Министерство образования и науки Российской Федерации  
Федеральное‌ ‌государственное‌ ‌бюджетное‌ ‌образовательное‌ ‌учреждение‌

высшего‌ ‌образования‌

**«Пермский национальный исследовательский  
политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**О Т Ч Ё Т**

**по лабораторной работе №12**

Дисциплина: основы алгоритмизации и программирования

Тема: Строковый ввод-вывод

Выполнил работу

студент группы ИВТ-20-2б

Механошин А.А.

Проверила

Доцент кафедры ИТАС

к.т.н ПоляковаО.А.

Пермь, 2021

Структура данных

struct Data

{

string fio;

string city17;

string ulitsa;

int home;

int passport = 0;

};

Функция создания множества

void RANDOM(Data \* arr)

{

string DIO[] = {". . . " };

string ulits[] = { ". . ." };

string LEGOCITY[] = { ". . ."};

for (int i = 0; i < 100; i++)

{

int H = (sizeof(DIO) / sizeof(DIO[0]));

int R = rand() % H;

arr[i].fio = DIO[R];

arr[i].city17 = LEGOCITY[R];

arr[i].ulitsa = ulits[R];

arr[i].passport = 100000 + rand() % 99999;

arr[i].home = (rand() % 250);

}

}

Функция вывода функции

void print(Data\* arr)

{

for (int i = 0; i < 100; i++)

{

cout << "ФИО:" << arr[i].fio << endl;

cout << "Паспорт:" << arr[i].passport << endl;

cout << "Город:" << arr[i].city17 << endl;

cout << "Улица:" << arr[i].ulitsa << endl;

cout << "Дом:" << arr[i].home << endl;

}

}

Функция сортировки множества по номеру дома

void Sort(Data\*& arr)

{

Data temp;

for (int i = 0; i < 100; i++) {

for (int j = 0; j < 100 - 1; j++) {

if (arr[j].home > arr[j + 1].home)

{

temp = arr[j];

arr[j] = arr[j + 1];

arr[j + 1] = temp;

}

}

}

printh(arr);

};

Метод прямого поиска

void search1(Data\* arr,string key)

{

for (int i = 0; i < 100; i++)

{

string result = arr[i].city17 + " " + arr[i].ulitsa + " " + to\_string(arr[i].home);

if (result == key)

{

cout << "ФИО:" << arr[i].fio << endl;

cout << "Паспорт:" << arr[i].passport << endl;

cout << "Город:" << arr[i].city17 << endl;

cout << "Улитса:" << arr[i].ulitsa << endl;

cout << "Дом:" << arr[i].home << endl;

}

}

}

Метод прямого поиска подстроки в строке

void search2(Data\* arr, string key)

{

bool f = 0;

int j = 0;

int l = 0;

int p = 0;

int count = 0;

int b = key.length();

for (int i = 0; i < 100; i++)

{

string result = arr[i].city17 + " " + arr[i].ulitsa + " " + to\_string(arr[i].home);

int a = result.length();

bool f = 0;

while (l < key.length() && j < (a-b) && f!= true)

{

p = j;

while (key[l] == result[j] && f!= true )

{

count++; j++; l++;

if (count == key.length())

{

f = true;

cout << "ФИО:" << arr[i].fio << endl;

cout << "Паспорт:" << arr[i].passport << endl;

cout << "Город:" << arr[i].city17 << endl;

cout << "Улица:" << arr[i].ulitsa << endl;

cout << "Дом:" << arr[i].home << endl;

}

}

j = p;

count = 0;

l = 0;

j++;

}

j = 0;

l = 0;

}

}

Метод итерационного поиска

int search3(Data\* arr, string key1)

{

Sort(arr);

string keyc = key1;

bool f = 0;

int i = key1.length();

while (i > 0 && f != 1)

{

if (key1[i] == ' ')

{

{

keyc.erase(0, i+1);

f = 1;

}

}

i--;

}

int key = stoi(keyc);

int mid, left = 0, right = 250 ;

while (arr[left].home <= key && arr[right].home >= key)

{

mid = left + ((key - arr[left].home) \* (right - left) / (arr[right].home - arr[left].home));

if (arr[mid].home < key) left = mid + 1;

else if (arr[mid].home > key) right = mid - 1;

else

{

{

bool f = 0;

while (key == arr[mid].home)

{

string result = arr[mid].city17 + " " + arr[mid].ulitsa + " " + to\_string(arr[mid].home);

if (result == key1)

{

return mid;

f = 1;

}

else

{

mid++;

}

}

if (f != 1) return -1;

}

}

}

if (arr[left].home == key) return left;

else return -1;

}

Меню

void Menu(Data\* arr)

{

bool f = 1;

string key;

while (f == 1)

{

cout << "Меню:" << endl << "открыть спиcок 1" << endl << "сохранить список 2" << endl << "поиск элемента по ключу 3" << endl;

int c;

cin >> c;

switch (c)

{

case(1): print(arr); break;

case(2): break;

case(3):

{

int c1;

int c2 = 0;

cout <<"Метод прямого поиска 1" << endl <<"Метод прямого поиска подстроки в строке 2" << endl <<"Итерационный поиск 3" <<endl << "выход press any key" << endl;

cin >> c1;

switch (c1)

{

case(1):cout << "Введите адрес полностью" << endl;

cin.ignore(); getline(cin, key); search1(arr, key); break;

case(2):cout << "Введите адрес полностью или частично" << endl;

cin.ignore(); getline(cin, key); search2(arr, key); break;

case(3): cout << "Введите адрес полностью" << endl;

while (c2 == 0)

{

cin.ignore();

getline(cin, key);

c2 = search3(arr, key);

switch (c2)

{

case (-1): cout << "не найдено" << endl; break;

case (0): cout << "Ввод не корректный" << endl; break;

default:

cout << "Найдено!" << endl;

cout << "ФИО:" << arr[c2].fio << endl;

cout << "Паспорт:" << arr[c2].passport << endl;

cout << "Город:" << arr[c2].city17 << endl;

cout << "Улица:" << arr[c2].ulitsa << endl;

cout << "Дом:" << arr[c2].home << endl;

break;

}

}

break;

default: break;

}

}

default: f = 0;

break;

}

}}

Main

int main()

{

system("chcp 1251>nul");

srand(time(0));

Data\* arr = new Data[100];

RANDOM(arr);

Menu(arr);return 0;}

Текст программы

#include <iostream>

#include <ctime>

#include <string>

#include <ostream>

#include <stdlib.h>

using namespace std;

struct Data

{

string fio;

string city17;

string ulitsa;

int home;

int passport = 0;

};

void RANDOM(Data \* arr)

{

string DIO[] = {"Юдина Василиса Дмитриевна" ,"Климов Иван Даниилович" ,"Жмышенко Михаил Петрович" ,"Пупкин Владислав Альюертович" ,"Диего Сергеевич Брандо" ,"Афанасьева Ольга Тихоновна" ,"Николаева Вера Егоровна" ,"Баранов Степан Михайлович" ,"Логинов Максим Антонович" ,"Логинов Максим Антонович" ,"Волков Савва Захарович" ,"Платонова Антонина Кирилловна" ,"Павлов Владислав Ярославович" ,"Власов Платон Григорьевич" ,"Голубев Роберт Богданович" ,"Соколова Елизавета Никитична" ,"Никитина Полина Святославовна" ,"Моисеева Мария Дмитриевна" ,"Тимофеева Виктория Фёдоровна" ,"Тимофеева Виктория Фёдоровна" ,"Пономарева Элина Юрьевна" ,"Морозов Макар Давидович" ,"Ефимов Юрий Платонович" };

string ulits[] = { "Ленина","Мира","Дедюкина","Стаханова","Сталина","Красной Армии","Сапожника","Акихабара","Собянина","Кривичей","Зифиркина","Набережная","Комсомольский проспект","Коровкина","Полосухина","Перекатная","Чукотская","Кисельная","Карпинского","Спартаковская","Уездный Проспект","Ульяновская","Карамельная","Сахарная","Озерная","Боброва","Каменная","Пирогова", };

string LEGOCITY[] = { "LEGOCITY","Москва","Калыма","Рязань","Казань","Серов","Югорск","Нижняя Тура","Верхняя Тура","Нижний Тагил","Суздаль","Коноха","Сити17","Готем","Метрополис","Зверополис","Токио","Осака","Киото","Екатеринбург","Горнозаводск","Краснотуринск","Берлин","Владивосток","Чебаркуль","Златоуст","Уфа","Новосибирск", };

for (int i = 0; i < 100; i++)

{

int H = (sizeof(DIO) / sizeof(DIO[0]));

int R = rand() % H;

arr[i].fio = DIO[R];

arr[i].city17 = LEGOCITY[R];

arr[i].ulitsa = ulits[R];

arr[i].passport = 100000 + rand() % 99999;

arr[i].home = (rand() % 250);

}

}

void print(Data\* arr)

{

for (int i = 0; i < 100; i++)

{

cout << "ФИО:" << arr[i].fio << endl;

cout << "Паспорт:" << arr[i].passport << endl;

cout << "Город:" << arr[i].city17 << endl;

cout << "Улица:" << arr[i].ulitsa << endl;

cout << "Дом:" << arr[i].home << endl;

}

}

void printh(Data\* arr)

{

for (int i = 0; i < 100; i++)

{

cout << "Home " << i+1 << " :"<<arr[i].home << endl;

}

}

void Sort(Data\*& arr)

{

Data temp;

for (int i = 0; i < 100; i++) {

for (int j = 0; j < 100 - 1; j++) {

if (arr[j].home > arr[j + 1].home)

{

temp = arr[j];

arr[j] = arr[j + 1];

arr[j + 1] = temp;

}

}

}

printh(arr);

}

void search1(Data\* arr,string key)

{

for (int i = 0; i < 100; i++)

{

string result = arr[i].city17 + " " + arr[i].ulitsa + " " + to\_string(arr[i].home);

if (result == key)

{

cout << "ФИО:" << arr[i].fio << endl;

cout << "Паспорт:" << arr[i].passport << endl;

cout << "Город:" << arr[i].city17 << endl;

cout << "Улитса:" << arr[i].ulitsa << endl;

cout << "Дом:" << arr[i].home << endl;

}

}

}

void search2(Data\* arr, string key)

{

bool f = 0;

int j = 0;

int l = 0;

int p = 0;

int count = 0;

int b = key.length();

for (int i = 0; i < 100; i++)

{

string result = arr[i].city17 + " " + arr[i].ulitsa + " " + to\_string(arr[i].home);

int a = result.length();

bool f = 0;

while (l < key.length() && j < (a-b) && f!= true)

{

p = j;

while (key[l] == result[j] && f!= true )

{

count++; j++; l++;

if (count == key.length())

{

f = true;

cout << "ФИО:" << arr[i].fio << endl;

cout << "Паспорт:" << arr[i].passport << endl;

cout << "Город:" << arr[i].city17 << endl;

cout << "Улица:" << arr[i].ulitsa << endl;

cout << "Дом:" << arr[i].home << endl;

}

}

j = p;

count = 0;

l = 0;

j++;

}

j = 0;

l = 0;

}

}

int search3(Data\* arr, string key1)

{

Sort(arr);

string keyc = key1;

bool f = 0;

int i = key1.length();

while (i > 0 && f != 1)

{

if (key1[i] == ' ')

{

{

keyc.erase(0, i+1);

f = 1;

}

}

i--;

}

int key = stoi(keyc);

int mid, left = 0, right = 250 ;

while (arr[left].home <= key && arr[right].home >= key)

{

mid = left + ((key - arr[left].home) \* (right - left) / (arr[right].home - arr[left].home));

if (arr[mid].home < key) left = mid + 1;

else if (arr[mid].home > key) right = mid - 1;

else

{

{

bool f = 0;

while (key == arr[mid].home)

{

string result = arr[mid].city17 + " " + arr[mid].ulitsa + " " + to\_string(arr[mid].home);

if (result == key1)

{

return mid;

f = 1;

}

else

{

mid++;

}

}

if (f != 1) return -1;

}

}

}

if (arr[left].home == key) return left;

else return -1;

}

void Menu(Data\* arr)

{

bool f = 1;

string key;

while (f == 1)

{

cout << "Меню:" << endl << "открыть спиcок 1" << endl << "сохранить список 2" << endl << "поиск элемента по ключу 3" << endl;

int c;

cin >> c;

switch (c)

{

case(1): print(arr); break;

case(2): break;

case(3):

{

int c1;

int c2 = 0;

cout <<"Метод прямого поиска 1" << endl <<"Метод прямого поиска подстроки в строке 2" << endl <<"Итерационный поиск 3" <<endl << "выход press any key" << endl;

cin >> c1;

switch (c1)

{

case(1):cout << "Введите адрес полностью" << endl;

cin.ignore(); getline(cin, key); search1(arr, key); break;

case(2):cout << "Введите адрес полностью или частично" << endl;

cin.ignore(); getline(cin, key); search2(arr, key); break;

case(3): cout << "Введите адрес полностью" << endl;

while (c2 == 0)

{

cin.ignore();

getline(cin, key);

c2 = search3(arr, key);

switch (c2)

{

case (-1): cout << "не найдено" << endl; break;

case (0): cout << "Ввод не корректный" << endl; break;

default:

cout << "Найдено!" << endl;

cout << "ФИО:" << arr[c2].fio << endl;

cout << "Паспорт:" << arr[c2].passport << endl;

cout << "Город:" << arr[c2].city17 << endl;

cout << "Улица:" << arr[c2].ulitsa << endl;

cout << "Дом:" << arr[c2].home << endl;

break;

}

}

break;

default: break;

}

}

default: f = 0;

break;

}

}

}

int main()

{

system("chcp 1251>nul");

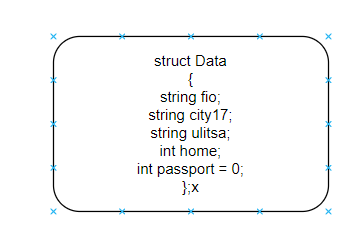
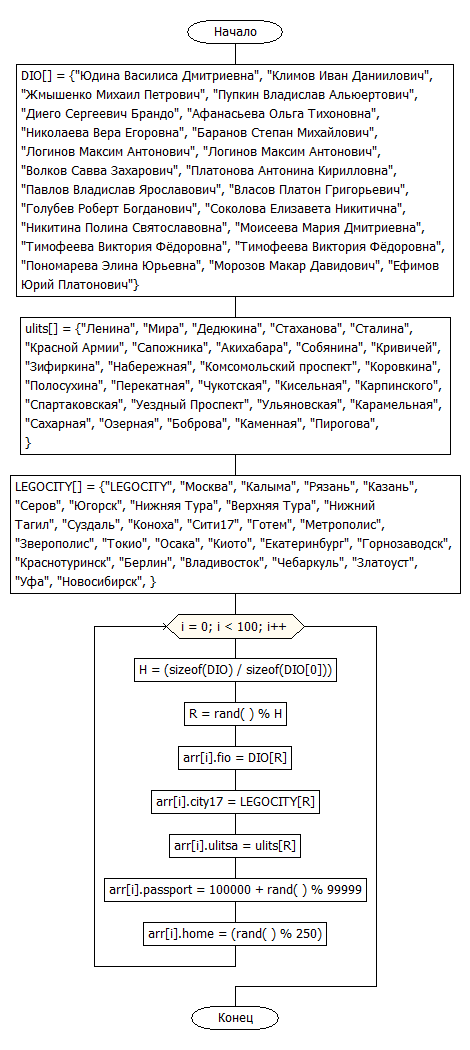
srand(time(0));

Data\* arr = new Data[100];

RANDOM(arr);

Menu(arr);

return 0;}

Блок схема

