

# Звіт

До лабораторної роботи №2

Виконав:

Студент Руденко Альберт ПЗ-16

**Тема.** Документування етапів проектування та розробки

**Мета.** Навчитися документувати основні результати етапу проектування та розробки найпростіших програм

## Теоретичні відомості

24. Функції повинні виноситися у окремі модулі. Вони повинні мати одну точку входу та виходу
34. Вона має інший тип написання програмного коду. Мова С++ є об'єктно орієнтованою мовою, а С функціональною
3. Опис підсистем і компонентів програмної системи, а також зв'язків між ними. Архітектура, в першу чергу, визначає внутрішню структуру системи, задаючи спосіб, яким система буде організована або конструюється.

## Постановка завдання

**Частина І.** У розробленій раніше програмі до лабораторної роботи з дисципліни «Основи програмування» внести зміни – привести її до модульної структури, де модуль – окрема функція-підпрограма. У якості таких функцій запрограмувати алгоритми зчитування та запису у файл, сортування, пошуку, редагування, видалення елементів та решта функцій згідно варіанту.

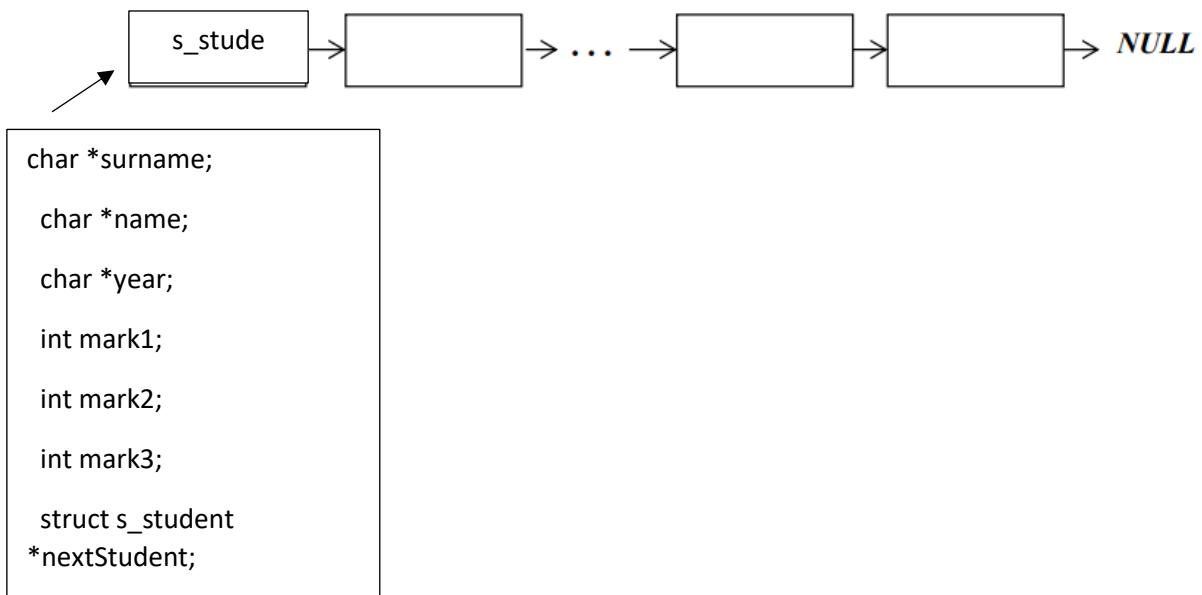
**Частина ІІ.** Сформувати пакет документів до розробленої раніше власної програми:

1. схематичне зображення структур даних, які використовуються для збереження інформації ;
2. блок-схема алгоритмів – основної функції й двох окремих функцій-підпрограм (наприклад, сортування та редагування);
3. текст програми з коментарями та оформленний згідно вище наведених рекомендацій щодо забезпечення читабельності й зрозумілості.

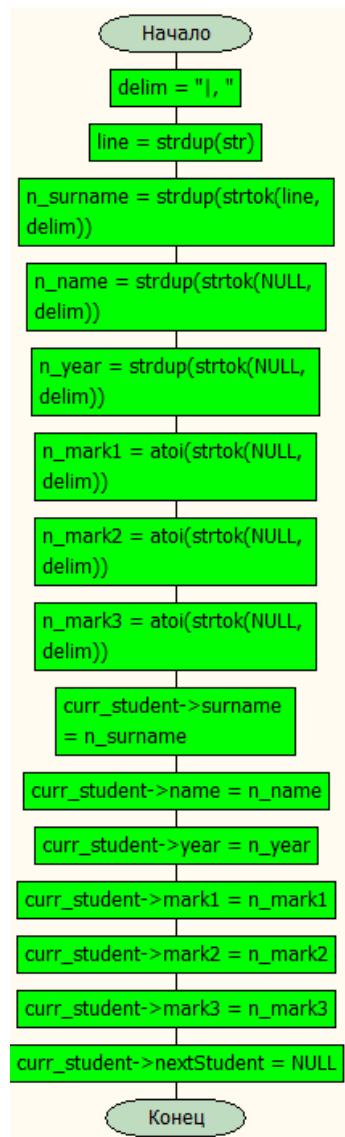
Для схематичного зображення структур даних, блок-схеми алгоритму використати редактор MS-Visio.

**Частина ІІІ.** У редакторі MS-Visio розробити зразки фігур, які були використані для схематичного зображення структур даних програм, як готові трафарети до використання . Сформувати свою бібліотеку фігур – окремий користувачький файл із використаними зразками.

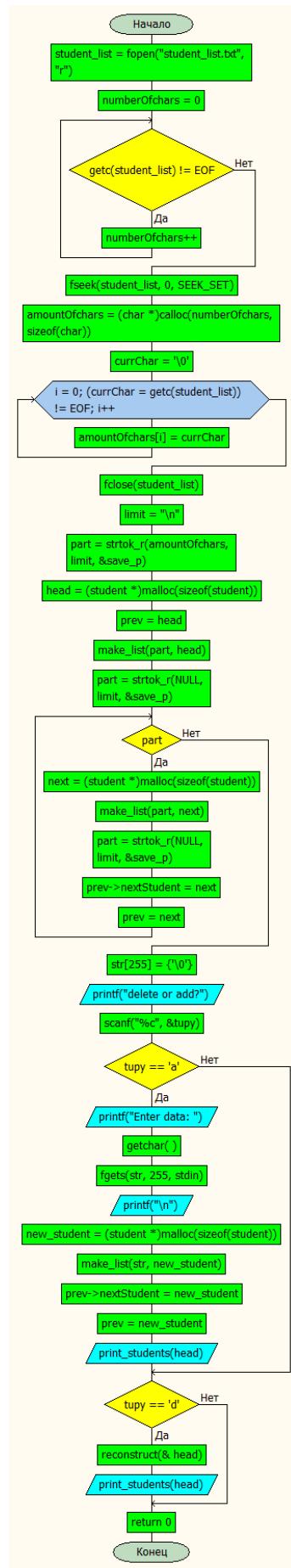
Зв'язний список – одна з найважливіших структур даних, в якій елементи лінійно впорядковані, але порядок визначається не номерами елементів, а вказівниками, які входять в склад елементів списку та вказують на наступний за елемент (в однозв'язних списках (ланцюг), рис. 4) або на наступний та попередній елементи (в двозв'язних списках, рис. 5).



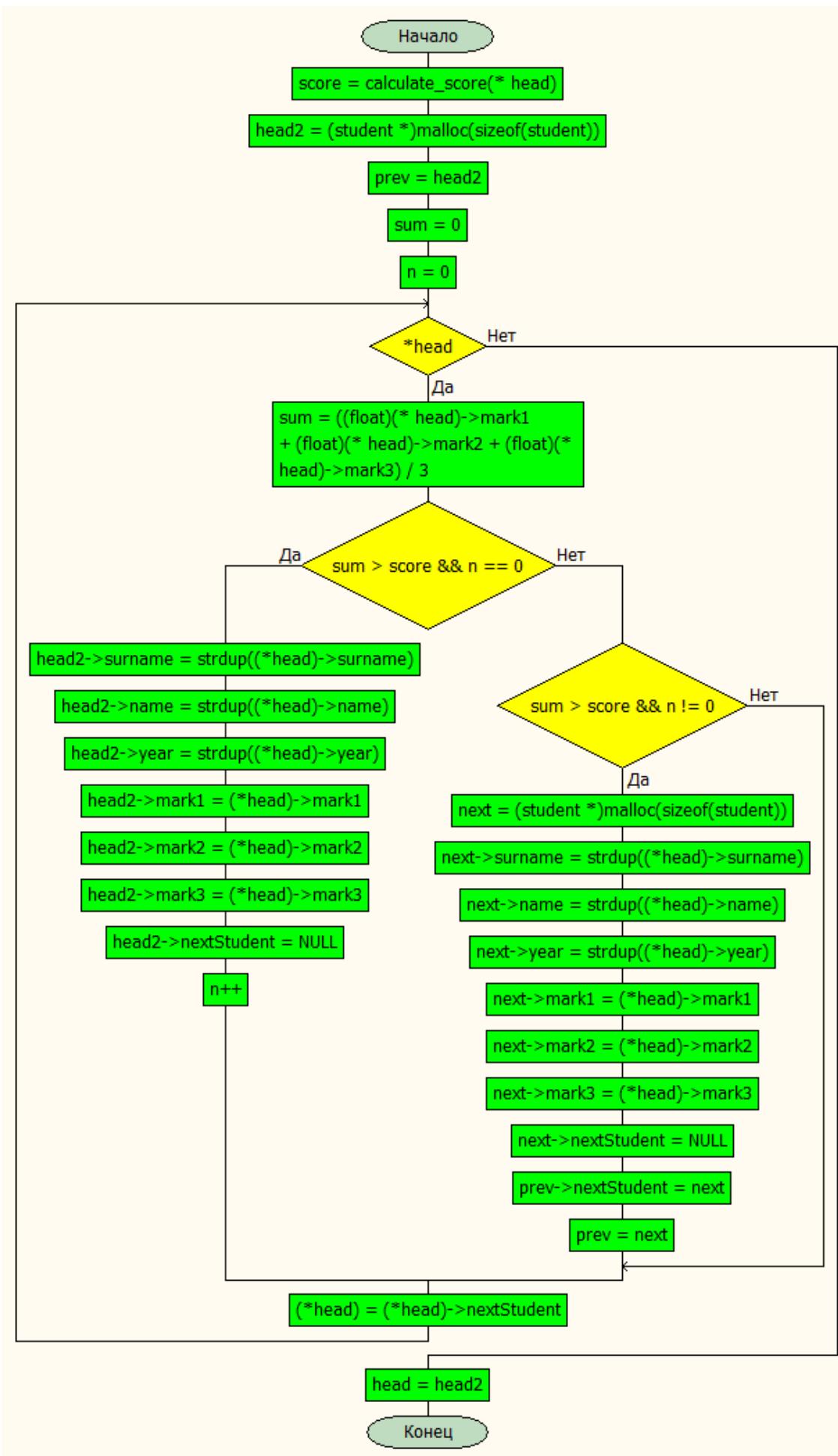
Блох-схема алгоритмів – основної функції й двох окремих функцій-підпрограм (додавання елемента у список та виведення елемента, у якого рейтинговий бал нижчий за середній у групі)



Function1



Main.c



## function4.c

### **header.h**

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>

void print_students(student *head);

< /**
 * Виводить у консоль відсортований список студентів
 */
float calculate_score(student *head);

< /**
 * Підраховує середній бал студентів та зберігає це значення
 */
void make_list(char *str, student *curr_student);

< /**
 * Видаляє зі списку студентів з балом нижче заданого
 */
void reconstruct (student ** head);

< /**
 * Перезбирає список студентів зі змінами
 */

```

### **function1.c**

```
void print_students(student *head)
{
    student *curr_student = head;

    while(curr_student){
        printf("%s %s %s %d %d %d\n", curr_student->surname, curr_student->name, curr_student->year,
curr_student->mark1, curr_student->mark2, curr_student->mark3);

        curr_student = curr_student->nextStudent;
    }
}

//-----
```

### **function2.c**

```
float calculate_score(student *head){

    int numOfs = 0;
    float sum = 0;

    for (student *curr_student = head; curr_student != NULL; curr_student = curr_student->nextStudent)
    {
```

```
    sum += (float)curr_student->mark1 + (float)curr_student->mark2 + (float)curr_student->mark3;
    numOfs++;
}

float score = sum / (numOfs*3);
printf("average score - %lf\n",score);
return score;
}
```

---

**function3.c**

```
void make_list(char *str, student *curr_student) {

    char *delim = "|, ";
    char *line = strdup(str);
    char *n_surname = strdup(strtok(line, delim));
    char *n_name = strdup(strtok(NULL, delim));
    char *n_year = strdup(strtok(NULL, delim));
    int n_mark1 = atoi(strtok(NULL, delim));
    int n_mark2 = atoi(strtok(NULL, delim));
    int n_mark3 = atoi(strtok(NULL, delim));
    curr_student->surname = n_surname;
    curr_student->name = n_name;
    curr_student->year = n_year;
    curr_student->mark1 = n_mark1;
    curr_student->mark2 = n_mark2;
    curr_student->mark3 = n_mark3;
    curr_student->nextStudent = NULL;
}
```

---

**function4.c**

**main.c**

```
#include<header.h>
typedef struct s_student {
    char *surname;
    char *name;
```

```
char *year;
int mark1;
int mark2;
int mark3;
struct s_student *nextStudent;
} student;
int main()
{
FILE* student_list = fopen("student_list.txt","r");
int numberOfchars = 0;
while (getc(student_list) != EOF) {
    numberOfchars++;
}
fseek(student_list, 0, SEEK_SET); //повернення на початок файлу

//записати у динамічний масив
char* amountOfchars = (char*)calloc(numberOfchars,sizeof(char));
char currChar = '\0';
for(int i = 0; (currChar = getc(student_list)) != EOF; i++) {
    amountOfchars[i] = currChar;
}
fclose(student_list);
char *limit = "\n";
char *part;
char *save_p;
part = strtok_r(amountOfchars, limit,&save_p);
student *head = (student*) malloc(sizeof(student));
student *prev = head;
student *next;
make_list(part, head);
part = strtok_r(NULL, limit,&save_p); //заплатка
```

```

while(part) {
    next = (student*) malloc(sizeof(student));
    make_list(part, next);
    part = strtok_r(NULL, limit,&save_p);
    prev->nextStudent = next;
    prev = next;
}

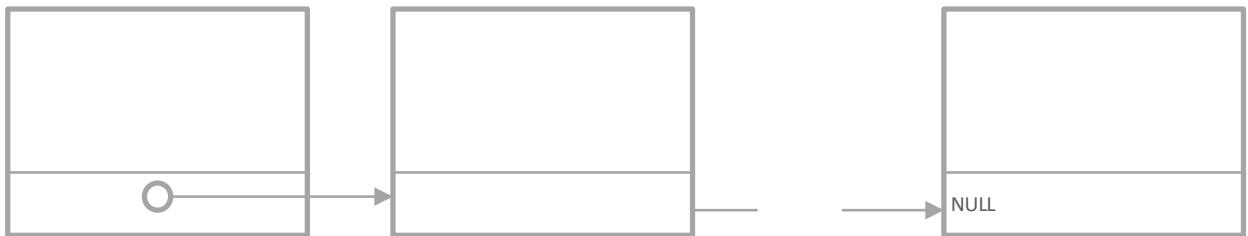
char str[255] ={\0};
printf("delete or add?");
char tupy;
scanf("%c",&tupy);
if(tupy == 'a'){
    printf("Enter data: ");
    getchar();
    fgets(str,255,stdin);
    printf("\n");
    student *new_student = (student*) malloc(sizeof(student));
    make_list(str,new_student);
    prev->nextStudent = new_student;
    prev = new_student;
    print_students(head);
}

if(tupy == 'd'){
    reconstruct(&head);
    print_students(head);
}

return 0;
}

```

Сформований користувацький Visio-файл фігур



## Висновок

Я навчився документувати основні результати етапу проектування та розробки найпростіших програм