**Лабораторная работа №6**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 6,18,30 | Проект: Почта (ценное письмо)  Создайте структуру **letter** со следующими полями:   |  |  | | --- | --- | | Тип данных | Назначение поля | | double | Оценка письма | | int | Индекс отправителя | | char\* | Фамилия отправителя | | char\* | Имя отправителя | | int | Индекс получателя | | char\* | Фамилия получателя | | char\* | Имя получателя |   Реализовать функции :   * добавить новое письмо; * распечатать информацию о письме в табличном виде; * Найти все письма заданного отправителя (отправитель – это фамилия + имя) результат вывести на экран; * найти все письма с оценкой большей заданного, результат сортировать по алфавиту (по получателю), запомнить в массиве и вывести на экран; |

**Задание**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include <clocale>

#include <cstdio>

#include <cstdlib>

#include <cstring>

#include <cmath>

#include <windows.h>

#include <iomanip>

using namespace std;

struct letter

{

double mark;

int indexOt;

char\* surnameOt;

char\* nameOt;

int indexPol;

char\* surnamePol;

char\* namePol;

};

void tablPrint(letter\* mass, int ot, int dok)

{

printf\_s("%-12s %-15s %-12s %-17s %-15s %-13s %-11s\n", "Оценка", "Индекс", "Фамилия", "Имя", "Индекс", "Фамилия", "Имя");

printf\_s("%-12s %-15s %-12s %-17s %-15s %-13s %-11s\n", "письма", "отправ.", "отправ.", "отправ.", "получ.", "получ.", "получ.");

printf\_s("----------------------------------------------------------------------------------------------\n");

for (int i = ot; i < dok; i++)

{

printf\_s("%-12f %-15d %-12s %-18s", mass[i].mark, mass[i].indexOt, mass[i].surnameOt, mass[i].nameOt);

printf\_s("%-16d %-14s %-11s\n", mass[i].indexPol, mass[i].surnamePol, mass[i].namePol);

}

printf\_s("---------------------------------------------------------------------------------------------------\n");

}

void ways()

{

printf\_s("\nДобавить письмо------------------1\n");

printf\_s("Вывод всей таблицы---------------2\n");

printf\_s("Вывод таблицы для одного письма--3\n");

printf\_s("Поиск писем отправителя----------4\n");

printf\_s("Поиск писем с большей оценкой----5\n");

printf\_s("Выход----------------------------6\n");

}

void searchLetters(letter\* mass, int N, char\* nameOtprv, char\* surnameOtprv)

{

int forSurname, forName, t, a = 0;

int\* numbersLetters = (int\*)malloc(sizeof(int));

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

{

forName = strcmp(mass[i].nameOt, nameOtprv);

forSurname = strcmp(mass[i].surnameOt, surnameOtprv);

if ((forSurname == 0) && (forName == 0))

{

numbersLetters = (int\*)realloc(numbersLetters, sizeof(int) \* (a + 1));

numbersLetters[a] = i;

a++;

}

}

if (a != 0)

{

printf\_s("\nПисьма этого отправителя по оценке:\n");

for (int j = 0; j < N; j++)

{

t = 0;

for (int k = 0; k < a; k++)

if (j == numbersLetters[k]) t++;

if (t >= 1)

{

printf\_s("%f\n", mass[j].mark);

}

}

}

else if (a == 0)

printf\_s("\nНе нашлось писем заданного отправителя\n");

free(numbersLetters);

}

int searchRating(letter\* mass, int N, char\*\*\* addressees, double ratingMail)

{

int a = 0;

char str[100];

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

{

if (mass[i].mark > ratingMail)

{

(\*addressees) = (char\*\*)realloc((\*addressees), sizeof(char\*) \* (a + 1));

(\*addressees)[a] = mass[i].surnamePol;

a++;

}

}

if (a != 0)

{

for (int i = 0; i < (a - 1); i++)

{

for (int j = i + 1; j < a; j++)

{

if (strcmp((\*addressees)[i], (\*addressees)[j]) > 0)

{

strcpy(str, (\*addressees)[i]);

strcpy((\*addressees)[i], (\*addressees)[j]);

strcpy((\*addressees)[j], str);

}

}

}

printf\_s("\nПолучатели письма с такой оценкой(пофамильно):\n");

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

printf\_s("%s\n", (\*addressees)[i]);

return a;

}

else

{

printf\_s("\nПолучателей письма с оценкой больше %f не найдено\n", &ratingMail);

return 0;

}

}

int main()

{

setlocale(0, "Russian");

letter\* mail;

mail = (letter\*)malloc(sizeof(letter) \* 1);

int indicator = 1, vibor, sizeMail = 0, ntail = 2000, sizeAddresses = 0;

float rating;

char strokaVvoda[100], \* nameOfOtpr = nullptr, \* surnameOfOtpr = nullptr, \*\* addresses;

addresses = (char\*\*)malloc(sizeof(char\*) \* 1);

printf\_s("База данных <<Почта>>\n");

while (indicator == 1)

{

ways();

printf\_s("\nВыберите номер согласно действию:");

scanf("%d", &vibor);

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

switch (vibor)

{

case 1:

{

tablPrint(mail, 0, sizeMail);

break;

}

case 2:

{

printf\_s("\nВведите оценку письма:");

scanf("%f", &rating);

for (int j = 0; j < sizeMail; j++)

{

if (mail[j].mark == rating)

ntail = j;

}

if (ntail == 2000)

printf("\nНет писем с такой оценкой");

tablPrint(mail, ntail, ntail + 1);

break;

}

case 3:

{

while (fgetc(stdin) != '\n');

printf\_s("Введите фамилию отправителя:");

gets\_s(strokaVvoda, 100);

surnameOfOtpr = (char\*)malloc(1 + strlen(strokaVvoda));

strcpy(surnameOfOtpr, strokaVvoda);

printf\_s("Введите имя отправителя:");

gets\_s(strokaVvoda, 100);

nameOfOtpr = (char\*)malloc(1 + strlen(strokaVvoda));

strcpy(nameOfOtpr, strokaVvoda);

searchLetters(mail, sizeMail, nameOfOtpr, surnameOfOtpr);

break;

}

case 4:

{

printf\_s("Введите оценку:");

scanf("%f", &rating);

sizeAddresses = searchRating(mail, sizeMail, &addresses, rating);

break;

}

case 5:

indicator = 0;

break;

}

}

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

{

free(mail[i].nameOt);

free(mail[i].surnameOt);

free(mail[i].surnamePol);

free(mail[i].namePol);

}

free(mail);

