

Лабораторна робота №9

Тема: Розробка та реалізація програм із використанням структур та файлового введення-виведення.

Ціль роботи: отримати навички створення та реалізації програм, що використовують структури та реалізують операції введення-виведення із файлами.

Завдання до лабораторної роботи

Завдання 9.1. Нехай є наступний фрагмент програми, поданий у варіантах (табл. 9.1). Поясніть яку задачу реалізує вказаний фрагмент, при наступному початку програми.

```
#include <iostream>
#include <cstring>
#include <cctype>
using namespace std;
const int N=4;
struct myType
{
    char fff[20];
    char aaa[20];
    int b;
    double c;
};
void printStruct(myType st[],int n)
{
    for (int i=0;i<n;i++){
        cout << " fff: " << st[i].fff << "   aaa: " << st[i].aaa ;
        cout << "   b : " << st[i].b<<"   c : " << st[i].c << endl;
    }
}
int main(){
    myType st[]={{"assdss","ssdhshf",4,4.5},
                 {"jherhhgb","xczvf dg",9,2.4},
                 {"xcvfd","hhgeryt",5,4.1},
                 {"jrhjdfd","tyyyyy",7,3.5}    };
    printStruct(st,N);
```

Таблиця 9.1 – Варіанти завдання 9.1

№	Фрагмент програми	№	Фрагмент програми
1-5	<pre>double k=0; char c='d'; for (int i=0; i<N; i++){ if (strchr(st[i].fff,c)) k++; }</pre>	6-10	<pre>char t[20]; for (int i=0; i< N; i++) { for (int j= N-1; j>i; j--){ if (strcmp(st[j].aaa ,st[j-1].aaa)<0) { strcpy(t, st[j].aaa); strcpy(st[j].aaa, st[j-1].aaa); strcpy(st[j-1].aaa ,t); } } }</pre>
11-15	<pre>double a=0; for (int i=0; i< N; i++) a+=st[i].c; a=a/N; for (int i=0; i< N; i++) if (st[i].c>a) cout<<st[i].fff<<"\n";</pre>	16-20	<pre>for (int i=0; i< N; i++) if ((*st+i).b%2) cout<<(*st+i).fff<<"\n";</pre>
21-25	<pre>char c='j';char *l; for (int i=0; i<N; i++){ if (l=strchr(st[i].fff,c)) for (int j=0; j<strlen(l); j++) st[i].fff[j]-=32; }</pre>	26-30	<pre>char c='f';char *l; for (int i=0; i<N; i++){ if (l=strchr(st[i].aaa,c)) { int k=strlen(st[i].aaa); for (int j=0;j<k/2;j++) { char c=st[i].aaa[j]; st[i].aaa[j]=st[i].aaa[k-j-1]; st[i].aaa[k-j-1]=c; } } }</pre>

Завдання 9.2. Створити структуру, специфікація якої наведена нижче.

Визначити функції, що створюють масив структур. Завдати критерій відбору.

1. Student: id, Прізвище, Ім'я, По батькові, Дата народження, Адреса, Телефон, Факультет, Курс, Група.

Створити масив структур. Вивести:

- список студентів вказаного факультету;
- список студентів, що народились після вказаного року;
- список навчальної групи в алфавітному порядку.

2. Customer: id, Прізвище, Ім'я, По батькові, Адреса, Номер кредитної картки, Номер банківського рахунку.

Створити масив структур. Вивести:

- список покупців в алфавітному порядку;

b) список покупців, у яких номер кредитної картки знаходиться в заданому інтервалі;

c) список покупців, чия у яких номер банківського рахунку закінчується на вказану цифру.

3. Patient: id, Прізвище, Ім'я, По батькові, Адреса, Телефон, Номер медичної картки, Діагноз.

Створити масив структур. Вивести:

a) список пацієнтів, що мають вказаний діагноз;

b) список пацієнтів чий номер медичної картки при діленні на 7 дасть вказану остачу;

c) список пацієнтів, номер медичної карти яких знаходиться в заданому інтервалі.

4. Abiturient: id, Прізвище, Ім'я, По батькові, Адреса, Телефон, Оцінки.

Створити масив структур. Вивести:

a) список абітурієнтів, що мають незадовільні оцінки;

b) список абітурієнтів, у яких сума балів вище заданої;

c) вибрати вказану кількість n абітурієнтів, що мають найбільшу суму балів.

5. Book: id, Назва, Автор (и), Видавництво, Рік видання, Кількість сторінок, Ціна, Тип папіру.

Створити масив структур. Вивести:

a) список книг заданого автора;

b) список книг, що видані вказаним видавництвом;

c) список книг, що видані після заданого року.

6. House: id, Номер квартири, Площа, Поверх, Кількість кімнат, Вулиця, Тип будівлі, Термін експлуатації.

Створити масив структур. Вивести:

a) список квартир, які мають задану кількість кімнат;

b) список квартир, що мають вказану кількість кімнат і розташованих між вказаними поверхами;

c) список квартир, які мають площу, що більше заданої.

7. Phone: id, Прізвище, Ім'я, По батькові, Адреса, Номер кредитної картки, Дебет, Кредит, Час міських та міжнародних розмов.

Створити масив структур. Вивести:

a) відомості про абонентів, у яких час міських розмов перевищує вказаний;

b) відомості про абонентів, які користувались міжміським зв'язком;

c) відомості про абонентів в алфавітному порядку.

8. Car: id, Марка, Модель, Рік випуску, Колір, Ціна, Реєстраційний номер.

Створити масив структур. Вивести:

a) список автомобілів заданої марки;

b) список автомобілів заданої моделі, які експлуатуються більше n років;

c) список автомобілів вказаного року випуску, ціна яких більше вказаної.

9. Product: id, Найменування, Тип, Виробник, Ціна, Термін зберігання, Кількість.

Створити масив структур. Вивести:

a) список товарів заданого найменування;

b) список товарів заданого найменування, ціна яких не більше заданої;

c) список товарів, термін зберігання яких більше заданого.

10. Train: Пункт призначення, Номер поїзда, Час відправлення, Число місць (загальних, плацкарт, купе, люкс).

Створити масив структур. Вивести:

a) список поїздів, які прямують до заданого пункту призначення;

b) список поїздів, які прямують до заданого пункту призначення та відправляються після вказаної

години;

c) список поїздів, які відправляються до заданого пункту призначення та мають загальні місця.

Завдання 9.3. Створити файл для зберігання даних, що зберігаються у масивах структур, описаних у завданні 9.2. Розробити програму, що зчитує дані з файлу у масив структур та записує дані у файл.

Виконати запити, описані в завданні 9.2 із використанням масивів структур, та (якщо це можливо) без них.

Приклад виконання роботи

Завдання 9.1. Визначити дію фрагмента програми

```
char c='j';
for (int i=0; i<N; i++){
    if (st[i].fff[0]==c){
        strcat(st[i].fff,"123");
    }
}
```

Розв'язання

Цей фрагмент програми додає до символного поля структури послідовність 123, якщо рядок починається з символу 'j'.

```
fff: asdss  aaa: ssdhshf  b : 4    c : 4.5
fff: jherhhgb123  aaa: xczvfdg  b : 9    c : 2.4
fff: xcvfd  aaa: hhgeryt  b : 5    c : 4.1
fff: jrhjdffd123  aaa: tyuyuy  b : 7    c : 3.5
```

Завдання 9.3. Скласти програму, яка зберігає дані про студентів (прізвище, ім'я, по батькові, номер групи) у файл, зчитує дані з файлу та виводить на екран, упорядковує дані в алфавітному порядку.

Розв'язання

1. Алгоритм розв'язання задачі

Алгоритм розв'язання задачі можна представити у вигляді такої послідовності дій:

Дія 1. Виконувати дію 2, дію 3, дію 4, дію 5 в нескінченному циклі.

Дія 2. Якщо обрана дія 2, тоді

Дія 2.1. Визначити масив структур `sd`.

Дія 2.2. Ввести значення полів структури.

Дія 2.3. Ввести ім'я файлу.

Дія 2.4. Додати дані до існуючого файлу або створити новий файл.

Дія 3. Якщо обрана дія 3, тоді

Дія 3.1. Ввести ім'я файлу.

Дія 3.2. Якщо файл відкривається перейти до дії 3.3, інакше до головного меню.

Дія 3.3. Визначити розмір масиву структур `sd`.

Дія 3.4. Зчитати дані з файлу у масив структур `sd`.

Дія 3.5. Ввести значення полів масиву структур `sd`.

Дія 4. Якщо обрана дія 4, тоді

Дія 4.1. Ввести ім'я файлу.

Дія 4.2. Якщо файл відкривається перейти до дії 4.3, інакше до головного меню.

Дія 4.3. Визначити розмір масиву структур `sd`.

Дія 4.4. Зчитати дані з файлу у масив структур `sd`.

Дія 4.5. Відсортувати масив структур за прізвищем.

Дія 4.6. Вивести елементи масиву після сортування.

Дія 4.7. Запитати у користувача необхідність збереження упорядкованих даних, якщо так, тоді перейти до дії 4.8, інакше до головного меню.

Дія 4.8. Переписати файл

Дія 5. Якщо обрана дія 5, тоді кінець роботи програми.

2. Текст програми

```
#include <iostream>
#include <cstring>
#include <fstream>
#include <windows.h>

using namespace std;
struct stud{
    char surname[30];
    char name[30];
    char grupa[10];};

void enterStruct(stud sd[], int kol){
    for(int i=0; i < kol; i++){
        cout << "\nПрізвище: ";cin >> sd[i].surname;
        cout << "Им'я: "; cin >> sd[i].name;
        cout << "Група: "; cin >> sd[i].grupa; }
}

void fileWrite(char* txt,stud sd[], int kol) {
    ofstream t(txt,ios::binary | ios::app);
    for(int i=0;i<kol;i++){
        t.write((char*)&sd[i], sizeof sd[i]); }
    t.close(); }

void printStruct(stud* sd,int kol){
    for(int i = 0; i < kol; i++){
        cout << "\nПрізвище: "<< sd[i].surname;
        cout << "\nИм'я: " << sd[i].name;
        cout << "\nГрупа: " << sd[i].grupa<<"\n";}}
```

```

int fileOfSize (char *txt) {
    ifstream t(txt,ios::binary);
    if (!t.is_open()){
        cout << "Файл не може бути відкритий !\n";return 0; }
    t.seekg(0,ios_base::end);
    int k = t.tellg()/sizeof (stud);
    t.close();
    return k;}

```

```

void readFile(char *txt,stud* sd,int kol) {
    ifstream t(txt,ios::binary );
    for(int i=0;i<kol;i++){
t.read((char*)&sd[i], sizeof sd[i]); }
    t.close();}

```

```

void rewriteFile(char* txt,stud* sd,int kol) {
    ofstream t(txt,ios::binary);
    for(int i=0;i<kol;i++){
        t.write((char*)&sd[i], sizeof sd[i]);}
    t.close();}

```

```

void sortStud(stud* sd,int kol){
    stud s;
    for(int i=1;i<kol;i++)
    for(int j=kol-1;j>=i;j--){
        if(stricmp(sd[j].surname, sd[j-1].surname) < 0){
            s = sd[j];
            sd[j]=sd[j-1];
            sd[j-1]=s;
        }
        else if(stricmp(sd[j].surname, sd[j-1].surname) ==0){
            if(stricmp(sd[j].name, sd[j-1].name) < 0){
                s = sd[j];
                sd[j]=sd[j-1];
                sd[j-1]=s;
            }
        }
    }
}

```



```
}}
```

```
int main(){
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    char fName[80];
    int k = 0; stud * sd;
    while(1){
        cout<<"0: Exit\n";
        cout<<"1: Додати записи у файл\n";
        cout<<"2: Вивести вміст файлу на екран\n";
        cout<<"3: Сортування за прізвищем\n ";
        cin >> k;
        switch(k){
            case 1:
                cout<<"\nВведіть кількість студентів: "; int zn;
                cin>>zn; sd = new stud[zn];
                enterStruct(sd, zn);
                cout << "\nВведіть ім'я файлу: ";cin.ignore();
                cin.getline(fName,80);
                fileWrite(fName,sd,zn);
                break;
            case 2:
                cout << "\nВведіть ім'я файлу: ";cin.ignore();
                cin.getline(fName,80);
                zn=fileOfSize(fName);
                if (zn!=0){sd = new stud[zn];
                readFile(fName,sd,zn);
                cout << "\n Вміст файлу: \n";
                printStruct(sd,zn);cin.ignore();}
                break;
            case 3:
                cout << "\n Сортування даних за прізвищем: \n";
                cout << "\nВведіть ім'я файлу: ";cin.ignore();
                cin.getline(fName,80);
                zn=fileOfSize(fName);
```

```
        if (zn!=0){sd = new stud[zn];
        readFile(fName, sd, zn);
        sortStud(sd, zn);
        printStruct(sd, zn);
        cout << "\nЗберегти упорядковані дані? (Y/N): ";
        char tak;cin>>tak;
        if (tak=='Y' || tak =='y')
        rewriteFile(fName, sd, zn);}
        break;
    case 0:
        exit(0);
        break;
    }}
return 0;
}
```