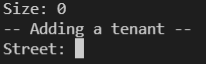
return 0;

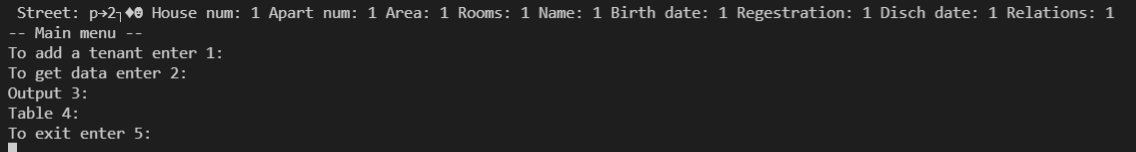
}

В main() написано меню с помощью цикла while со следующими параметрами:

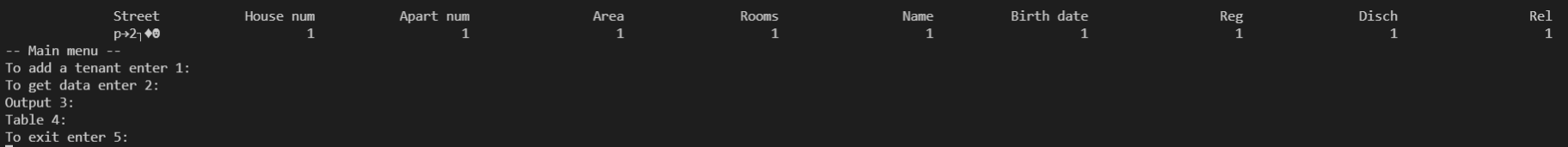


Также в main() объявляется структура. Присутствуют void функции: add\_tenant() – добавляет жильца, get\_data() – получаем ответ на задание, output() – вывод списка жильцов, table() – вывод списка жильцов в табличном формате.

** Добавление жильца**

****

**Вывод списка жильцов**

****

**Вывод списка жильцов в табличном формате**

**Вывод:**

Изучены принципы работы с памятью и функции для работы с ней. Разработана программа с использованием динамических структур на смежной памяти.