#include <list>

#include <iostream>

#include <string>

#include<fstream>

using namespace std;

structSklad {

int number;

string typeTovar;

string adress;

intcountPlase;

intadminID;

};

void Add(list<Sklad>skladList, list<Sklad>::iterator ptr);

void Remove(list<Sklad>skladList, list<Sklad>::iterator ptr);

void Find(list<Sklad>skladList, list<Sklad>::iterator ptr);

void Save(list<Sklad>skladList, list<Sklad>::iterator ptr);

void Print(list<Sklad>skladList, list<Sklad>::iterator ptr);

void Sort(list<Sklad>skladList, list<Sklad>::iterator ptr);

void Exit(list<Sklad>skladList, list<Sklad>::iterator ptr);

void menu(list<Sklad>skladList, list<Sklad>::iterator ptr);

void Add(list<Sklad>skladList, list<Sklad>::iterator ptr) {

SkladnewSklad;

cout<< "Введите номер склада" <<endl;

cin>>newSklad.number;

cout<< "Введите тип товара" <<endl;

cin>>newSklad.typeTovar;

cout<< "Ведите адрес" <<endl;

cin>>newSklad.adress;

cout<< "Ведите количество мест" <<endl;

cin>>newSklad.countPlase;

cout<< "Ведите номер сотрудника" <<endl;

cin>>newSklad.adminID;

skladList.push\_back(newSklad);

menu(skladList, ptr);

}

void Remove(list<Sklad>skladList, list<Sklad>::iterator ptr) {

int n;

cout<< "Введите номер склада для удаления";

cin>> n;

ptr = skladList.begin();

for (inti = 0; i<skladList.size() - 1; i++) {

if (ptr->number == n) {

ptr = skladList.erase(ptr);

ptr--;

}

ptr++;

}

if (ptr->number == n) {

skladList.pop\_back();

}

menu(skladList, ptr);

}

void Find(list<Sklad>skladList, list<Sklad>::iterator ptr) {/\*делаем самостоятельно\*/}

void Save(list<Sklad>skladList, list<Sklad>::iterator ptr) {

ofstreamfileSclad("sklad1.txt");//после отладки программы меняем на sklad.txt

if(!fileSclad) cout<< "Склад открыть невозможно" <<endl;

string str;

for (ptr = skladList.begin(); ptr != skladList.end(); ptr++) {

str = to\_string(ptr->number) + ";" + ptr->typeTovar + ";" + ptr->adress + ";" + to\_string(ptr->countPlase) + ";" + to\_string(ptr->adminID);

fileSclad<<str<<endl;

}

menu(skladList, ptr);

}

void Sort(list<Sklad>skladList, list<Sklad>::iterator ptr) {

int n;

cout<< "Выберите направление сортировки" <<endl;

cout<< "По возрастанию -- 1" <<endl;

cout<< "По убыванию --2 " <<endl;

cin>> n;

if (n == 2) {

skladList.sort([](constSklad& a, constSklad& b) {return a.countPlase>b.countPlase; });

}

else

{

skladList.sort([](constSklad& a, constSklad& b) {return a.countPlase<b.countPlase; });

}

menu(skladList, ptr);

}

void Print(list<Sklad>skladList, list<Sklad>::iterator ptr) {

for (ptr = skladList.begin(); ptr != skladList.end(); ptr++) {

cout<<ptr->number << " " <<ptr->typeTovar<< " " <<ptr->adress<< " " <<ptr->countPlase<< " " <<ptr->adminID<<endl;

}

menu(skladList, ptr);

}

void Exit(list<Sklad>skladList, list<Sklad>::iterator ptr) {

string ans;

cout<< "Завершить работу? y/n" <<endl;

cin>>ans;

if (ans == "y")

{

cout<< "Сохранить изменения перед выходом y/n" <<endl;

cin>>ans;

if (ans == "y")

{

//фрагмент функции Save

ofstreamfileSclad("sklad1.txt");

if(!fileSclad) cout<< " файл открыть невозможно!" <<endl;

string str;

for (ptr = skladList.begin(); ptr != skladList.end(); ptr++) {

str += to\_string(ptr->number) + ";" + ptr->typeTovar + ";" + ptr->adress + ";" + to\_string(ptr->countPlase) + ";" + to\_string(ptr->adminID);

fileSclad<<str<<endl;

str = "";

}

exit(2);

}

else exit(2);

}

else menu(skladList, ptr);

}

void menu(list<Sklad>skladList, list<Sklad>::iterator ptr) {

cout<< "Выберите команду" <<endl;

cout<< "Добавить запись -- 1 " <<endl;

cout<< "Удалить запись -- 2 " <<endl;

cout<< "Поиск по номеру склада-- 3 " <<endl;

cout<< "Сортировка по количеству мест -- 4 " <<endl;

cout<< "сохранение изменений -- 5 " <<endl;

cout<< "вывод данных на экран -- 6 " <<endl;

cout<< "завершить работу -- 7 " <<endl;

intpunct;

cin>>punct;

switch (punct)

{

case 1:

Add(skladList, ptr);

break;

case 2:

Remove(skladList, ptr);

break;

case 3:

Find(skladList, ptr);

break;

case 4:

Sort(skladList, ptr);

break;

case 5:

Save(skladList, ptr);

break;

case 6:

Print(skladList, ptr);

break;

case 7:

Exit(skladList, ptr);

break;

default:

cout<< "Введите число от 1-5" <<endl;

menu(skladList, ptr);

break;

}

}

int main(){

setlocale(LC\_ALL, "Rus");

list<Sklad>skladList;

list<Sklad>::iterator ptr;

ifstreamfileSclad("sklad.txt");

string skladCortej;

intendVal[4], n=0;

while (getline(fileSclad, skladCortej))

{

for (inti = 0; i< 4; i++) {

n = skladCortej.find(";", n);

endVal[i] = n;

n++;

}

n = 0;

SkladinfoSklad;

infoSklad.number = stoi(skladCortej.substr(0, endVal[0]));

infoSklad.typeTovar = skladCortej.substr(endVal[0] + 1, endVal[1] - endVal[0] - 1);

infoSklad.adress = skladCortej.substr(endVal[1] + 1, endVal[2] - endVal[1] - 1);

infoSklad.countPlase = stoi(skladCortej.substr(endVal[2] + 1, endVal[3] - endVal[2] - 1));

infoSklad.adminID = stoi(skladCortej.substr(endVal[3] + 1, skladCortej.length()));

skladList.push\_back(infoSklad);

}

menu(skladList, ptr);

}