**Компьютерная академия шаг**

Итоговый проект

по курсу Процедурное программирование на С

на тему «Салон Красоты»

Выполнили:

студенты группы 02ПР33

- Савченко Денис Андреевич

- Логвинов Дмитрий Андреевич

Харьков, 2018 г.

1 Постановка задачи

Требуется создать программную систему, предназначенную для работников салона красоты. Такая система должна хранить сведений о каждом из видов услуги, который предоставляет салон, таких как: парикмахерская, массаж, маникюр и педикюр, косметология и визаж.

Функциональность данной программной системы заключается в том, чтоб принимать заказы клиентов и после всех процедур выводить чек с полным перечнем использованных услуг и суммарной суммой. Также, администраторы салона могут распечатывать прайс-лист по каждому виду услуг, и сортировать каждый вид услуг по цене с последующим выводом на экран. В программном обеспечении, также, есть учётная запись менеджера салона, который может добавлять новый подвид услуги к уже ранее добавленным видам и записью его в файл. Удаления видов услуг также присутствует в меню менеджера. Поскольку мы живём в очень нестабильной стране, то у менеджера есть право менять цены за выбранную им услугу.

Программное обеспечение даёт возможность выбора языка (русский и английский), поскольку в салоне работают иностранцы. Также внизу меню вам будет всегда предоставлена текущая дата и время.

2 Проектирование системы

MyString

Строка(char)

Длинна(int)

Date

Год(int)

Месяц(int)

День(int)

Часы(int)

Минуты(int)

BarberShop

NailsService

Сosmetology

MakeUp

Имя мастера(MS)

Тип услуги(MS)

Цена (float)

Massage

Имя мастера(MS)

Тип услуги(MS)

Время(int)

Цена(float)

Main.cpp

#pragma once

#include"Services.h"

int main() {

setlocale(0, "");

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int N\_foradmins; char user\_tatiana[] = "Tatiana";

Admins \*admins = nullptr; char user\_eveline[] = "Eveline";

admins = arrFromFileadmins(N\_foradmins); char user\_maria[] = "Maria";

int N\_forbarber;

BarberShop \*barber = nullptr;

barber = ArrFromFilebarber(N\_forbarber);

int N\_formassage;

Massage \*massage = nullptr;

massage = ArrFromFilemassage(N\_formassage);

int N\_fornails;

NailsService \*nails = nullptr;

nails = ArrFromFilenails(N\_fornails);

int N\_forcosm;

Cosmetology \*cosmetology = nullptr;

cosmetology = ArrFromFilecosmet(N\_forcosm);

int N\_formakeup;

MakeUp \*makeup = nullptr;

makeup = ArrFromFilemakeup(N\_formakeup);

int N = 0;

Check \*check = nullptr;

Date nowdate;

nowdate.getNowDate();

FILE\*file = nullptr;

int menu1 = 0; int bar = 0; int mak = 0; char type\_of\_sevice[100] = "";

int menu2 = 0; int mas = 0; float totalprice = 0; float someprice = 0;

int menu3 = 0; int nail = 0; char password[50] = "";

int menu4 = 0; int cosm = 0; char name\_of\_master[60] = "";

do {

system("cls");

cout << " \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

cout << "| |" << endl;

cout << "|########################################################################|" << endl;

cout << "|#\t\t\tPlease, select your language #|" << endl;

cout << "|#\t\t\tПожалуйста, выберите язык #|" << endl;

cout << "|########################################################################|" << endl;

cout << "|#\t\t(1) - Английский (English) #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(2) - Русский (Russian) #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(3) - Exit #|" << endl;

cout << "|########################################################################|" << endl;

cout << "|# "; nowdate.printDate(); cout << "\t\t#|" << endl;

cout << "|########################################################################|" << endl;

cout << "|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|" << endl;

cin >> menu1;

switch (menu1) {

case 1://///////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////1/

do {

system("cls");

cout << " \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

cout << "| |" << endl;

cout << "|########################################################################|" << endl;

cout << "|#\t\t\tPlease select your account #|" << endl;

cout << "|########################################################################|" << endl;

cout << "|#\t\t(1) - Administrator Tatiana #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(2) - Administrator Eveline #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(3) - Manager Maria #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(4) - Exit #|" << endl;

cout << "|########################################################################|" << endl;

cout << "|# "; nowdate.printDate(); cout << "\t\t#|" << endl;

cout << "|########################################################################|" << endl;

cout << "|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|" << endl;

cin >> menu2;

switch (menu2) {

case 1:////////////////////////////////////////////////////////////////////////////////////////////////////////////////////2

cout << "Please enter your password -> ";

cin.get();

cin.getline(password, 50);

if (checkPassword(admins, N\_foradmins, user\_tatiana, password) == true) { // - функция проверяющая пароли

do {

system("cls");

cout << " \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

cout << "| |" << endl;

cout << "|########################################################################|" << endl;

cout << "|#\t\t\tHello Tatiana #|" << endl;

cout << "|########################################################################|" << endl;

cout << "|#\t(1) - Accept the Order #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t(2) - Print price-list of each services #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t(3) - Sort by price of each services #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t(4) - Exit #|" << endl;

cout << "|########################################################################|" << endl;

cout << "|# "; nowdate.printDate(); cout << "\t\t#|" << endl;

cout << "|########################################################################|" << endl;

cout << "|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|" << endl;

cin >> menu3;

switch (menu3) {

case 1:///////////////////////////////////////////////////////////////////////3

do {

system("cls");

cout << " \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

cout << "| |" << endl;

cout << "|########################################################################|" << endl;

cout << "|#\t\t\tAccept the Order #|" << endl;

cout << "|########################################################################|" << endl;

cout << "|#\t\t(1) - BarberShop #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(2) - Massage #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(3) - Nails Services #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(4) - Cosmetology #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(5) - MakeUp #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(6) - Print Check #|" << endl;

cout << "|########################################################################|" << endl;

cout << "|# "; nowdate.printDate(); cout << "\t\t#|" << endl;

cout << "|########################################################################|" << endl;

cout << "|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|" << endl;

cin >> menu4;

switch (menu4) {

case 1:////////////////////////////////////////////////////////4

bar++;

check = addCheck(check, N);

AddToFileCheck(check, N);

/\*check = AddFromFileCheck(N);

printCheckEngl(check, N);

system("pause");\*/

break;

case 2://///////////////////////////////////////////////////////4

mas++;

check = addCheck(check, N);

AddToFileCheck(check, N);

break;

case 3://///////////////////////////////////////////////////////4

nail++;

check = addCheck(check, N);

AddToFileCheck(check, N);

break;

case 4://///////////////////////////////////////////////////////4

cosm++;

check = addCheck(check, N);

AddToFileCheck(check, N);

break;

case 5://///////////////////////////////////////////////////////4

mak++;

check = addCheck(check, N);

AddToFileCheck(check, N);

break;

case 6://///////////////////////////////////////////////////////4

system("cls");

if (bar != 0 || mas != 0 || nail != 0 || cosm != 0 || mak != 0) {

totalPrice(check, N, totalprice);

cout.width();

cout << "#################################################################" << endl;

cout << "# ! Check ! #" << endl;

cout << "#################################################################" << endl;

printCheckEngl(check, N);

cout << "#################################################################" << endl;

cout.width(); cout << "# TOTAL PRICE : " << totalprice << " grn. #" << endl;

nowdate.printDateforChek();

cout << "#################################################################" << endl;

}

else

perror("\t\tNot enough information");

system("pause");

break;

default:perror("\t\tError\n"); perror("\t\tTry again later"); system("pause"); break;

}

} while (menu4 != 6);

break;

break;

case 2:///////////////////////////////////////////////////////////////////////3

do {

system("cls");

cout << " \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

cout << "| |" << endl;

cout << "|########################################################################|" << endl;

cout << "|#\t\t\tPrint price-list of each services #|" << endl;

cout << "|########################################################################|" << endl;

cout << "|#\t\t(1) - BarberShop #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(2) - Massage #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(3) - Nails Services #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(4) - Cosmetology #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(5) - MakeUp #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(6) - Exit #|" << endl;

cout << "|########################################################################|" << endl;

cout << "|# "; nowdate.printDate(); cout << "\t\t#|" << endl;

cout << "|########################################################################|" << endl;

cout << "|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|" << endl;

cin >> menu4;

switch (menu4) {

case 1:////////////////////////////////////////////////////////4

printarrbarberEngl(barber, N\_forbarber);

system("pause");

break;

case 2://///////////////////////////////////////////////////////4

printarrmassageEngl(massage, N\_formassage);

system("pause");

break;

case 3://///////////////////////////////////////////////////////4

printarrnailsEngl(nails, N\_fornails);

system("pause");

break;

case 4://///////////////////////////////////////////////////////4

printarrcosmetEngl(cosmetology, N\_forcosm);

system("pause");

break;

case 5:

printarrmakeupEngl(makeup, N\_formakeup);

system("pause");

break;

default:perror("\t\tError\n"); perror("\t\tTry again later"); break;

}

} while (menu4 != 6);

break;

case 3:///////////////////////////////////////////////////////////////////////3

do {

system("cls");

cout << " \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

cout << "| |" << endl;

cout << "|########################################################################|" << endl;

cout << "|#\t\t\tSort by price of each services #|" << endl;

cout << "|########################################################################|" << endl;

cout << "|#\t\t(1) - BarberShop #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(2) - Massage #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(3) - Nails Services #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(4) - Cosmetology #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(5) - MakeUp #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(6) - Exit #|" << endl;

cout << "|########################################################################|" << endl;

cout << "|# "; nowdate.printDate(); cout << "\t\t#|" << endl;

cout << "|########################################################################|" << endl;

cout << "|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|" << endl;

cin >> menu4;

switch (menu4) {

case 1:////////////////////////////////////////////////////////4

sortBarberPrice(barber, N\_forbarber);

printarrbarberEngl(barber, N\_forbarber);

system("pause");

break;

case 2://///////////////////////////////////////////////////////4

sortMassagePrice(massage, N\_formassage);

printarrmassageEngl(massage, N\_formassage);

system("pause");

break;

case 3://///////////////////////////////////////////////////////4

sortNailsServicePrice(nails, N\_fornails);

printarrnailsEngl(nails, N\_fornails);

system("pause");

break;

case 4://///////////////////////////////////////////////////////4

sortCosmetologyPrice(cosmetology, N\_forcosm);

printarrcosmetEngl(cosmetology, N\_forcosm);

system("pause");

break;

case 5://///////////////////////////////////////////////////////4

sortMakeUpPrice(makeup, N\_formakeup);

printarrmakeupEngl(makeup, N\_formakeup);

system("pause");

break;

case 6:

break;

default:perror("\t\tError\n"); perror("\t\tTry again later"); break;

}

} while (menu4 != 6);

break;

case 4:///////////////////////////////////////////////////////////////////////3

cout << "\t\t\t\t\t\BYE Tatiana :-)\n";

system("pause");

break;

default:perror("\t\tError\n"); perror("\t\tTry again "); break;

}

} while (menu3 != 4);

break;

}

else {

perror("Passwords don`t match\nTry Again :(");

system("pause");

break;

}

break;

case 2:

cout << "Please enter your password -> ";

cin.get();

cin.getline(password, 50);

if (checkPassword(admins, N\_foradmins, user\_eveline, password) == true) { // - функция проверяющая пароли

do {

system("cls");

cout << " \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

cout << "| |" << endl;

cout << "|########################################################################|" << endl;

cout << "|#\t\t\tHello Eveline #|" << endl;

cout << "|########################################################################|" << endl;

cout << "|#\t(1) - Accept the Order #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t(2) - Print price-list of each services #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t(3) - Sort by price of each services #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t(4) - Exit #|" << endl;

cout << "|########################################################################|" << endl;

cout << "|# "; nowdate.printDate(); cout << "\t\t#|" << endl;

cout << "|########################################################################|" << endl;

cout << "|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|" << endl;

cin >> menu3;

switch (menu3) {

case 1:///////////////////////////////////////////////////////////////////////3

do {

system("cls");

cout << " \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

cout << "| |" << endl;

cout << "|########################################################################|" << endl;

cout << "|#\t\t\tAccept the Order #|" << endl;

cout << "|########################################################################|" << endl;

cout << "|#\t\t(1) - BarberShop #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(2) - Massage #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(3) - Nails Services #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(4) - Cosmetology #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(5) - MakeUp #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(6) - Print Check #|" << endl;

cout << "|########################################################################|" << endl;

cout << "|# "; nowdate.printDate(); cout << "\t\t#|" << endl;

cout << "|########################################################################|" << endl;

cout << "|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|" << endl;

cin >> menu4;

switch (menu4) {

case 1:////////////////////////////////////////////////////////4

bar++;

check = addCheck(check, N);

AddToFileCheck(check, N);

/\*check = AddFromFileCheck(N);

printCheckEngl(check, N);

system("pause");\*/

break;

case 2://///////////////////////////////////////////////////////4

mas++;

check = addCheck(check, N);

AddToFileCheck(check, N);

break;

case 3://///////////////////////////////////////////////////////4

nail++;

check = addCheck(check, N);

AddToFileCheck(check, N);

break;

case 4://///////////////////////////////////////////////////////4

cosm++;

check = addCheck(check, N);

AddToFileCheck(check, N);

break;

case 5://///////////////////////////////////////////////////////4

mak++;

check = addCheck(check, N);

AddToFileCheck(check, N);

break;

case 6://///////////////////////////////////////////////////////4

system("cls");

if (bar != 0 || mas != 0 || nail != 0 || cosm != 0 || mak != 0) {

totalPrice(check, N, totalprice);

cout.width();

cout << "#################################################################" << endl;

cout << "# ! Check ! #" << endl;

cout << "#################################################################" << endl;

printCheckEngl(check, N);

cout << "#################################################################" << endl;

cout.width(); cout << "# TOTAL PRICE : " << totalprice << " grn. #" << endl;

nowdate.printDateforChek();

cout << "#################################################################" << endl;

}

else

perror("\t\tNot enough information");

system("pause");

break;

default:perror("\t\tError\n"); perror("\t\tTry again later"); system("pause"); break;

}

} while (menu4 != 6);

break;

break;

case 2:///////////////////////////////////////////////////////////////////////3

do {

system("cls");

cout << " \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

cout << "| |" << endl;

cout << "|########################################################################|" << endl;

cout << "|#\t\t\tPrint price-list of each services #|" << endl;

cout << "|########################################################################|" << endl;

cout << "|#\t\t(1) - BarberShop #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(2) - Massage #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(3) - Nails Services #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(4) - Cosmetology #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(5) - MakeUp #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(6) - Exit #|" << endl;

cout << "|########################################################################|" << endl;

cout << "|# "; nowdate.printDate(); cout << "\t\t#|" << endl;

cout << "|########################################################################|" << endl;

cout << "|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|" << endl;

cin >> menu4;

switch (menu4) {

case 1:////////////////////////////////////////////////////////4

printarrbarberEngl(barber, N\_forbarber);

system("pause");

break;

case 2://///////////////////////////////////////////////////////4

printarrmassageEngl(massage, N\_formassage);

system("pause");

break;

case 3://///////////////////////////////////////////////////////4

printarrnailsEngl(nails, N\_fornails);

system("pause");

break;

case 4://///////////////////////////////////////////////////////4

printarrcosmetEngl(cosmetology, N\_forcosm);

system("pause");

break;

case 5:

printarrmakeupEngl(makeup, N\_formakeup);

system("pause");

break;

default:perror("\t\tError\n"); perror("\t\tTry again later"); break;

}

} while (menu4 != 6);

break;

case 3:///////////////////////////////////////////////////////////////////////3

do {

system("cls");

cout << " \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

cout << "| |" << endl;

cout << "|########################################################################|" << endl;

cout << "|#\t\t\tSort by price of each services #|" << endl;

cout << "|########################################################################|" << endl;

cout << "|#\t\t(1) - BarberShop #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(2) - Massage #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(3) - Nails Services #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(4) - Cosmetology #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(5) - MakeUp #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(6) - Exit #|" << endl;

cout << "|########################################################################|" << endl;

cout << "|# "; nowdate.printDate(); cout << "\t\t#|" << endl;

cout << "|########################################################################|" << endl;

cout << "|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|" << endl;

cin >> menu4;

switch (menu4) {

case 1:////////////////////////////////////////////////////////4

sortBarberPrice(barber, N\_forbarber);

printarrbarberEngl(barber, N\_forbarber);

system("pause");

break;

case 2://///////////////////////////////////////////////////////4

sortMassagePrice(massage, N\_formassage);

printarrmassageEngl(massage, N\_formassage);

system("pause");

break;

case 3://///////////////////////////////////////////////////////4

sortNailsServicePrice(nails, N\_fornails);

printarrnailsEngl(nails, N\_fornails);

system("pause");

break;

case 4://///////////////////////////////////////////////////////4

sortCosmetologyPrice(cosmetology, N\_forcosm);

printarrcosmetEngl(cosmetology, N\_forcosm);

system("pause");

break;

case 5://///////////////////////////////////////////////////////4

sortMakeUpPrice(makeup, N\_formakeup);

printarrmakeupEngl(makeup, N\_formakeup);

system("pause");

break;

case 6:

break;

default:perror("\t\tError\n"); perror("\t\tTry again later"); break;

}

} while (menu4 != 6);

break;

case 4:///////////////////////////////////////////////////////////////////////3

cout << "\t\t\t\t\t\BYE Eveline :-)\n";

system("pause");

break;

default:perror("\t\tError\n"); perror("\t\tTry again "); break;

}

} while (menu3 != 4);

break;

}

else {

perror("Passwords don`t match\nTry Again :(");

system("pause");

break;

}

break;

case 3:

cout << "Please enter your password -> ";

cin.get();

cin.getline(password, 50);

if (checkPassword(admins, N\_foradmins, user\_maria, password) == true) { // - функция проверяющая пароли

do {

system("cls");

cout << " \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

cout << "| |" << endl;

cout << "|########################################################################|" << endl;

cout << "|#\t\t\tHello Maria #|" << endl;

cout << "|########################################################################|" << endl;

cout << "|#\t\t(1) - Delete service #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(2) - Add service #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(3) - Change price of selected service #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(4) - Exit #|" << endl;

cout << "|########################################################################|" << endl;

cout << "|# "; nowdate.printDate(); cout << "\t\t#|" << endl;

cout << "|########################################################################|" << endl;

cout << "|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|" << endl;

cin >> menu3;

switch (menu3) {

case 1:///////////////////////////////////////////////////////////////////////3

do {

system("cls");

cout << " \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

cout << "| |" << endl;

cout << "|########################################################################|" << endl;

cout << "|#\t\t\tDelete Service #|" << endl;

cout << "|########################################################################|" << endl;

cout << "|#\t\t(1) - BarberShop #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(2) - Massage #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(3) - Nails Services #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(4) - Cosmetology #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(5) - MakeUp #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(6) - Exit #|" << endl;

cout << "|########################################################################|" << endl;

cout << "|# "; nowdate.printDate(); cout << "\t\t#|" << endl;

cout << "|########################################################################|" << endl;

cout << "|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|" << endl;

cin >> menu4;

switch (menu4) {

case 1:////////////////////////////////////////////////////////4

system("cls");

printarrbarberEngl(barber, N\_forbarber);

system("pause");

system("cls");

cinget();

cout << "Enter type of the service -> " << endl;

cin.getline(type\_of\_sevice, 256);

cout << "Enter price of its service -> " << endl;

cin >> someprice;

barber = deleteBarber(barber, N\_forbarber, type\_of\_sevice, someprice);

system("cls");

ArrToFilebarber(barber, N\_forbarber);

barber = ArrFromFilebarber(N\_forbarber);

printarrbarberEngl(barber, N\_forbarber);

system("pause");

break;

case 2://///////////////////////////////////////////////////////4

system("cls");

printarrmassageEngl(massage, N\_formassage);

system("pause");

system("cls");

cinget();

cout << "Enter type of the service -> " << endl;

cin.getline(type\_of\_sevice, 256);

cout << "Enter price of its service -> " << endl;

cin >> someprice;

massage = deleteMassage(massage, N\_formassage, type\_of\_sevice, someprice);

system("cls");

ArrToFilemassage(massage, N\_formassage);

massage = ArrFromFilemassage(N\_formassage);

printarrmassageEngl(massage, N\_formassage);

system("pause");

break;

case 3://///////////////////////////////////////////////////////4

system("cls");

printarrnailsEngl(nails, N\_fornails);

system("pause");

system("cls");

cinget();

cout << "Enter type of the service -> " << endl;

cin.getline(type\_of\_sevice, 256);

cout << "Enter price of its service -> " << endl;

cin >> someprice;

nails = deleteNails(nails, N\_fornails, type\_of\_sevice, someprice);

system("cls");

ArrToFilenails(nails, N\_fornails);

nails = ArrFromFilenails(N\_fornails);

printarrnailsEngl(nails, N\_fornails);

system("pause");

break;

case 4://///////////////////////////////////////////////////////4

system("cls");

printarrcosmetEngl(cosmetology, N\_forcosm);

system("pause");

system("cls");

cinget();

cout << "Enter type of the service -> " << endl;

cin.getline(type\_of\_sevice, 256);

cout << "Enter price of its service -> " << endl;

cin >> someprice;

cosmetology = deleteCosm(cosmetology, N\_forcosm, type\_of\_sevice, someprice);

system("cls");

ArrToFilecosmet(cosmetology, N\_forcosm);

cosmetology = ArrFromFilecosmet(N\_forcosm);

printarrcosmetEngl(cosmetology, N\_forcosm);

system("pause");

break;

case 5:

system("cls");

printarrmakeupEngl(makeup, N\_formakeup);

system("pause");

system("cls");

cinget();

cout << "Enter type of the service -> " << endl;

cin.getline(type\_of\_sevice, 256);

cout << "Enter price of its service -> " << endl;

cin >> someprice;

makeup = deleteMake(makeup, N\_formakeup, type\_of\_sevice, someprice);

system("cls");

ArrToFilemakeup(makeup, N\_formakeup);

makeup = ArrFromFilemakeup(N\_formakeup);

printarrmakeupEngl(makeup, N\_formakeup);

system("pause");

break;

default:perror("\t\tError\n"); perror("\t\tTry again later"); break;

}

} while (menu4 != 6);

break;

case 2:///////////////////////////////////////////////////////////////////////3

do {

system("cls");

cout << " \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

cout << "| |" << endl;

cout << "|########################################################################|" << endl;

cout << "|#\t\t\tAdd Service #|" << endl;

cout << "|########################################################################|" << endl;

cout << "|#\t\t(1) - BarberShop #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(2) - Massage #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(3) - Nails Services #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(4) - Cosmetology #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(5) - MakeUp #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(6) - Exit #|" << endl;

cout << "|########################################################################|" << endl;

cout << "|# "; nowdate.printDate(); cout << "\t\t#|" << endl;

cout << "|########################################################################|" << endl;

cout << "|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|" << endl;

cin >> menu4;

switch (menu4) {

case 1:////////////////////////////////////////////////////////4

system("cls");

printarrbarberEngl(barber, N\_forbarber);

system("pause");

barber = addBarber(barber, N\_forbarber);

ArrToFilebarber(barber, N\_forbarber);

barber = ArrFromFilebarber(N\_forbarber);

system("cls");

printarrbarberEngl(barber, N\_forbarber);

system("pause");

break;

case 2://///////////////////////////////////////////////////////4

system("cls");

printarrmassageEngl(massage, N\_formassage);

system("pause");

massage = addMassage(massage, N\_formassage);

ArrToFilemassage(massage, N\_formassage);

massage = ArrFromFilemassage(N\_formassage);

system("cls");

printarrmassageEngl(massage, N\_formassage);

system("pause");

break;

case 3://///////////////////////////////////////////////////////4

system("cls");

printarrnailsEngl(nails, N\_fornails);

system("pause");

nails = addNails(nails, N\_fornails);

ArrToFilenails(nails, N\_fornails);

nails = ArrFromFilenails(N\_fornails);

system("cls");

printarrnailsEngl(nails, N\_fornails);

system("pause");

break;

case 4://///////////////////////////////////////////////////////4

system("cls");

printarrcosmetEngl(cosmetology, N\_forcosm);

system("pause");

cosmetology = addCosm(cosmetology, N\_forcosm);

ArrToFilecosmet(cosmetology, N\_forcosm);

cosmetology = ArrFromFilecosmet(N\_forcosm);

system("cls");

printarrcosmetEngl(cosmetology, N\_forcosm);

system("pause");

break;

case 5:

system("cls");

printarrmakeupEngl(makeup, N\_formakeup);

system("pause");

makeup = addMake(makeup, N\_formakeup);

ArrToFilemakeup(makeup, N\_formakeup);

makeup = ArrFromFilemakeup(N\_formakeup);

system("cls");

printarrmakeupEngl(makeup, N\_formakeup);

system("pause");

break;

default:perror("\t\tError\n"); perror("\t\tTry again later"); break;

}

} while (menu4 != 6);

break;

case 3:///////////////////////////////////////////////////////////////////////3

do {

system("cls");

cout << " \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

cout << "| |" << endl;

cout << "|########################################################################|" << endl;

cout << "|#\t\t\tChange price of selected service #|" << endl;

cout << "|########################################################################|" << endl;

cout << "|#\t\t(1) - BarberShop #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(2) - Massage #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(3) - Nails Services #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(4) - Cosmetology #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(5) - MakeUp #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(6) - Exit #|" << endl;

cout << "|########################################################################|" << endl;

cout << "|# "; nowdate.printDate(); cout << "\t\t#|" << endl;

cout << "|########################################################################|" << endl;

cout << "|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|" << endl;

cin >> menu4;

switch (menu4) {

case 1:////////////////////////////////////////////////////////4

system("cls");

printarrbarberEngl(barber, N\_forbarber);

system("pause");

system("cls");

cinget();

cout << "Enter type of the service -> " << endl;

cin.getline(type\_of\_sevice, 256);

cout << "Enter name of the master -> " << endl;

cin.getline(name\_of\_master, 256);

changePriceBarber(barber, N\_forbarber, type\_of\_sevice, name\_of\_master);

system("pause");

system("cls");

ArrToFilebarber(barber, N\_forbarber);

barber = ArrFromFilebarber(N\_forbarber);

printarrbarberEngl(barber, N\_forbarber);

system("pause");

//функция изменения

break;

case 2://///////////////////////////////////////////////////////4

system("cls");

printarrmassageEngl(massage, N\_formassage);

system("pause");

system("cls");

cinget();

cout << "Enter type of the service -> " << endl;

cin.getline(type\_of\_sevice, 256);

cout << "Enter name of the master -> " << endl;

cin.getline(name\_of\_master, 256);

changePriceMassage(massage, N\_formassage, type\_of\_sevice, name\_of\_master);

system("pause");

system("cls");

ArrToFilemassage(massage, N\_formassage);

massage = ArrFromFilemassage(N\_formassage);

printarrmassageEngl(massage, N\_formassage);

system("pause");

break;

case 3://///////////////////////////////////////////////////////4

system("cls");

printarrnailsEngl(nails, N\_fornails);

system("pause");

system("cls");

cinget();

cout << "Enter type of the service -> " << endl;

cin.getline(type\_of\_sevice, 256);

cout << "Enter name of the master -> " << endl;

cin.getline(name\_of\_master, 256);

changePriceNails(nails, N\_fornails, type\_of\_sevice, name\_of\_master);

system("pause");

system("cls");

ArrToFilenails(nails, N\_fornails);

nails = ArrFromFilenails(N\_fornails);

printarrnailsEngl(nails, N\_fornails);

system("pause");

break;

case 4://///////////////////////////////////////////////////////4

system("cls");

printarrcosmetEngl(cosmetology, N\_forcosm);

system("pause");

system("cls");

cinget();

cout << "Enter type of the service -> " << endl;

cin.getline(type\_of\_sevice, 256);

cout << "Enter name of the master -> " << endl;

cin.getline(name\_of\_master, 256);

changePriceCosm(cosmetology, N\_forcosm, type\_of\_sevice, name\_of\_master);

system("pause");

system("cls");

ArrToFilecosmet(cosmetology, N\_forcosm);

cosmetology = ArrFromFilecosmet(N\_forcosm);

printarrcosmetEngl(cosmetology, N\_forcosm);

system("pause");

break;

case 5://///////////////////////////////////////////////////////4

system("cls");

printarrmakeupEngl(makeup, N\_formakeup);

system("pause");

system("cls");

cinget();

cout << "Enter type of the service -> " << endl;

cin.getline(type\_of\_sevice, 256);

cout << "Enter name of the master -> " << endl;

cin.getline(name\_of\_master, 256);

changePriceMake(makeup, N\_formakeup, type\_of\_sevice, name\_of\_master);

system("pause");

system("cls");

ArrToFilemakeup(makeup, N\_formakeup);

makeup = ArrFromFilemakeup(N\_formakeup);

printarrmakeupEngl(makeup, N\_formakeup);

system("pause");

break;

default:perror("\t\tError\n"); perror("\t\tTry again later"); break;

}

} while (menu4 != 6);

break;

break;

case 4:///////////////////////////////////////////////////////////////////////3

cout << "\t\t\t\t\t\BYE Maria :-)";

break;

default:perror("\t\tError\n"); perror("\t\tTry again "); break;

}

} while (menu3 != 4);

break;

}

else {

perror("Passwords don`t match\nTry Again :(");

system("pause");

break;

}

break;

case 4:

break;

default:

perror("\t\t\tFail//Ошибка");

break;

}

} while (menu2 != 4);

break;

case 2://///////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////1/

do {

system("cls");

cout << " \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

cout << "| |" << endl;

cout << "|########################################################################|" << endl;

cout << "|#\t\t\tПожалуйста выберите ваш акканут #|" << endl;

cout << "|########################################################################|" << endl;

cout << "|#\t\t(1) - Администратор Татьяна #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(2) - Администратор Эвелин #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(3) - Менеджер Мария #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(4) - Выход #|" << endl;

cout << "|########################################################################|" << endl;

cout << "|# "; nowdate.printDate(); cout << "\t\t#|" << endl;

cout << "|########################################################################|" << endl;

cout << "|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|" << endl;

cin >> menu2;

switch (menu2) {

case 1:////////////////////////////////////////////////////////////////////////////////////////////////////////////////////2

cout << "Пожалуйста введите пароль -> ";

cin.get();

cin.getline(password, 50);

if (checkPassword(admins, N\_foradmins, user\_tatiana, password) == true) { // - функция проверяющая пароли

do {

system("cls");

cout << " \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

cout << "| |" << endl;

cout << "|########################################################################|" << endl;

cout << "|#\t\t\tЗдравствуйте Татьяна #|" << endl;

cout << "|########################################################################|" << endl;

cout << "|#\t(1) - Принять заказ #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t(2) - Распечатать прайс-лист #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t(3) - Сортировка по цене #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t(4) - Выход #|" << endl;

cout << "|########################################################################|" << endl;

cout << "|# "; nowdate.printDate(); cout << "\t\t#|" << endl;

cout << "|########################################################################|" << endl;

cout << "|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|" << endl;

cin >> menu3;

switch (menu3) {

case 1:///////////////////////////////////////////////////////////////////////3

do {

system("cls");

cout << " \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

cout << "| |" << endl;

cout << "|########################################################################|" << endl;

cout << "|#\t\t\tПринять заказ #|" << endl;

cout << "|########################################################################|" << endl;

cout << "|#\t\t(1) - Парикмахерская #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(2) - Массаж #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(3) - Маникюр-Педикюр #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(4) - Косметология #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(5) - Визаж #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(6) - Рспечатать чек #|" << endl;

cout << "|########################################################################|" << endl;

cout << "|# "; nowdate.printDate(); cout << "\t\t#|" << endl;

cout << "|########################################################################|" << endl;

cout << "|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|" << endl;

cin >> menu4;

switch (menu4) {

case 1:////////////////////////////////////////////////////////4

bar++;

check = addCheckRus(check, N);

AddToFileCheck(check, N);

/\*check = AddFromFileCheck(N);

printCheckEngl(check, N);

system("pause");\*/

break;

case 2://///////////////////////////////////////////////////////4

mas++;

check = addCheckRus(check, N);

AddToFileCheck(check, N);

break;

case 3://///////////////////////////////////////////////////////4

nail++;

check = addCheckRus(check, N);

AddToFileCheck(check, N);

break;

case 4://///////////////////////////////////////////////////////4

cosm++;

check = addCheckRus(check, N);

AddToFileCheck(check, N);

break;

case 5://///////////////////////////////////////////////////////4

mak++;

check = addCheckRus(check, N);

AddToFileCheck(check, N);

break;

case 6://///////////////////////////////////////////////////////4

system("cls");

if (bar != 0 || mas != 0 || nail != 0 || cosm != 0 || mak != 0) {

totalPrice(check, N, totalprice);

cout.width();

cout << "#################################################################" << endl;

cout << "# ! ЧЕК ! #" << endl;

cout << "#################################################################" << endl;

printCheckEngl(check, N);

cout << "#################################################################" << endl; //

cout.width(); cout << "# ИТОГОВАЯ ЦЕНА : " << totalprice << " grn. #" << endl;

nowdate.printDateforChek();

cout << "#################################################################" << endl;

}

else

perror("\t\tНедостаточно информации");

system("pause");

break;

default:perror("\t\tОшибка\n"); perror("\t\tПопробуйте снова"); system("pause"); break;

}

} while (menu4 != 6);

break;

break;

case 2:///////////////////////////////////////////////////////////////////////3

do {

system("cls");

cout << " \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

cout << "| |" << endl;

cout << "|########################################################################|" << endl;

cout << "|#\t\t\tРаспечатать прайс-лист #|" << endl;

cout << "|########################################################################|" << endl;

cout << "|#\t\t(1) - Парикмахерская #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(2) - Массаж #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(3) - Маникюр-Педикюр #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(4) - Косметология #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(5) - Визаж #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(6) - Выход #|" << endl;

cout << "|########################################################################|" << endl;

cout << "|# "; nowdate.printDate(); cout << "\t\t#|" << endl;

cout << "|########################################################################|" << endl;

cout << "|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|" << endl;

cin >> menu4;

switch (menu4) {

case 1:////////////////////////////////////////////////////////4

printarrbarberRus(barber, N\_forbarber);

system("pause");

break;

case 2://///////////////////////////////////////////////////////4

printarrmassageRus(massage, N\_formassage);

system("pause");

break;

case 3://///////////////////////////////////////////////////////4

printarrnailsRus(nails, N\_fornails);

system("pause");

break;

case 4://///////////////////////////////////////////////////////4

printarrcosmetRus(cosmetology, N\_forcosm);

system("pause");

break;

case 5:

printarrmakeupRus(makeup, N\_formakeup);

system("pause");

break;

case 6:

break;

default:perror("\t\tОшибка\n"); perror("\t\tПопробуйте снова"); break;

}

} while (menu4 != 6);

break;

case 3:///////////////////////////////////////////////////////////////////////3

do {

system("cls");

cout << " \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

cout << "| |" << endl;

cout << "|########################################################################|" << endl;

cout << "|#\t\t\tСортировка по цене #|" << endl;

cout << "|########################################################################|" << endl;

cout << "|#\t\t(1) - Парикмахерская #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(2) - Массаж #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(3) - Маникюр-Педикюр #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(4) - Косметология #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(5) - Визаж #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(6) - Выход #|" << endl;

cout << "|########################################################################|" << endl;

cout << "|# "; nowdate.printDate(); cout << "\t\t#|" << endl;

cout << "|########################################################################|" << endl;

cout << "|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|" << endl;

cin >> menu4;

switch (menu4) {

case 1:////////////////////////////////////////////////////////4

sortBarberPrice(barber, N\_forbarber);

printarrbarberRus(barber, N\_forbarber);

system("pause");

break;

case 2://///////////////////////////////////////////////////////4

sortMassagePrice(massage, N\_formassage);

printarrmassageRus(massage, N\_formassage);

system("pause");

break;

case 3://///////////////////////////////////////////////////////4

sortNailsServicePrice(nails, N\_fornails);

printarrnailsRus(nails, N\_fornails);

system("pause");

break;

case 4://///////////////////////////////////////////////////////4

sortCosmetologyPrice(cosmetology, N\_forcosm);

printarrcosmetRus(cosmetology, N\_forcosm);

system("pause");

break;

case 5://///////////////////////////////////////////////////////4

sortMakeUpPrice(makeup, N\_formakeup);

printarrmakeupRus(makeup, N\_formakeup);

system("pause");

break;

case 6:

break;

default:perror("\t\tОшибка\n"); perror("\t\tПопробуйте снова"); break;

}

} while (menu4 != 6);

break;

case 4:///////////////////////////////////////////////////////////////////////3

cout << "\t\t\tДо свидания, Татьяна:-)\n";

system("pause");

break;

default:perror("\t\tОшибка\n"); perror("\t\tПопробуйте снова "); break;

}

} while (menu3 != 4);

break;

}

else {

perror("Пароли не совпадают\nПопробуйте снова:(");

system("pause");

break;

}

break;

case 2:

cout << "Пожалуйста введите пароль -> -> ";

cin.get();

cin.getline(password, 50);

if (checkPassword(admins, N\_foradmins, user\_eveline, password) == true) { // - функция проверяющая пароли

do {

system("cls");

cout << " \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

cout << "| |" << endl;

cout << "|########################################################################|" << endl;

cout << "|#\t\t\tЗдравствуйте Эвелин #|" << endl;

cout << "|########################################################################|" << endl;

cout << "|#\t(1) - Принять заказ #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t(2) - Распечатать прайс-лист #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t(3) - Сортировка по цене #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t(4) - Выход #|" << endl;

cout << "|########################################################################|" << endl;

cout << "|# "; nowdate.printDate(); cout << "\t\t#|" << endl;

cout << "|########################################################################|" << endl;

cout << "|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|" << endl;

cin >> menu3;

switch (menu3) {

case 1:///////////////////////////////////////////////////////////////////////3 //

do {

system("cls");

cout << " \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

cout << "| |" << endl;

cout << "|########################################################################|" << endl;

cout << "|#\t\t\tПринять заказ #|" << endl;

cout << "|########################################################################|" << endl;

cout << "|#\t\t(1) - Парикмахерская #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(2) - Массаж #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(3) - Маникюр-Педикюр #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(4) - Косметология #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(5) - Визаж #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(6) - Рспечатать чек #|" << endl;

cout << "|########################################################################|" << endl;

cout << "|# "; nowdate.printDate(); cout << "\t\t#|" << endl;

cout << "|########################################################################|" << endl;

cout << "|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|" << endl;

cin >> menu4;

switch (menu4) {

case 1:////////////////////////////////////////////////////////4

bar++;

check = addCheckRus(check, N);

AddToFileCheck(check, N);

/\*check = AddFromFileCheck(N);

printCheckEngl(check, N);

system("pause");\*/

break;

case 2://///////////////////////////////////////////////////////4

mas++;

check = addCheckRus(check, N);

AddToFileCheck(check, N);

break;

case 3://///////////////////////////////////////////////////////4

nail++;

check = addCheckRus(check, N);

AddToFileCheck(check, N);

break;

case 4://///////////////////////////////////////////////////////4

cosm++;

check = addCheckRus(check, N);

AddToFileCheck(check, N);

break;

case 5://///////////////////////////////////////////////////////4

mak++;

check = addCheckRus(check, N);

AddToFileCheck(check, N);

break;

case 6://///////////////////////////////////////////////////////4

system("cls");

if (bar != 0 || mas != 0 || nail != 0 || cosm != 0 || mak != 0) {

totalPrice(check, N, totalprice);

cout.width();

cout << "#################################################################" << endl;

cout << "# ! ЧЕК ! #" << endl;

cout << "#################################################################" << endl;

printCheckEngl(check, N);

cout << "#################################################################" << endl; //

cout.width(); cout << "# ИТОГОВАЯ ЦЕНА : " << totalprice << " grn. #" << endl;

nowdate.printDateforChek();

cout << "#################################################################" << endl;

}

else

perror("\t\tНедостаточно информации");

system("pause");

break;

default:perror("\t\tОшибка\n"); perror("\t\tПопробуйте снова"); system("pause"); break;

}

} while (menu4 != 6);

break;

break;

case 2:///////////////////////////////////////////////////////////////////////3

do {

system("cls");

cout << " \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

cout << "| |" << endl;

cout << "|########################################################################|" << endl;

cout << "|#\t\t\tРаспечатать прайс-лист #|" << endl;

cout << "|########################################################################|" << endl;

cout << "|#\t\t(1) - Парикмахерская #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(2) - Массаж #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(3) - Маникюр-Педикюр #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(4) - Косметология #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(5) - Визаж #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(6) - Выход #|" << endl;

cout << "|########################################################################|" << endl;

cout << "|# "; nowdate.printDate(); cout << "\t\t#|" << endl;

cout << "|########################################################################|" << endl;

cout << "|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|" << endl;

cin >> menu4;

switch (menu4) {

case 1:////////////////////////////////////////////////////////4

printarrbarberRus(barber, N\_forbarber);

system("pause");

break;

case 2://///////////////////////////////////////////////////////4

printarrmassageRus(massage, N\_formassage);

system("pause");

break;

case 3://///////////////////////////////////////////////////////4

printarrnailsRus(nails, N\_fornails);

system("pause");

break;

case 4://///////////////////////////////////////////////////////4

printarrcosmetRus(cosmetology, N\_forcosm);

system("pause");

break;

case 5:

printarrmakeupRus(makeup, N\_formakeup);

system("pause");

break;

case 6:

break;

default:perror("\t\tОшибка\n"); perror("\t\tПопробуйте снова"); break;

}

} while (menu4 != 6);

break;

case 3:///////////////////////////////////////////////////////////////////////3

do {

system("cls");

cout << " \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

cout << "| |" << endl;

cout << "|########################################################################|" << endl;

cout << "|#\t\t\tСортировка по цене #|" << endl;

cout << "|########################################################################|" << endl;

cout << "|#\t\t(1) - Парикмахерская #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(2) - Массаж #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(3) - Маникюр-Педикюр #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(4) - Косметология #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(5) - Визаж #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(6) - Выход #|" << endl;

cout << "|########################################################################|" << endl;

cout << "|# "; nowdate.printDate(); cout << "\t\t#|" << endl;

cout << "|########################################################################|" << endl;

cout << "|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|" << endl;

cin >> menu4;

switch (menu4) {

case 1:////////////////////////////////////////////////////////4

sortBarberPrice(barber, N\_forbarber);

printarrbarberRus(barber, N\_forbarber);

system("pause");

break;

case 2://///////////////////////////////////////////////////////4

sortMassagePrice(massage, N\_formassage);

printarrmassageRus(massage, N\_formassage);

system("pause");

break;

case 3://///////////////////////////////////////////////////////4

sortNailsServicePrice(nails, N\_fornails);

printarrnailsRus(nails, N\_fornails);

system("pause");

break;

case 4://///////////////////////////////////////////////////////4

sortCosmetologyPrice(cosmetology, N\_forcosm);

printarrcosmetRus(cosmetology, N\_forcosm);

system("pause");

break;

case 5://///////////////////////////////////////////////////////4

sortMakeUpPrice(makeup, N\_formakeup);

printarrmakeupRus(makeup, N\_formakeup);

system("pause");

break;

case 6:

break;

default:perror("\t\tОшибка\n"); perror("\t\tПопробуйте снова"); break;

}

} while (menu4 != 6);

break;

case 4:///////////////////////////////////////////////////////////////////////3

cout << "\t\t\tДо свидания, Эвелин:-)\n";

system("pause");

break;

default:perror("\t\tОшибка\n"); perror("\t\tПопробуйте снова "); break;

}

} while (menu3 != 4);

break;

}

else {

perror("Пароли не совпадают\nПопробуйте снова :(");

system("pause");

break;

}

break;

case 3:

cout << "Пожалуйста введите пароль -> ";

cin.get();

cin.getline(password, 50);

if (checkPassword(admins, N\_foradmins, user\_maria, password) == true) { // - функция проверяющая пароли

do {

system("cls");

cout << " \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

cout << "| |" << endl;

cout << "|########################################################################|" << endl;

cout << "|#\t\t\tЗдравствуйте Мария #|" << endl;

cout << "|########################################################################|" << endl;

cout << "|#\t\t(1) - Удалить услугу #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(2) - Добавить услугу #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(3) - Изменить цену за услугу #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(4) - Выход #|" << endl;

cout << "|########################################################################|" << endl;

cout << "|# "; nowdate.printDate(); cout << "\t\t#|" << endl;

cout << "|########################################################################|" << endl;

cout << "|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|" << endl;

cin >> menu3;

switch (menu3) {

case 1:///////////////////////////////////////////////////////////////////////3

do {

system("cls");

cout << " \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

cout << "| |" << endl;

cout << "|########################################################################|" << endl;

cout << "|#\t\t\tУдалить услугу #|" << endl;

cout << "|########################################################################|" << endl;

cout << "|#\t\t(1) - Парикмахерская #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(2) - Массаж #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(3) - Маникюр-Педикюр #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(4) - Косметология #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(5) - Визаж #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(6) - Выход #|" << endl;

cout << "|########################################################################|" << endl;

cout << "|# "; nowdate.printDate(); cout << "\t\t#|" << endl;

cout << "|########################################################################|" << endl;

cout << "|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|" << endl;

cin >> menu4;

switch (menu4) {

case 1:////////////////////////////////////////////////////////4

system("cls");

printarrbarberRus(barber, N\_forbarber);

system("pause");

system("cls");

cinget();

cout << "Введите тип услуги -> " << endl;

cin.getline(type\_of\_sevice, 256);

cout << "Введите цену за данную услугу -> " << endl;

cin >> someprice;

barber = deleteBarber(barber, N\_forbarber, type\_of\_sevice, someprice);

system("cls");

ArrToFilebarber(barber, N\_forbarber);

barber = ArrFromFilebarber(N\_forbarber);

printarrbarberRus(barber, N\_forbarber);

system("pause");

break;

case 2://///////////////////////////////////////////////////////4

system("cls");

printarrmassageRus(massage, N\_formassage);

system("pause");

system("cls");

cinget();

cout << "Введите тип услуги -> " << endl;

cin.getline(type\_of\_sevice, 256);

cout << "Введите цену за данную услугу -> " << endl;

cin >> someprice;

massage = deleteMassage(massage, N\_formassage, type\_of\_sevice, someprice);

system("cls");

ArrToFilemassage(massage, N\_formassage);

massage = ArrFromFilemassage(N\_formassage);

printarrmassageRus(massage, N\_formassage);

system("pause");

break;

case 3://///////////////////////////////////////////////////////4

system("cls");

printarrnailsRus(nails, N\_fornails);

system("pause");

system("cls");

cinget();

cout << "Введите тип услуги -> " << endl;

cin.getline(type\_of\_sevice, 256);

cout << "Введите цену за данную услугу -> " << endl;

cin >> someprice;

nails = deleteNails(nails, N\_fornails, type\_of\_sevice, someprice);

system("cls");

ArrToFilenails(nails, N\_fornails);

nails = ArrFromFilenails(N\_fornails);

printarrnailsRus(nails, N\_fornails);

system("pause");

break;

case 4://///////////////////////////////////////////////////////4

system("cls");

printarrcosmetRus(cosmetology, N\_forcosm);

system("pause");

system("cls");

cinget();

cout << "Введите тип услуги -> " << endl;

cin.getline(type\_of\_sevice, 256);

cout << "Введите цену за данную услугу -> " << endl;

cin >> someprice;

cosmetology = deleteCosm(cosmetology, N\_forcosm, type\_of\_sevice, someprice);

system("cls");

ArrToFilecosmet(cosmetology, N\_forcosm);

cosmetology = ArrFromFilecosmet(N\_forcosm);

printarrcosmetRus(cosmetology, N\_forcosm);

system("pause");

break;

case 5:

system("cls");

printarrmakeupRus(makeup, N\_formakeup);

system("pause");

system("cls");

cinget();

cout << "Введите тип услуги -> " << endl;

cin.getline(type\_of\_sevice, 256);

cout << "Введите цену за данную услугу -> " << endl;

cin >> someprice;

makeup = deleteMake(makeup, N\_formakeup, type\_of\_sevice, someprice);

system("cls");

ArrToFilemakeup(makeup, N\_formakeup);

makeup = ArrFromFilemakeup(N\_formakeup);

printarrmakeupRus(makeup, N\_formakeup);

system("pause");

break;

default:perror("\t\tОшибка\n"); perror("\t\tПопробуйте снова"); break;

}

} while (menu4 != 6);

break;

case 2:///////////////////////////////////////////////////////////////////////3

do {

system("cls");

cout << " \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

cout << "| |" << endl;

cout << "|########################################################################|" << endl;

cout << "|#\t\t\tДобавить услугу #|" << endl;

cout << "|########################################################################|" << endl;/////////////////////////////////////////////////////////////////////////

cout << "|#\t\t(1) - Парикмахерская #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(2) - Массаж #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(3) - Маникюр-Педикюр #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(4) - Косметология #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(5) - Визаж #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(6) - Выход #|" << endl;

cout << "|########################################################################|" << endl;

cout << "|# "; nowdate.printDate(); cout << "\t\t#|" << endl;

cout << "|########################################################################|" << endl;

cout << "|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|" << endl;

cin >> menu4;

switch (menu4) {

case 1:////////////////////////////////////////////////////////4

system("cls");

printarrbarberRus(barber, N\_forbarber);

system("pause");

barber = addBarber(barber, N\_forbarber);

ArrToFilebarber(barber, N\_forbarber);

barber = ArrFromFilebarber(N\_forbarber);

system("cls");

printarrbarberRus(barber, N\_forbarber);

system("pause");

break;

case 2://///////////////////////////////////////////////////////4

system("cls");

printarrmassageRus(massage, N\_formassage);

system("pause");

massage = addMassage(massage, N\_formassage);

ArrToFilemassage(massage, N\_formassage);

massage = ArrFromFilemassage(N\_formassage);

system("cls");

printarrmassageRus(massage, N\_formassage);

system("pause");

break;

case 3://///////////////////////////////////////////////////////4

system("cls");

printarrnailsRus(nails, N\_fornails);

system("pause");

nails = addNails(nails, N\_fornails);

ArrToFilenails(nails, N\_fornails);

nails = ArrFromFilenails(N\_fornails);

system("cls");

printarrnailsRus(nails, N\_fornails);

system("pause");

break;

case 4://///////////////////////////////////////////////////////4

system("cls");

printarrcosmetRus(cosmetology, N\_forcosm);

system("pause");

cosmetology = addCosm(cosmetology, N\_forcosm);

ArrToFilecosmet(cosmetology, N\_forcosm);

cosmetology = ArrFromFilecosmet(N\_forcosm);

system("cls");

printarrcosmetRus(cosmetology, N\_forcosm);

system("pause");

break;

case 5:

system("cls");

printarrmakeupRus(makeup, N\_formakeup);

system("pause");

makeup = addMake(makeup, N\_formakeup);

ArrToFilemakeup(makeup, N\_formakeup);

makeup = ArrFromFilemakeup(N\_formakeup);

system("cls");

printarrmakeupRus(makeup, N\_formakeup);

system("pause");

break;

default:perror("\t\tОшибка\n"); perror("\t\tПопробуйте снова"); break;

}

} while (menu4 != 6);

break;

case 3:///////////////////////////////////////////////////////////////////////3

do {

system("cls");

cout << " \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

cout << "| |" << endl;

cout << "|########################################################################|" << endl;

cout << "|#\t\t\tИзменить цену за услугу #|" << endl;

cout << "|########################################################################|" << endl;

cout << "|#\t\t(1) - Парикмахерская #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(2) - Массаж #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(3) - Маникюр-Педикюр #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(4) - Косметология #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(5) - Визаж #|" << endl;

cout << "|#----------------------------------------------------------------------#|" << endl;

cout << "|#\t\t(6) - Выход #|" << endl;

cout << "|########################################################################|" << endl;

cout << "|# "; nowdate.printDate(); cout << "\t\t#|" << endl;

cout << "|########################################################################|" << endl;

cout << "|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|" << endl;

cin >> menu4;

switch (menu4) {

case 1:////////////////////////////////////////////////////////4

system("cls");

printarrbarberRus(barber, N\_forbarber);

system("pause");

system("cls");

cinget();

cout << "Введите тип услуги -> " << endl;

cin.getline(type\_of\_sevice, 256);

cout << "Введите им мастера -> " << endl;

cin.getline(name\_of\_master, 256);

changePriceBarber(barber, N\_forbarber, type\_of\_sevice, name\_of\_master);

system("pause");

system("cls");

ArrToFilebarber(barber, N\_forbarber);

barber = ArrFromFilebarber(N\_forbarber);

printarrbarberRus(barber, N\_forbarber);

system("pause");

//функция изменения

break;

case 2://///////////////////////////////////////////////////////4

system("cls");

printarrmassageRus(massage, N\_formassage);

system("pause");

system("cls");

cinget();

cout << "Введите тип услуги -> " << endl;

cin.getline(type\_of\_sevice, 256);

cout << "Введите им мастера -> " << endl;

cin.getline(name\_of\_master, 256);

changePriceMassage(massage, N\_formassage, type\_of\_sevice, name\_of\_master);

system("pause");

system("cls");

ArrToFilemassage(massage, N\_formassage);

massage = ArrFromFilemassage(N\_formassage);

printarrmassageRus(massage, N\_formassage);

system("pause");

break;

case 3://///////////////////////////////////////////////////////4

system("cls");

printarrnailsRus(nails, N\_fornails);

system("pause");

system("cls");

cinget();

cout << "Введите тип услуги -> " << endl;

cin.getline(type\_of\_sevice, 256);

cout << "Введите им мастера -> " << endl;

cin.getline(name\_of\_master, 256);

changePriceNails(nails, N\_fornails, type\_of\_sevice, name\_of\_master);

system("pause");

system("cls");

ArrToFilenails(nails, N\_fornails);

nails = ArrFromFilenails(N\_fornails);

printarrnailsRus(nails, N\_fornails);

system("pause");

break;

case 4://///////////////////////////////////////////////////////4

system("cls");

printarrcosmetRus(cosmetology, N\_forcosm);

system("pause");

system("cls");

cinget();

cout << "Введите тип услуги -> " << endl;

cin.getline(type\_of\_sevice, 256);

cout << "Введите им мастера -> " << endl;

cin.getline(name\_of\_master, 256);

changePriceCosm(cosmetology, N\_forcosm, type\_of\_sevice, name\_of\_master);

system("pause");

system("cls");

ArrToFilecosmet(cosmetology, N\_forcosm);

cosmetology = ArrFromFilecosmet(N\_forcosm);

printarrcosmetRus(cosmetology, N\_forcosm);

system("pause");

break;

case 5://///////////////////////////////////////////////////////4

system("cls");

printarrmakeupRus(makeup, N\_formakeup);

system("pause");

system("cls");

cinget();

cout << "Введите тип услуги -> " << endl;

cin.getline(type\_of\_sevice, 256);

cout << "Введите им мастера -> " << endl;

cin.getline(name\_of\_master, 256);

changePriceMake(makeup, N\_formakeup, type\_of\_sevice, name\_of\_master);

system("pause");

system("cls");

ArrToFilemakeup(makeup, N\_formakeup);

makeup = ArrFromFilemakeup(N\_formakeup);

printarrmakeupRus(makeup, N\_formakeup);

system("pause");

break;

default:perror("\t\tОшибка\n"); perror("\t\tПопробуйте снова"); break;

}

} while (menu4 != 6);

break;

break;

case 4:///////////////////////////////////////////////////////////////////////3

cout << "\t\t\tДо свидания, Мария :-)";

break;

default:perror("\t\tОшибка\n"); perror("\t\tПопробуйте снова"); break;

}

} while (menu3 != 4);

break;

}

else {

perror("Пароли не совпадают\nПопробуйте снова :(");

system("pause");

break;

}

break;

default:

perror("\t\t\tFail//Ошибка");

break;

}

} while (menu2 != 4);

break;

case 3:

system("cls");

cout << "\n\n\n\n\t\t\tВсего доброго/Good luck\n\n\n" << endl;

break;

default:

perror("\t\t\tFail//Ошибка");

break;

}

} while (menu1 != 3);

system("pause");

return 0;

}

Services.h

#pragma once

#include"MyString.h"

#pragma once

#include"Administrators.h"

struct BarberShop {

MyString name\_master;

MyString type;

float price;

void fillbarber() {

char tmp[256] = "";

cout << "Введите имя мастера -> ";

cin.getline(tmp, 256);

name\_master.setMyString(tmp);

cout << "Введите название услуги -> ";

cin.getline(tmp, 256);

type.setMyString(tmp);

cout << "Введите цену за данную услугу -> ";

cin >> price;

}

void fillbarber2() {

char tmp[256] = "";

cin.get();

cout << "Введите имя мастера -> ";

cin.getline(tmp, 256);

name\_master.setMyString(tmp);

cout << "Введите название услуги -> ";

cin.getline(tmp, 256);

type.setMyString(tmp);

cout << "Введите цену за данную услугу -> ";

cin >> price;

}

void printbarberRus() {

cout << "Имя мастера -> " << name\_master.str << " ;" << endl;

cout << "-------------------------------------" << endl;

cout << "Название услуги -> " << type.str << " ;" << endl;

cout << "-------------------------------------" << endl;

cout << "Цена за данную услугу -> " << price << " ." << endl;

}

void printbarberEngl() {

cout << "Specialist name -> " << name\_master.str << " ;" << endl;

cout << "-------------------------------------" << endl;

cout << "Name of the service -> " << type.str << " ;" << endl;

cout << "-------------------------------------" << endl;

cout << "Price for this service -> " << price << " ." << endl;

}

void toFilebarber(FILE \*fp) {

fwrite(&name\_master.length, sizeof(int), 1, fp);

fwrite(name\_master.str, sizeof(char), name\_master.length, fp);

fwrite(&type.length, sizeof(int), 1, fp);

fwrite(type.str, sizeof(char), type.length, fp);

fwrite(&price, sizeof(float), 1, fp);

}

void fromFilebarber(FILE\*fp) {

fread(&name\_master.length, sizeof(int), 1, fp);

name\_master.str = new char[name\_master.length + 1];

fread(name\_master.str, sizeof(char), name\_master.length, fp);

name\_master.str[name\_master.length] = '\0';

fread(&type.length, sizeof(int), 1, fp);

type.str = new char[type.length + 1];

fread(type.str, sizeof(char), type.length, fp);

type.str[type.length] = '\0';

fread(&price, sizeof(float), 1, fp);

}

void fromFilebarbercheck(FILE\*fp) {

fread(&name\_master.length, sizeof(int), 1, fp);

name\_master.str = new char[name\_master.length + 1];

fread(name\_master.str, sizeof(char), name\_master.length, fp);

name\_master.str[name\_master.length] = '\0';

fread(&type.length, sizeof(int), 1, fp);

type.str = new char[type.length + 1];

fread(type.str, sizeof(char), type.length, fp);

type.str[type.length] = '\0';

}

};

void fillarrbarber(BarberShop\*b, int N);

void printarrbarberRus(BarberShop\*b, int N);

void printarrbarberEngl(BarberShop\*b, int N);

void ArrToFilebarber(BarberShop\*b, int N);

BarberShop \*ArrFromFilebarber(int&N);

void sumTotalBarberRus(BarberShop \*b, int N, char \*name, char \*type, float &total\_price);

void sumTotalBarberEngl(BarberShop \*b, int N, char \*name, char \*type, float &total\_price);

void sortBarberPrice(BarberShop \*b, int N);

void changePriceBarber(BarberShop \*b, int N,char \*type,char \*name\_of\_master);

BarberShop \*deleteBarber(BarberShop \*b, int &N, char \*type, float price);

BarberShop\*addBarber(BarberShop \*b, int &N);

struct Massage {

MyString name\_master;

MyString type;

int time = 0;

float price = 0;

void fillmassage() {

char tmp[256] = "";

cout << "Введите имя мастера -> ";

cin.getline(tmp, 256);

name\_master.setMyString(tmp);

cout << "Введите название услуги -> ";

cin.getline(tmp, 256);

type.setMyString(tmp);

cout << "Введите время данной услуги -> ";

cin >> time;

cout << "Введите цену данной услуги -> ";

cin >> price;

}

void fillmassage2() {

char tmp[256] = "";

cin.get();

cout << "Введите имя мастера -> ";

cin.getline(tmp, 256);

name\_master.setMyString(tmp);

cout << "Введите название услуги -> ";

cin.getline(tmp, 256);

type.setMyString(tmp);

cout << "Введите время данной услуги -> ";

cin >> time;

cout << "Введите цену данной услуги -> ";

cin >> price;

}

void printmassageRus() {

cout << "Имя мастера -> " << name\_master.str << " ;" << endl;

cout << "-------------------------------------" << endl;

cout << "Название услуги -> " << type.str << " ;" << endl;

cout << "-------------------------------------" << endl;

cout << "Время данной услуги -> " << time << " ." << endl;

cout << "-------------------------------------" << endl;

cout << "Цена за данную услугу -> " << price << " ." << endl;

}

void printmassageEngl() {

cout << "Specialist name -> " << name\_master.str << " ;" << endl;

cout << "-------------------------------------" << endl;

cout << "Name of the service -> " << type.str << " ;" << endl;

cout << "-------------------------------------" << endl;

cout << "Time of this service -> " << time << " ." << endl;

cout << "-------------------------------------" << endl;

cout << "Price for this service -> " << price << " ." << endl;

}

void toFilemassage(FILE \*fp) {

fwrite(&name\_master.length, sizeof(int), 1, fp);

fwrite(name\_master.str, sizeof(char), name\_master.length, fp);

fwrite(&type.length, sizeof(int), 1, fp);

fwrite(type.str, sizeof(char), type.length, fp);

fwrite(&time, sizeof(int), 1, fp);

fwrite(&price, sizeof(float), 1, fp);

}

void fromFilemassage(FILE\*fp) {

fread(&name\_master.length, sizeof(int), 1, fp);

name\_master.str = new char[name\_master.length + 1];

fread(name\_master.str, sizeof(char), name\_master.length, fp);

name\_master.str[name\_master.length] = '\0';

fread(&type.length, sizeof(int), 1, fp);

type.str = new char[type.length + 1];

fread(type.str, sizeof(char), type.length, fp);

type.str[type.length] = '\0';

fread(&time, sizeof(int), 1, fp);

fread(&price, sizeof(float), 1, fp);

}

};

void fillarrmassage(Massage\*b, int N);

void printarrmassageRus(Massage\*b, int N);

void printarrmassageEngl(Massage\*b, int N);

void ArrToFilemassage(Massage\*b, int N);

Massage \*ArrFromFilemassage(int&N);

void sumTotalMassageRus(Massage \*b, int N, char \*name, char \*type, float &total\_price);

void sumTotalMassageEngl(Massage \*b, int N, char \*name, char \*type, float &total\_price);

void sortMassagePrice(Massage \*b, int N);

void changePriceMassage(Massage \*b, int N, char \*type, char \*name\_of\_master);

Massage \*deleteMassage(Massage \*b, int &N, char \*type, float price);

Massage\*addMassage(Massage \*b, int &N);

struct NailsService {

MyString name\_master;

MyString type;

float price;

void fillnails() {

char tmp[256] = "";

cout << "Введите имя мастера -> ";

cin.getline(tmp, 256);

name\_master.setMyString(tmp);

cout << "Введите название услуги -> ";

cin.getline(tmp, 256);

type.setMyString(tmp);

cout << "Введите цену за данную услугу -> ";

cin >> price;

}

void fillnails2() {

char tmp[256] = "";

cin.get();

cout << "Введите имя мастера -> ";

cin.getline(tmp, 256);

name\_master.setMyString(tmp);

cout << "Введите название услуги -> ";

cin.getline(tmp, 256);

type.setMyString(tmp);

cout << "Введите цену за данную услугу -> ";

cin >> price;

}

void printnailsRus() {

cout << "Имя мастера -> " << name\_master.str << " ;" << endl;

cout << "-------------------------------------" << endl;

cout << "Название услуги -> " << type.str << " ;" << endl;

cout << "-------------------------------------" << endl;

cout << "Цена за данную услугу -> " << price << " ." << endl;

}

void printnailsEngl() {

cout << "Specialist name -> " << name\_master.str << " ;" << endl;

cout << "-------------------------------------" << endl;

cout << "Name of the service -> " << type.str << " ;" << endl;

cout << "-------------------------------------" << endl;

cout << "Price for this service -> " << price << " ." << endl;

}

void toFilenails(FILE \*fp) {

fwrite(&name\_master.length, sizeof(int), 1, fp);

fwrite(name\_master.str, sizeof(char), name\_master.length, fp);

fwrite(&type.length, sizeof(int), 1, fp);

fwrite(type.str, sizeof(char), type.length, fp);

fwrite(&price, sizeof(float), 1, fp);

}

void fromFilenails(FILE\*fp) {

fread(&name\_master.length, sizeof(int), 1, fp);

name\_master.str = new char[name\_master.length + 1];

fread(name\_master.str, sizeof(char), name\_master.length, fp);

name\_master.str[name\_master.length] = '\0';

fread(&type.length, sizeof(int), 1, fp);

type.str = new char[type.length + 1];

fread(type.str, sizeof(char), type.length, fp);

type.str[type.length] = '\0';

fread(&price, sizeof(float), 1, fp);

}

};

void fillarrnails(NailsService\*b, int N);

void printarrnailsRus(NailsService\*b, int N);

void printarrnailsEngl(NailsService\*b, int N);

void ArrToFilenails(NailsService\*b, int N);

NailsService \*ArrFromFilenails(int&N);

void sumTotalNailsServiceRus(NailsService \*b, int N, char \*name, char \*type, float &total\_price);

void sumTotalNailsServiceEngl(NailsService \*b, int N, char \*name, char \*type, float &total\_price);

void sortNailsServicePrice(NailsService \*b, int N);

void changePriceNails(NailsService \*b, int N, char \*type, char \*name\_of\_master);

NailsService \*deleteNails(NailsService \*b, int &N, char \*type, float price);

NailsService\*addNails(NailsService \*b, int &N);

struct Cosmetology {

MyString name\_master;

MyString type;

float price;

void fillcosmet() {

char tmp[256] = "";

cout << "Введите имя мастера -> ";

cin.getline(tmp, 256);

name\_master.setMyString(tmp);

cout << "Введите название услуги -> ";

cin.getline(tmp, 256);

type.setMyString(tmp);

cout << "Введите цену за данную услугу -> ";

cin >> price;

}

void fillcosmet2() {

char tmp[256] = "";

cin.get();

cout << "Введите имя мастера -> ";

cin.getline(tmp, 256);

name\_master.setMyString(tmp);

cout << "Введите название услуги -> ";

cin.getline(tmp, 256);

type.setMyString(tmp);

cout << "Введите цену за данную услугу -> ";

cin >> price;

}

void printcosmetRus() {

cout << "Имя мастера -> " << name\_master.str << " ;" << endl;

cout << "-------------------------------------" << endl;

cout << "Название услуги -> " << type.str << " ;" << endl;

cout << "-------------------------------------" << endl;

cout << "Цена за данную услугу -> " << price << " ." << endl;

}

void printcosmetEngl() {

cout << "Specialist name -> " << name\_master.str << " ;" << endl;

cout << "-------------------------------------" << endl;

cout << "Name of the service -> " << type.str << " ;" << endl;

cout << "-------------------------------------" << endl;

cout << "Price for this service -> " << price << " ." << endl;

}

void toFilecosmet(FILE \*fp) {

fwrite(&name\_master.length, sizeof(int), 1, fp);

fwrite(name\_master.str, sizeof(char), name\_master.length, fp);

fwrite(&type.length, sizeof(int), 1, fp);

fwrite(type.str, sizeof(char), type.length, fp);

fwrite(&price, sizeof(float), 1, fp);

}

void fromFilecosmet(FILE\*fp) {

fread(&name\_master.length, sizeof(int), 1, fp);

name\_master.str = new char[name\_master.length + 1];

fread(name\_master.str, sizeof(char), name\_master.length, fp);

name\_master.str[name\_master.length] = '\0';

fread(&type.length, sizeof(int), 1, fp);

type.str = new char[type.length + 1];

fread(type.str, sizeof(char), type.length, fp);

type.str[type.length] = '\0';

fread(&price, sizeof(float), 1, fp);

}

};

void fillarrcosmet(Cosmetology\*b, int N);

void printarrcosmetRus(Cosmetology\*b, int N);

void printarrcosmetEngl(Cosmetology\*b, int N);

void ArrToFilecosmet(Cosmetology\*b, int N);

Cosmetology \*ArrFromFilecosmet(int&N);

void sumTotalCosmetologyRus(Cosmetology \*b, int N, char \*name, char \*type, float &total\_price);

void sumTotalCosmetologyEngl(Cosmetology \*b, int N, char \*name, char \*type, float &total\_price);

void sortCosmetologyPrice(Cosmetology \*b, int N);

void changePriceCosm(Cosmetology \*b, int N, char \*type, char \*name\_of\_master);

Cosmetology \*deleteCosm(Cosmetology \*b, int &N, char \*type, float price);

Cosmetology\*addCosm(Cosmetology \*b, int &N);

struct MakeUp {

MyString name\_master;

MyString type;

float price;

void fillmakeup() {

char tmp[256] = "";

cout << "Введите имя мастера -> ";

cin.getline(tmp, 256);

name\_master.setMyString(tmp);

cout << "Введите название услуги -> ";

cin.getline(tmp, 256);

type.setMyString(tmp);

cout << "Введите цену за данную услугу -> ";

cin >> price;

}

void fillmakeup2() {

char tmp[256] = "";

cin.get();

cout << "Введите имя мастера -> ";

cin.getline(tmp, 256);

name\_master.setMyString(tmp);

cout << "Введите название услуги -> ";

cin.getline(tmp, 256);

type.setMyString(tmp);

cout << "Введите цену за данную услугу -> ";

cin >> price;

}

void printmakeupRus() {

cout << "Имя мастера -> " << name\_master.str << " ;" << endl;

cout << "-------------------------------------" << endl;

cout << "Название услуги -> " << type.str << " ;" << endl;

cout << "-------------------------------------" << endl;

cout << "Цена за данную услугу -> " << price << " ." << endl;

}

void printmakeupEngl() {

cout << "Specialist name -> " << name\_master.str << " ;" << endl;

cout << "-------------------------------------" << endl;

cout << "Name of the service -> " << type.str << " ;" << endl;

cout << "-------------------------------------" << endl;

cout << "Price for this service -> " << price << " ." << endl;

}

void toFilemakeup(FILE \*fp) {

fwrite(&name\_master.length, sizeof(int), 1, fp);

fwrite(name\_master.str, sizeof(char), name\_master.length, fp);

fwrite(&type.length, sizeof(int), 1, fp);

fwrite(type.str, sizeof(char), type.length, fp);

fwrite(&price, sizeof(float), 1, fp);

}

void fromFilemakeup(FILE\*fp) {

fread(&name\_master.length, sizeof(int), 1, fp);

name\_master.str = new char[name\_master.length + 1];

fread(name\_master.str, sizeof(char), name\_master.length, fp);

name\_master.str[name\_master.length] = '\0';

fread(&type.length, sizeof(int), 1, fp);

type.str = new char[type.length + 1];

fread(type.str, sizeof(char), type.length, fp);

type.str[type.length] = '\0';

fread(&price, sizeof(float), 1, fp);

}

};

void fillarrmakeup(MakeUp\*b, int N);

void printarrmakeupRus(MakeUp\*b, int N);

void printarrmakeupEngl(MakeUp\*b, int N);

void ArrToFilemakeup(MakeUp\*b, int N);

MakeUp \*ArrFromFilemakeup(int&N);

void sumTotalMakeUpRus(MakeUp \*b, int N, char \*name, char \*type, float &total\_price);

void sumTotalMakeUpEngl(MakeUp \*b, int N, char \*name, char \*type, float &total\_price);

void sortMakeUpPrice(MakeUp \*b, int N);

void changePriceMake(MakeUp \*b, int N, char \*type, char \*name\_of\_master);

MakeUp \*deleteMake(MakeUp \*b, int &N, char \*type, float price);

MakeUp\*addMake(MakeUp \*b, int &N);

void cinget();

//cout << "==========================================================================Чеккккккк===============================================================================================" << endl;

struct Check {

MyString name\_master;

MyString type;

float price;

void fillcheckRus() {

cin.get();

char tmp[256] = "";

cout << "Введите имя мастера -> ";

cin.getline(tmp, 256);

name\_master.setMyString(tmp);

cout << "Введите название услуги -> ";

cin.getline(tmp, 256);

type.setMyString(tmp);

cout << "Введите цену за данную услугу -> ";

cin >> price;

}

void fillcheckEngl() {

cin.get();

char tmp[256] = "";

cout << "Enter Specialist`s name -> ";

cin.getline(tmp, 256);

name\_master.setMyString(tmp);

cout << "Enter Name of the service -> ";

cin.getline(tmp, 256);

type.setMyString(tmp);

cout << "Enter the price of this service -> ";

cin >> price;

}

void printcheckRus() {

cout << "# Имя мастера -> " << name\_master.str << " ; # " << endl;

cout << "# Название услуги -> " << type.str << " ; #" << endl;

cout << "# Цена услуги -> " << price << " грн. ; #" << endl;

cout << "#---------------------------------------------------------------#" << endl;

}

void printcheckEngl() {

cout << "# Specialist`s name -> " << name\_master.str << " ; # " << endl;

cout << "# Name of the service -> " << type.str << " ; #" << endl;

cout << "# Price of this service -> " << price << " ua. ; #" << endl;

cout << "#---------------------------------------------------------------#" << endl;

}

void toFilecheck(FILE \*fp) {

fwrite(&name\_master.length, sizeof(int), 1, fp);

fwrite(name\_master.str, sizeof(char), name\_master.length, fp);

fwrite(&type.length, sizeof(int), 1, fp);

fwrite(type.str, sizeof(char), type.length, fp);

fwrite(&price, sizeof(float), 1, fp);

}

void fromFilecheck(FILE\*fp) {

fread(&name\_master.length, sizeof(int), 1, fp);

name\_master.str = new char[name\_master.length + 1];

fread(name\_master.str, sizeof(char), name\_master.length, fp);

name\_master.str[name\_master.length] = '\0';

fread(&type.length, sizeof(int), 1, fp);

type.str = new char[type.length + 1];

fread(type.str, sizeof(char), type.length, fp);

type.str[type.length] = '\0';

fread(&price, sizeof(float), 1, fp);

}

};

void fillarrcheckEngl(Check\*b, int N);

void fillarrcheckRus(Check\*b, int N);

void AddToFileCheck(Check\*c, int &N);

Check \*AddFromFileCheck(int&N);

void printCheckEngl(Check \*c, int N);

void printCheckRus(Check\*c, int N);

Check\*addCheck(Check \*b, int &N);

Check\*addCheckRus(Check \*b, int &N);

void totalPrice(Check \*c, int N, float &totalprice);

Services.cpp

#include "Services.h"

void fillarrbarber(BarberShop \* b, int N)

{

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

cin.get();

b[i].fillbarber();

}

}

void printarrbarberRus(BarberShop \* b, int N)

{

cout << "=====================================" << endl;

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

b[i].printbarberRus();

cout << "=====================================" << endl;

}

}

void printarrbarberEngl(BarberShop \* b, int N)

{

cout << "=====================================" << endl;

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

b[i].printbarberEngl();

cout << "=====================================" << endl;

}

}

void ArrToFilebarber(BarberShop \* b, int N)

{

FILE \*fp = fopen("barbershop.bin", "wb");

if (fp == NULL)

perror("Error//Ошибка");

else {

fwrite(&N, sizeof(int), 1, fp);

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

b[i].toFilebarber(fp);

}

}

fclose(fp);

}

BarberShop \* ArrFromFilebarber(int & N)

{

FILE\*fp = fopen("barbershop.bin", "rb");

if (fp == NULL) {

perror("Ошибка открытия файла");

N = 0;

return nullptr;

}

else {

fread(&N, sizeof(int), 1, fp);

BarberShop \*b = new BarberShop[N];

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

b[i].fromFilebarber(fp);

}

fclose(fp);

return b;

}

}

void sumTotalBarberRus(BarberShop \*b, int N, char \* name, char \* type, float & total\_price)

{

int c = 0;

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

if (strcmp(name, b[i].name\_master.str) == NULL && strcmp(type, b[i].type.str) == NULL) {

total\_price += b[i].price;

c++;

}

}

if (c == 0)

perror("\tНеправильно указаны имя мастера или тип услуги :(\n\tПопробуйте позже");

}

void sumTotalBarberEngl(BarberShop \* b, int N, char \* name, char \* type, float & total\_price)

{

int c = 0;

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

if (strcmp(name, b[i].name\_master.str) == NULL && strcmp(type, b[i].type.str) == NULL) {

total\_price += b[i].price;

c++;

}

}

if (c == 0)

perror("\tIncorrect name of the master or type of the service :(\n\tPlease, try again ");

}

void sortBarberPrice(BarberShop \* b, int N)

{

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

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

if (b[j].price>b[j + 1].price)

swap(b[j], b[j + 1]);

}

}

}

void changePriceBarber(BarberShop \* b, int N, char \*type, char \*name\_of\_master)

{

int c = 0;

float newprice = 0;

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

if (strcmp(b[i].type.str, type) == NULL&&strcmp(b[i].name\_master.str, name\_of\_master) == NULL) {

c++;

cout << "New Price(Новая цена) -> ";

cin >> newprice;

b[i].price = newprice;

}

}

if (c == 0)

perror("\t\t\tFail//Ошибка");

}

BarberShop \* deleteBarber(BarberShop \* b, int & N, char \* type, float price)

{

int c = 0;

N--;

BarberShop\*tmp = new BarberShop[N];

int j = 0;

for (int i = 0; i < N+1; i++) {

if (strcmp(b[i].type.str, type) == NULL && price==b[i].price) {

c++;

}

if (!(strcmp(b[i].type.str, type) == NULL && price == b[i].price))

{

tmp[j] = b[i];

j++;

}

}

if (c == 0) {

perror("\t\tFail//Ошибка");

N++;

return b;

}

else {

delete[]b;

return tmp;

}

}

BarberShop \* addBarber(BarberShop \* b, int & N)

{

BarberShop \*tmp = new BarberShop[N+1];

int j = 0;

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

tmp[j] = b[i];

j++;

}

tmp[j].fillbarber2();

//delete[]b;

N++;

return tmp;

}

void fillarrmassage(Massage \* b, int N)

{

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

cin.get();

b[i].fillmassage();

}

}

void printarrmassageRus(Massage \* b, int N)

{

cout << "=====================================" << endl;

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

b[i].printmassageRus();

cout << "=====================================" << endl;

}

}

void printarrmassageEngl(Massage \* b, int N)

{

cout << "=====================================" << endl;

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

b[i].printmassageEngl();

cout << "=====================================" << endl;

}

}

void ArrToFilemassage(Massage \* b, int N)

{

FILE \*fp = fopen("massage.bin", "wb");

if (fp == NULL)

perror("Error//Ошибка");

else {

fwrite(&N, sizeof(int), 1, fp);

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

b[i].toFilemassage(fp);

}

}

fclose(fp);

}

Massage \* ArrFromFilemassage(int & N)

{

FILE\*fp = fopen("massage.bin", "rb");

if (fp == NULL) {

perror("Ошибка открытия файла");

N = 0;

return nullptr;

}

else {

fread(&N, sizeof(int), 1, fp);

Massage \*b = new Massage[N];

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

b[i].fromFilemassage(fp);

}

fclose(fp);

return b;

}

}

void sumTotalMassageRus(Massage \* b, int N, char \* name, char \* type, float & total\_price)

{

int c = 0;

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

if (strcmp(name, b[i].name\_master.str) == NULL && strcmp(type, b[i].type.str) == NULL) {

total\_price += b[i].price;

c++;

}

}

if (c == 0)

perror("\tНеправильно указаны имя мастера или тип услуги :(\n\tПопробуйте позже");

}

void sumTotalMassageEngl(Massage \* b, int N, char \* name, char \* type, float & total\_price)

{

int c = 0;

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

if (strcmp(name, b[i].name\_master.str) == NULL && strcmp(type, b[i].type.str) == NULL) {

total\_price += b[i].price;

c++;

}

}

if (c == 0)

perror("\tIncorrect name of the master or type of the service :(\n\tPlease, try again ");

}

void sortMassagePrice(Massage \* b, int N)

{

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

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

if (b[j].price>b[j + 1].price)

swap(b[j], b[j + 1]);

}

}

}

void changePriceMassage(Massage \* b, int N, char \* type, char \* name\_of\_master)

{

int c = 0;

float newprice = 0;

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

if (strcmp(b[i].type.str, type) == NULL&&strcmp(b[i].name\_master.str, name\_of\_master) == NULL) {

c++;

cout << "New Price(Новая цена) -> ";

cin >> newprice;

b[i].price = newprice;

}

}

if (c == 0)

perror("\t\t\tFail//Ошибка");

}

Massage \* deleteMassage(Massage \* b, int & N, char \* type, float price)

{

int c = 0;

N--;

Massage\*tmp = new Massage[N];

int j = 0;

for (int i = 0; i < N + 1; i++) {

if (strcmp(b[i].type.str, type) == NULL && price == b[i].price) {

c++;

}

if (!(strcmp(b[i].type.str, type) == NULL && price == b[i].price))

{

tmp[j] = b[i];

j++;

}

}

if (c == 0) {

perror("\t\tFail//Ошибка");

N++;

return b;

}

else {

delete[]b;

return tmp;

}

}

Massage \* addMassage(Massage \* b, int & N)

{

Massage \*tmp = new Massage[N + 1];

int j = 0;

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

tmp[j] = b[i];

j++;

}

tmp[j].fillmassage2();

//delete[]b;

N++;

return tmp;

}

void fillarrnails(NailsService \* b, int N)

{

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

cin.get();

b[i].fillnails();

}

}

void printarrnailsRus(NailsService \* b, int N)

{

cout << "=====================================" << endl;

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

b[i].printnailsRus();

cout << "=====================================" << endl;

}

}

void printarrnailsEngl(NailsService \* b, int N)

{

cout << "=====================================" << endl;

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

b[i].printnailsEngl();

cout << "=====================================" << endl;

}

}

void ArrToFilenails(NailsService \* b, int N)

{

FILE \*fp = fopen("nailsservice.bin", "wb");

if (fp == NULL)

perror("Error//Ошибка");

else {

fwrite(&N, sizeof(int), 1, fp);

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

b[i].toFilenails(fp);

}

}

fclose(fp);

}

NailsService \* ArrFromFilenails(int & N)

{

FILE\*fp = fopen("nailsservice.bin", "rb");

if (fp == NULL) {

perror("Ошибка открытия файла");

N = 0;

return nullptr;

}

else {

fread(&N, sizeof(int), 1, fp);

NailsService \*b = new NailsService[N];

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

b[i].fromFilenails(fp);

}

fclose(fp);

return b;

}

}

void sumTotalNailsServiceRus(NailsService \* b, int N, char \* name, char \* type, float & total\_price)

{

int c = 0;

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

if (strcmp(name, b[i].name\_master.str) == NULL && strcmp(type, b[i].type.str) == NULL) {

total\_price += b[i].price;

c++;

}

}

if (c == 0)

perror("\tНеправильно указаны имя мастера или тип услуги :(\n\tПопробуйте позже");

}

void sumTotalNailsServiceEngl(NailsService \* b, int N, char \* name, char \* type, float & total\_price)

{

int c = 0;

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

if (strcmp(name, b[i].name\_master.str) == NULL && strcmp(type, b[i].type.str) == NULL) {

total\_price += b[i].price;

c++;

}

}

if (c == 0)

perror("\tIncorrect name of the master or type of the service :(\n\tPlease, try again ");

}

void sortNailsServicePrice(NailsService \* b, int N)

{

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

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

if (b[j].price>b[j + 1].price)

swap(b[j], b[j + 1]);

}

}

}

void changePriceNails(NailsService \* b, int N, char \* type, char \* name\_of\_master)

{

int c = 0;

float newprice = 0;

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

if (strcmp(b[i].type.str, type) == NULL&&strcmp(b[i].name\_master.str, name\_of\_master) == NULL) {

cout << "New Price(Новая цена) -> ";

cin >> newprice;

b[i].price = newprice;

c++;

}

}

if (c == 0)

perror("\t\t\tFail//Ошибка");

}

NailsService \* deleteNails(NailsService \* b, int & N, char \* type, float price)

{

int c = 0;

N--;

NailsService\*tmp = new NailsService[N];

int j = 0;

for (int i = 0; i < N + 1; i++) {

if (strcmp(b[i].type.str, type) == NULL && price == b[i].price) {

c++;

}

if (!(strcmp(b[i].type.str, type) == NULL && price == b[i].price))

{

tmp[j] = b[i];

j++;

}

}

if (c == 0) {

perror("\t\tFail//Ошибка");

N++;

return b;

}

else {

delete[]b;

return tmp;

}

}

NailsService \* addNails(NailsService \* b, int & N)

{

NailsService \*tmp = new NailsService[N + 1];

int j = 0;

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

tmp[j] = b[i];

j++;

}

tmp[j].fillnails2();

//delete[]b;

N++;

return tmp;

}

void fillarrcosmet(Cosmetology \* b, int N)

{

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

cin.get();

b[i].fillcosmet();

}

}

void printarrcosmetRus(Cosmetology \* b, int N)

{

cout << "=====================================" << endl;

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

b[i].printcosmetRus();

cout << "=====================================" << endl;

}

}

void printarrcosmetEngl(Cosmetology \* b, int N)

{

cout << "=====================================" << endl;

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

b[i].printcosmetEngl();

cout << "=====================================" << endl;

}

}

void ArrToFilecosmet(Cosmetology \* b, int N)

{

FILE \*fp = fopen("cosmetology.bin", "wb");

if (fp == NULL)

perror("Error//Ошибка");

else {

fwrite(&N, sizeof(int), 1, fp);

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

b[i].toFilecosmet(fp);

}

}

fclose(fp);

}

Cosmetology \* ArrFromFilecosmet(int & N)

{

FILE\*fp = fopen("cosmetology.bin", "rb");

if (fp == NULL) {

perror("Ошибка открытия файла");

N = 0;

return nullptr;

}

else {

fread(&N, sizeof(int), 1, fp);

Cosmetology \*b = new Cosmetology[N];

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

b[i].fromFilecosmet(fp);

}

fclose(fp);

return b;

}

}

void sumTotalCosmetologyRus(Cosmetology \* b, int N, char \* name, char \* type, float & total\_price)

{

int c = 0;

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

if (strcmp(name, b[i].name\_master.str) == NULL && strcmp(type, b[i].type.str) == NULL) {

total\_price += b[i].price;

c++;

}

}

if (c == 0)

perror("\tНеправильно указаны имя мастера или тип услуги :(\n\tПопробуйте позже");

}

void sumTotalCosmetologyEngl(Cosmetology \* b, int N, char \* name, char \* type, float & total\_price)

{

int c = 0;

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

if (strcmp(name, b[i].name\_master.str) == NULL && strcmp(type, b[i].type.str) == NULL) {

total\_price += b[i].price;

c++;

}

}

if (c == 0)

perror("\tIncorrect name of the master or type of the service :(\n\tPlease, try again ");

}

void sortCosmetologyPrice(Cosmetology \* b, int N)

{

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

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

if (b[j].price>b[j + 1].price)

swap(b[j], b[j + 1]);

}

}

}

void changePriceCosm(Cosmetology \* b, int N, char \* type, char \* name\_of\_master)

{

int c = 0;

float newprice = 0;

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

if (strcmp(b[i].type.str, type) == NULL&&strcmp(b[i].name\_master.str, name\_of\_master) == NULL) {

c++;

cout << "New Price(Новая цена) -> ";

cin >> newprice;

b[i].price = newprice;

}

}

if (c == 0)

perror("\t\t\tFail//Ошибка");

}

Cosmetology \* deleteCosm(Cosmetology \* b, int & N, char \* type, float price)

{

int c = 0;

N--;

Cosmetology\*tmp = new Cosmetology[N];

int j = 0;

for (int i = 0; i < N + 1; i++) {

if (strcmp(b[i].type.str, type) == NULL && price == b[i].price) {

c++;

}

if (!(strcmp(b[i].type.str, type) == NULL && price == b[i].price))

{

tmp[j] = b[i];

j++;

}

}

if (c == 0) {

perror("\t\tFail//Ошибка");

N++;

return b;

}

else {

delete[]b;

return tmp;

}

}

Cosmetology \* addCosm(Cosmetology \* b, int & N)

{

Cosmetology \*tmp = new Cosmetology[N + 1];

int j = 0;

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

tmp[j] = b[i];

j++;

}

tmp[j].fillcosmet2();

//delete[]b;

N++;

return tmp;

}

void fillarrmakeup(MakeUp \* b, int N)

{

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

cin.get();

b[i].fillmakeup();

}

}

void printarrmakeupRus(MakeUp \* b, int N)

{

cout << "=====================================" << endl;

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

b[i].printmakeupRus();

cout << "=====================================" << endl;

}

}

void printarrmakeupEngl(MakeUp \* b, int N)

{

cout << "=====================================" << endl;

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

b[i].printmakeupEngl();

cout << "=====================================" << endl;

}

}

void ArrToFilemakeup(MakeUp \* b, int N)

{

FILE \*fp = fopen("makeupology.bin", "wb");

if (fp == NULL)

perror("Error//Ошибка");

else {

fwrite(&N, sizeof(int), 1, fp);

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

b[i].toFilemakeup(fp);

}

}

fclose(fp);

}

MakeUp \* ArrFromFilemakeup(int & N)

{

FILE\*fp = fopen("makeupology.bin", "rb");

if (fp == NULL) {

perror("Ошибка открытия файла");

N = 0;

return nullptr;

}

else {

fread(&N, sizeof(int), 1, fp);

MakeUp \*b = new MakeUp[N];

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

b[i].fromFilemakeup(fp);

}

fclose(fp);

return b;

}

}

void sumTotalMakeUpRus(MakeUp \* b, int N, char \* name, char \* type, float & total\_price)

{

int c = 0;

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

if (strcmp(name, b[i].name\_master.str) == NULL && strcmp(type, b[i].type.str) == NULL) {

total\_price += b[i].price;

c++;

}

}

if (c == 0)

perror("\tНеправильно указаны имя мастера или тип услуги :(\n\tПопробуйте позже");

}

void sumTotalMakeUpEngl(MakeUp \* b, int N, char \* name, char \* type, float & total\_price)

{

int c = 0;

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

if (strcmp(name, b[i].name\_master.str) == NULL && strcmp(type, b[i].type.str) == NULL) {

total\_price += b[i].price;

c++;

}

}

if (c == 0)

perror("\tIncorrect name of the master or type of the service :(\n\tPlease, try again ");

}

void sortMakeUpPrice(MakeUp \* b, int N)

{

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

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

if (b[j].price>b[j + 1].price)

swap(b[j], b[j + 1]);

}

}

}

void changePriceMake(MakeUp \* b, int N, char \* type, char \* name\_of\_master)

{

int c = 0;

float newprice = 0;

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

if (strcmp(b[i].type.str, type) == NULL&&strcmp(b[i].name\_master.str, name\_of\_master) == NULL) {

c++;

cout << "New Price(Новая цена) -> ";

cin >> newprice;

b[i].price = newprice;

}

}

if (c == 0)

perror("\t\t\tFail//Ошибка");

}

MakeUp \* deleteMake(MakeUp \* b, int & N, char \* type, float price)

{

int c = 0;

N--;

MakeUp\*tmp = new MakeUp[N];

int j = 0;

for (int i = 0; i < N + 1; i++) {

if (strcmp(b[i].type.str, type) == NULL && price == b[i].price) {

c++;

}

if (!(strcmp(b[i].type.str, type) == NULL && price == b[i].price))

{

tmp[j] = b[i];

j++;

}

}

if (c == 0) {

perror("\t\tFail//Ошибка");

N++;

return b;

}

else {

delete[]b;

return tmp;

}

}

MakeUp \* addMake(MakeUp \* b, int & N)

{

MakeUp \*tmp = new MakeUp[N + 1];

int j = 0;

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

tmp[j] = b[i];

j++;

}

tmp[j].fillmakeup2();

//delete[]b;

N++;

return tmp;

}

void cinget()

{

cin.get();

}

void fillarrcheckEngl(Check \* b, int N)

{

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

b[i].fillcheckEngl();

}

}

void fillarrcheckRus(Check \* b, int N)

{

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

b[i].fillcheckRus();

}

}

void AddToFileCheck(Check\*c, int &N)

{

FILE \*fp = fopen("barbershopcheck.bin", "wb");

if (fp == NULL)

perror("Error//Ошибка");

else {

fwrite(&N, sizeof(int), 1, fp);

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

c[i].toFilecheck(fp);

}

}

fclose(fp);

}

Check \*AddFromFileCheck(int&N)

{

FILE\*fp = fopen("barbershopcheck.bin", "rb");

if (fp == NULL) {

perror("Ошибка открытия файла");

N = 0;

return nullptr;

}

else {

fread(&N, sizeof(int), 1, fp);

Check \*c = new Check[N];

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

c[i].fromFilecheck(fp);

}

fclose(fp);

return c;

}

}

void printCheckEngl(Check \* c, int N)

{

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

c[i].printcheckEngl();

}

}

void printCheckRus(Check \* c, int N)

{

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

c[i].printcheckRus();

}

}

Check \* addCheck(Check \* b, int & N)

{

Check \*tmp = new Check[N + 1];

int j = 0;

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

tmp[j] = b[i];

j++;

}

tmp[j].fillcheckEngl();

//delete[]b;

N++;

return tmp;

}

Check \* addCheckRus(Check \* b, int & N)

{

Check \*tmp = new Check[N + 1];

int j = 0;

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

tmp[j] = b[i];

j++;

}

tmp[j].fillcheckRus();

//delete[]b;

N++;

return tmp;

}

void totalPrice(Check \* c, int N,float &totalprice)

{

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

totalprice += c[i].price;

}

}

MyString.h

#pragma once

#include "Libraries.h"

struct MyString {

//поля

int length = 0;

char\* str = nullptr;

//методы

void setMyString(char s[]) {

length = strlen(s);

str = new char[length + 1];

strcpy(str, s);

}

void delMyString() {

length = 0;

delete str;

}

void toFirstUpper() {

str[0] = toupper(str[0]);

}

void toAllLower() {

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

{

str[i] = tolower(str[i]);

}

}

void oneSpaсe() {

int j = 0;

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

{

if (!isspace(str[i]))

{

str[j] = str[i]; j++;

}

else if (isspace(str[i]) && !isspace(str[i - 1]))

{

str[j] = str[i]; j++;

}

}

str[j] = '\0';

length = j;

}

};

Data.h

#pragma once

#include "Libraries.h"

struct Date {

int year = 0;

int month = 0;

int day = 0;

int hour = 0;

int minute = 0;

void fillDateRus() {

cout << "Введите год -> ";

cin >> year;

cout << "Введите месяц -> ";

cin >> month;

cout << "Введите день -> ";

cin >> day;

cout << "Введите часы -> ";

cin >> hour;

cout << "Введите минуты-> ";

cin >> minute;

}

void fillDateEngl() {

cout << "Enter year -> ";

cin >> year;

cout << "Enter month -> ";

cin >> month;

cout << "Enter day -> ";

cin >> day;

cout << "Enter hours -> ";

cin >> hour;

cout << "Enter minutes -> ";

cin >> minute;

}

void getNowDate() {

// создаем стд объект даты и времени

struct tm \*tim = new tm;

// определяем текущую млисекунду

time\_t tt = time(NULL);

//заполняем все поля стд структуры

localtime\_s(tim, &tt);

//заполняем нужные поля нашей структуры

year = 1900 + tim->tm\_year;

month = 1 + tim->tm\_mon;

day = tim->tm\_mday;

hour = tim->tm\_hour;

minute = tim->tm\_min;

}

void printDate() {

cout << year << "/" << month / 10 << month % 10 << "/" << day << "\t\t\t\t\t\t" << hour << ":" << minute;

}

void printDateforChek() {

cout << "#################################################################" << endl;

cout << "# " << hour << ":" << minute << " #" << endl;

cout << "# " << year << "/" << month / 10 << month % 10 << "/" << day << " #" << endl;

}

char\* toString()

{

char\* str = new char[11];

sprintf(str, "%d/%d%d/%d%d", year, month / 10, month % 10, day / 10, day % 10);

return str;

}

bool isLeapYear() {

if (year % 4 == 0 && !(year % 100 == 0 && year % 400 != 0))

return true;

else

return false;

}

int monthDays() {

if (month == 1 || month == 3 || month == 5 || month == 7 || month == 8 || month == 10 || month == 12)

return 31;

else if (month == 4 || month == 6 || month == 9 || month == 11)

return 30;

else if (isLeapYear())

return 29;

else

return 28;

}

void toNextDay() {

if (month == 12 && day == 31)

{

month = 1;

day = 1;

year++;

}

else if (day == monthDays())

{

day = 1;

month++;

}

else

{

day++;

}

}

};

Administrators.h

#pragma once

#include"MyString.h"

#pragma once

#include"Data.h"

struct Admins {

MyString name;

MyString password\_name;

void fillAlladmins() {

char tmp[256] = "";

cout << "Введите имя администратора -> ";

cin.getline(tmp, 256);

name.setMyString(tmp);

cout << "Введите пароль администратора -> ";

cin.getline(tmp, 256);

password\_name.setMyString(tmp);

}

void printAlladmins() {

cout << " Имя администратора - " << name.str << endl;

cout << "Пароль администратора - " << password\_name.str << endl;

}

void toFileadmins(FILE\*fp) {

fwrite(&name.length, sizeof(int), 1, fp);

fwrite(name.str, sizeof(char), name.length, fp);

fwrite(&password\_name.length, sizeof(int), 1, fp);

fwrite(password\_name.str, sizeof(char), password\_name.length, fp);

}

void fromFileadmins(FILE \*fp) {

fread(&name.length, sizeof(int), 1, fp);

name.str = new char[name.length + 1];

fread(name.str, sizeof(char), name.length, fp);

name.str[name.length] = '\0';

fread(&password\_name.length, sizeof(int), 1, fp);

password\_name.str = new char[password\_name.length + 1];

fread(password\_name.str, sizeof(char), password\_name.length, fp);

password\_name.str[password\_name.length] = '\0';

}

};

void arrToFileadmins(Admins \*a, int N);

Admins \* arrFromFileadmins(int& N);

void filladmins(Admins \*a, int N);

void printadmins(Admins \*a, int N);

bool checkPassword(Admins \*a, int N, char \*username, char \*password);

Administrators.cpp

#pragma once

#include "Administrators.h"

void arrToFileadmins(Admins \* a, int N)

{

FILE \*fp = fopen("passwords.bin", "wb");

if (fp == NULL)

perror("Ошибка открытия файла");

else {

fwrite(&N, sizeof(int), 1, fp);

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

a[i].toFileadmins(fp);

}

fclose(fp);

}

}

Admins \* arrFromFileadmins(int & N)

{

FILE \*fp = fopen("passwords.bin", "rb");

if (fp == NULL)

{

perror("Ошибка открытия файла");

N = 0;

return nullptr;

}

else {

fread(&N, sizeof(int), 1, fp);

Admins \* a = new Admins[N];

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

a[i].fromFileadmins(fp);

}

fclose(fp);

return a;

}

}

void filladmins(Admins \* a, int N)

{

cin.get();

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

a[i].fillAlladmins();

}

}

void printadmins(Admins \* a, int N)

{

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

a[i].printAlladmins();

cout << "------" << endl;

}

}

bool checkPassword(Admins \* a, int N, char \* username, char \*password)

{

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

if (strcmp(username, a[i].name.str) == NULL) {

if (strcmp(password, a[i].password\_name.str) == NULL)

return true;

else

return false;

}

}

}

Libraries.h

#include<iostream>

#include<clocale>

#include<cstdio>

#include<cmath>

#include<ctime>

#include<Windows.h>

#include<cstring>

using namespace std;