

Центральноукраїнський національний технічний університет  
Кафедра програмування комп'ютерних систем і мереж  
Дисципліна: Базові методології та технології програмування

## Звіт

з лабораторних робіт №22

Теми: Програмування динамічно зв'язаних структур

Мета: Отримати знання та навички у програмуванні динамічно зв'язаних структур.

Виконав: ст. гр. КМ-20

Калиновський В.М.

Перевірила: доцент кафедри ПКСМ

Рибакова Л.В.

Варіант 3

Дата виконання: 30 травня 2021

Дата оформлення: 30 травня 2021

**Завдання 1.** Створити динамічно структуру — список студентів, що містить не менше 10 студентів. Для кожного студента зазначити рік народження та оцінки по чотирьом екзаменам. Реалізувати додання нових структур, видалення та виведення на екран. Організувати відбір елементів у окремий масив для стеків, що відносяться до студентів із середнім балом менше 4.

**Робота.** Відразу опишу код програми:

```
#include <windows.h>
#include <cstdio>
#include <iostream>
using namespace std;

/*
struct saveloadstud
{
    char name[20];
    int year, ratingOne, ratingTwo, ratingThree, ratingFour;
};
*/

struct STUD
{
    char name[20];
    int year, ratingOne, ratingTwo, ratingThree, ratingFour;
    STUD *next;
};

STUD *firsts; //Глобальный указатель на первый элемент
//STUD *tail; //Глобальный указатель на последний элемент (для загрузки)

void addStruct() //Функция заполнения
{
    STUD *newS = new STUD;
    if(firsts == NULL)
    {
        newS->next = NULL;
        firsts = newS;
    }
    else
    {
        newS->next = firsts;
        firsts = newS;
    }
    cout << "\nВведите фамилию студента: ";
    fflush(stdin);
    cin >> newS->name;
    cout << "Введите год рождения: ";
    cin >> newS->year;
    cout << "Введите балл по первому предмету: ";
    cin >> newS->ratingOne;
    cout << "Введите балл по второму предмету: ";
    cin >> newS->ratingTwo;
    cout << "Введите балл по третьему предмету: ";
```

```

    cin >> newS→ratingThree;
    cout << "Введите балл по четвёртому предмету: ";
    cin >> newS→ratingFour;
}

void delStruct()
{
    if(firsts≠NULL)
    {
        STUD *delS;
        delS = firsts;
        firsts = firsts→next;
        delete(delS);
        cout << "\nПоследняя структура удалена!";

    }
    else
        cout << "\nСтруктуры не обнаружены... ";
}

void printStruct()
{
    if(firsts≠NULL)
    {
        STUD *currS;
        currS = firsts;
        do
        {
            cout << "\nФамилия и инициалы студента: " << currS→name << "\n";
            cout << "Год рождения: " << currS→year << "\n";
            cout << "Балл по первому предмету: " << currS→ratingOne << "\n";
            cout << "Балл по второму предмету: " << currS→ratingTwo << "\n";
            cout << "Балл по третьему предмету: " << currS→ratingThree << "\n";
            cout << "Балл по четвёртому предмету: " << currS→ratingFour << "\n\n";
            currS = currS→next;
        }
        while(currS ≠ NULL);
    }
    else
        cout << "\nСтруктуры не обнаружены... ";
}

/*

void saveStruct()
{
    // NOTE: может придётся исправлять
    if(firsts≠NULL)
    {
        STUD *currS;
        currS = firsts;
        do
        {
            saveloadstud sa_st;

```

```

        for(int i = 0; i < 50; i++)
            sa_st.name[i] = currS→name[i];
        sa_st.year = currS→year;
        sa_st.ratingOne = currS→ratingOne;
        sa_st.ratingTwo = currS→ratingTwo;
        sa_st.ratingThree = currS→ratingOne;
        sa_st.ratingFour = currS→ratingOne;
        cout << "\n" << sa_st.name << " " << sa_st.year << " " <<
sa_st.ratingOne << " " << sa_st.ratingTwo << " " << sa_st.ratingThree << " " <<
sa_st.ratingFour;
        FILE *structs = fopen("Students.bin", "wb");
        fflush(structs);
        fwrite(currS, sizeof(currS), 70, structs);
        fclose(structs);
        currS = currS→next;
        //delete sa_st;
    }
    while(currS ≠ NULL);
    cout << "\nСтруктуры сохранены!";
}
else
    cout << "\nСтруктуры не обнаружены... ";
}

```

```

void forLoadStruct(int lstr)
{
    STUD *currS = new STUD;
    tail = currS;
    if(firsts == NULL)
    {
        saveloadstud *lo_st = new saveloadstud;
        FILE *structs = fopen("Students.bin", "rb");
        //fseek(structs, lstr * 438, SEEK_SET);
        fread(&lo_st, sizeof(lo_st), 55, structs);
        fclose(structs);
        for(int i = 0; i < 50; i++)
            currS→name[i] = lo_st→name[i];
        currS→year = lo_st→year;
        currS→ratingOne = lo_st→ratingOne;
        currS→ratingTwo = lo_st→ratingTwo;
        currS→ratingThree = lo_st→ratingThree;
        currS→ratingFour = lo_st→ratingFour;
        cout << "\n" << lo_st→name << " " << lo_st→year << " " << lo_st-
>ratingOne << " " << lo_st→ratingTwo << " " << lo_st→ratingThree << " " <<
lo_st→ratingFour;
        currS→next = NULL;
        firsts = currS;
        tail = currS;
    }
    else
    {
        saveloadstud *lo_st = new saveloadstud;
        FILE *structs = fopen("Students.bin", "rb");
        //fseek(structs, lstr * 55, SEEK_SET);
    }
}

```

```

        fread(&lo_st, sizeof(lo_st), 55, structs);
        fclose(structs);
        for(int i = 0; i < 50; i++)
            currS->name[i] = lo_st->name[i];
        currS->year = lo_st->year;
        currS->ratingOne = lo_st->ratingOne;
        currS->ratingOne = lo_st->ratingOne;
        currS->ratingOne = lo_st->ratingOne;
        currS->ratingOne = lo_st->ratingOne;
        currS->next = NULL;
        tail->next = currS;
        tail = currS;
    }
    fflush(stdin);
}

void loadStruct()
{
    FILE *structs = fopen("Students.bin", "rb");
    int size = 0, lstr = 0;
    fseek(structs, 0, SEEK_SET);
    while(getc(structs) != EOF) size++;
    fclose(structs);
    cout << "\n" << size << "\n";
    if(size != 0)
    {
        forLoadStruct(lstr);
        lstr++;
    }
    else
    {
        cout << "В программе есть несохранённые структуры. "
              "Загрузка из файла уничтожит несохранённые структуры. "
              "Хотите продолжить? (1 - да) ";
        char choise;
        cin >> choise;
        if (choise == '1')
        {
            firsts = NULL;
            forLoadStruct(lstr);
            lstr++;
        }
    }
}

void outputStruct()
{
    // TODO: сделать функцию загрузки структур, средний бал в которой меньше 4
    // в файл Out.txt

```

```

// NOTE: может придётся исправлять
if(firsts≠NULL)
{
    STUD *currS;
    currS = firsts;
    do
    {
        saveloadstud sa_st[10];
        for(int i = 0; i < 20; i++)
            sa_st[0].name[i] = currS→name[i];
        sa_st[0].year = currS→year;
        sa_st[0].ratingOne = currS→ratingOne;
        sa_st[0].ratingTwo = currS→ratingTwo;
        sa_st[0].ratingThree = currS→ratingOne;
        sa_st[0].ratingFour = currS→ratingOne;
        cout << "\n" << sa_st[0].name << " " << sa_st[0].year << " " <<
sa_st[0].ratingOne << " " << sa_st[0].ratingTwo << " " << sa_st[0].ratingThree <<
" " << sa_st[0].ratingFour;
        FILE *structs = fopen("Out.txt", "wt");
        fflush(structs);
        fwrite(&sa_st, sizeof(sa_st), 25, structs);
        fclose(structs);
        currS = currS→next;
        //delete sa_st;
    }
    while(currS ≠ NULL);
    cout << "\nСтруктуры сохранены!";
}
else
    cout << "\nСтруктуры не обнаружены... ";

```

```

if(firsts ≠ NULL)
{
    STUD *currS;
    currS = firsts;
    do
    {
        int score = (currS→ratingOne + currS→ratingTwo + currS→ratingThree +
currS→ratingFour) / 4.;
        if(score ≤ 4)
        {
            FILE *stud = fopen("Out.txt", "wt");
            fflush(stud);
            fputs("Студент:\n", stud);
            for(int i = 0; i < 50; i++)
                if((currS→name[i] > 64 && currS→name[i] < 91) || (currS-
>name[i] > 96 && currS→name[i] < 123))
                    fputc(currS→name[i], stud);

```

```

        fputs("\nГод рождения: \n", stud);
        char *years = (char*)currS→year;
        fputs(years, stud);
        fputs("\nСредний бал ≤ 4.\n\n", stud);
        fclose(stud);
    }
    currS = currS→next;
}
while(currS ≠ NULL);
cout << "\nСтруктуры сохранены!";
}
else
    cout << "\nСтруктуры не обнаружены... ";
}
*/

void fourScore()
{
    if(firsts ≠ NULL)
    {
        STUD *currS;
        currS = firsts;
        int numStud = 0;
        cout << "Студенты, у кого меньше четырёх баллов:\n";
        do
        {
            float score = (currS→ratingOne + currS→ratingTwo + currS→ratingThree +
currS→ratingFour) / 4.;
            if(score ≤ 4)
            {
                cout << "№" << numStud + 1 << " " << currS→name << ", "
                    << currS→year << " года рождения с баллами: "
                    << currS→ratingOne << ", " << currS→ratingTwo << ", "
                    << currS→ratingThree << ", " << currS→ratingFour
                    << ", средний балл: " << score << ";\n";
            }
            numStud++;
            currS = currS→next;
        }
        while(currS ≠ NULL);
    }
    else
        cout << "\nСтруктуры не обнаружены... ";
}

int main()
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    int choise;
    cout << "\n0 - выход, "
        << "1 - добавить структуру в список, "
        << "2 - удалить структуру со списка, "
        << "3 - вывести информацию, "

```

```

        "4 - сохранить структуры, "
        "5 - загрузить структуры, "
        "6 - отобразить студентов, "
        "7 - у кого меньше 4 баллов.";
firsts = NULL;
for(;;)
{
    cout << "\nВыбор: ";
    cin >> choise;
    if(choise==0) break;
    else if(choise==1) addStruct();
    else if(choise==2) delStruct();
    else if(choise==3) printStruct();
    //else if(choise==4) saveStruct();
    //else if(choise==5) loadStruct();
    //else if(choise==6) outputStruct();
    else if(choise==7) fourScore();
    else cout << "Введено неправильное значение!";
}
return 0;
}

```

Тепер будемо розбиратися із завданням. Я два дні потратив, щоби зрозуміти, що завдання поставлено таке, яке **НЕМОЖЛИВО ВИКОНАТИ!!!** Почнемо з першого. Записати динамічну структуру у файл неможливо. Точніше можливо, але він буде записаний некоректно, так як він має певну адресу на наступну структуру, яку при завантаженні можна було б перезаписати, і тут виявляється, що записана адреса ссилається на тимчасовий файл temp у каталозі C:\user\user\...тощо (повна адреса довга) я вирахувати її розмір неможливо, а перебирати побайтово займе декілька днів, до того ж кожного разу дана адреса буде змінюватись, так як тимчасові файли знищуються при завершенні сеансу роботи операційної системи. Тобто, навіть якщо завантажити цю адресу й переписати, до перед наступним запуском програми необхідно перезавантажувати систему. Тому **зчитування-запис динамічних структур неможливий!** Розглянемо костиль для цього метода: будемо переписувати інформацію крім адреси у статичну структуру. По-перше, навіщо взагалі тоді створювати динамічну структуру, якщо все одно головна робота буде виконуватись зі статичною, а по-друге, як виявилось **програма перестане працювати й вилітає, якщо спробувати записати статичну структуру, у яку було заповнено інформацію з динамічної структури**. Чому так? Незрозуміло. Тому робимо висновок: робота з файлами та робота з динамічними структурами не сумісна. Той же принцип працює і для завдання записати відібрану інформацію до файла Out.txt: **якщо зробити строку з інформацією з динамічної структури чи статичної, заповненої з динамічної, то інформація у файл буде заповнена некоректно — замість імені незалежно від зміста змінної до файла замість імені буде записано "Р", а замість дати народження "С"** (тоді



я ще не намагався вивести бали у файл)! Тому вся робота з файлами була покинута. Замість вивода у файл відібраних студентів із середнім балом нижче 4, я зробив, щоб програма їх виводила просто на екран. Все остальне завдання було виконано: додавання динамічних структур у стек, видалення виведення на екран. Код **ЧЕРВОНИМ** відображає функції, які неможливо реалізувати, а саме: збереження структур на дисці, завантаження у програму та виведення в Out.txt відібраних студентів, середній бал яких менше чотирьох. Чому саме це неможливо — описано вище.

Як працює додавання динамічної структури:

Крок 1: Створення динамічної структури;

Крок 2: Якщо структура перша — встановити показник цієї структури на NULL, а показник на першу структуру встановити на цю структуру;

Крок 3: Інакше — встановити показник на структуру, на яку вказує показник на першу структуру, а сам показник на першу структуру переназначити на дану структуру;

Крок 4: Заповнити покроково всі поля створеної структури;

Як працює видалення останньої структури:

Крок 1: Створити вказівник, який ссилається на ту ж структуру, на яку ссилається й показник на першу структуру;

Крок 2: Переназначити показник на першу структуру на значення адреси, яке лежить в її структурі (на другу);

Крок 3: Видалити показник на першу структуру (тепер можна вважати, що друга стала першою й структуру видалено);

Як працює виведення структур:

Крок 1: Створити показник на поточну структуру, який приймає значення показника на першу структуру;

Крок 2: Вивести позмінно інформацію зі структури;

Крок 3: Перейняти показник адреси з цієї структури (поточною структурою стане наступна);

Крок 4: Повторити 1-3, поки перейнявшийся показник не стане NULL;

Код функції додавання:

```
void addStruct()
{
    STUD *newS = new STUD;
    if(firsts == NULL)
    {
        newS->next = NULL;
        firsts = newS;
    }
    else
    {
        newS->next = firsts;
        firsts = newS;
    }
}
```

```

}
cout << "\nВведите фамилию студента: ";
fflush(stdin);
cin >> newS->name;
cout << "Введите год рождения: ";
cin >> newS->year;
cout << "Введите балл по первому предмету: ";
cin >> newS->ratingOne;
cout << "Введите балл по второму предмету: ";
cin >> newS->ratingTwo;
cout << "Введите балл по третьему предмету: ";
cin >> newS->ratingThree;
cout << "Введите балл по четвёртому предмету: ";
cin >> newS->ratingFour;
}

```

### Код функції видалення:

```

void delStruct()
{
    if(firsts≠NULL)
    {
        STUD *delS;
        delS = firsts;
        firsts = firsts->next;
        delete(delS);
        cout << "\nПоследняя структура удалена!";

    }
    else
        cout << "\nСтруктуры не обнаружены... ";
}

```

### Код функції виведення:

```

void printStruct()
{
    if(firsts≠NULL)
    {
        STUD *currS;
        currS = firsts;
        do
        {
            cout << "\nФамилия и инициалы студента: " << currS->name << "\n";
            cout << "Год рождения: " << currS->year << "\n";
            cout << "Балл по первому предмету: " << currS->ratingOne << "\n";
            cout << "Балл по второму предмету: " << currS->ratingTwo << "\n";
            cout << "Балл по третьему предмету: " << currS->ratingThree << "\n";
            cout << "Балл по четвёртому предмету: " << currS->ratingFour << "\n\n";
            currS = currS->next;
        }
        while(currS ≠ NULL);
    }
    else
        cout << "\nСтруктуры не обнаружены... ";
}

```

Ілюстрація 1. Результат виконання (заповнення)

```
Q:\Projects\C\build-Sem2Zad9-Desktop_x86_windows_msys_pe_64bit-De
0 - выход, 1 - добавить структуру в список, 2 - удалить структуру, 3 - изменить структуру, 4 - вывести структуру, 5 - загрузить структуры, 6 - отобразить структуры
Выбор: 1

Введите фамилию студента: Калиновский
Введите год рождения: 2001
Введите балл по первому предмету: 5
Введите балл по второму предмету: 5
Введите балл по третьему предмету: 5
Введите балл по четвертому предмету: 5

Выбор: 1

Введите фамилию студента: Дудченко
Введите год рождения: 2000
Введите балл по первому предмету: 4
Введите балл по второму предмету: 2
Введите балл по третьему предмету: 4
Введите балл по четвертому предмету: 3

Выбор: 1

Введите фамилию студента: Пономарь
Введите год рождения: 2002
Введите балл по первому предмету: 4
Введите балл по второму предмету: 3
Введите балл по третьему предмету: 3
Введите балл по четвертому предмету: 5
```

Ілюстрація 2. Результат виконання (виведення)

```
Выбор: 3

Фамилия и инициалы студента: Пономарь
Год рождения: 2002
Балл по первому предмету: 4
Балл по второму предмету: 3
Балл по третьему предмету: 3
Балл по четвертому предмету: 5

Фамилия и инициалы студента: Дудченко
Год рождения: 2000
Балл по первому предмету: 4
Балл по второму предмету: 2
Балл по третьему предмету: 4
Балл по четвертому предмету: 3

Фамилия и инициалы студента: Калиновский
Год рождения: 2001
Балл по первому предмету: 5
Балл по второму предмету: 5
Балл по третьему предмету: 5
Балл по четвертому предмету: 5

Выбор: 7
Студенты, у кого меньше четырёх баллов:
№1 Пономарь, 2002 года рождения с баллами: 4, 3, 3, 5, средний балл: 3.75;
№2 Дудченко, 2000 года рождения с баллами: 4, 2, 4, 3, средний балл: 3.25;

Выбор: _
```

*Ілюстрація 3. Результат виконання (виведення після видалення)*

```
Выбор: 2
Последняя структура удалена!
Выбор: 3
фамилия и инициалы студента: Дудченко
Год рождения: 2000
Балл по первому предмету: 4
Балл по второму предмету: 2
Балл по третьему предмету: 4
Балл по четвёртому предмету: 3

фамилия и инициалы студента: Калиновський
Год рождения: 2001
Балл по первому предмету: 5
Балл по второму предмету: 5
Балл по третьему предмету: 5
Балл по четвёртому предмету: 5

Выбор: 4
Введено неправильное значение!
Выбор: 3
фамилия и инициалы студента: Дудченко
Год рождения: 2000
Балл по первому предмету: 4
Балл по второму предмету: 2
Балл по третьему предмету: 4
Балл по четвёртому предмету: 3

фамилия и инициалы студента: Калиновський
Год рождения: 2001
Балл по первому предмету: 5
Балл по второму предмету: 5
Балл по третьему предмету: 5
Балл по четвёртому предмету: 5

Выбор: 7
Студенты, у кого меньше четырёх баллов:
№1 Дудченко, 2000 года рождения с баллами: 4, 2, 4, 3, средний балл: 3.25;
Выбор: _
```

**Результат:** Програма повністю виконує свій функціонал в міру доступних можливостей. Весь функціонал, який можливо реалізувати — реалізовано. Чому не все реалізовано — причини описано вище. *Вважаємо завдання виконаним!* ✓

**Висновок:** програма зчитує інформацію з файла і записує її у файл. Ми отримали знання та навички, необхідні для роботи з даними на основі використання файлів.