**Федеральное государственное автономное образовательное учреждение**

**высшего образования**

**"Национальный исследовательский университет**

**"Высшая школа экономики"**

Московский институт электроники и математики НИУ ВШЭ  
Департамент компьютерной инженерии

**Курс: Информатика и программирование**

**ОТЧЕТ**  
**по лабораторной работе № 6**

|  |  |  |
| --- | --- | --- |
| **Раздел** | **Max оценка** | **Итог.**  **Оценка** |
| Тесты | 1 |  |
| Работа программы | 1 |  |
| Алгоритм | 3 |  |
| Вопросы | 2 |  |
| Доп. Задание | 3 |  |

**Студент: Жалкова Наталия Евгеньевна**  
**Группа: БИВ153, 1 подгруппа**   
**Вариант: 76**  
**Руководитель: Хруслова Диана Владимировна  
Оценка:**

**МОСКВА 2016**

# Содержание

[Содержание 1](#_Toc451802533)

[Условия задач 2](#_Toc451802534)

[Распечатка программы 3](#_Toc451802535)

[Тестовые примеры 7](#_Toc451802536)

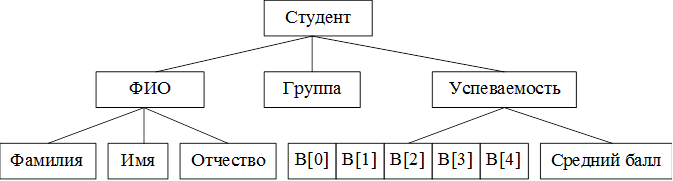
# Условия задач

Ввод данных, обработка списка и вывод результатов оформляются как отдельные функции. Анализ существования результата производится в главной функции. Результат (содержимое полученного списка) выводится в новый текстовый файл.

При выделении и освобождении памяти использовать *new* и *delete*.

При вводе данных и выводе результата использовать *fscanf, fgets и fprintf, fputs*.

I. Написать программу, которая вводит из файла структуры вида:



Массив баллов B[0:4] содержит данные о результатах сдачи экзаменов, по 10-балльной шкале. Каждое поле структуры занимает в файле одну строку, а массив оценок размещается на отдельной строке. Средний балл не записан в файле, а вычисляется в процессе чтения данных. Структуры размещаются **в стеке, реализованном с помощью линейного списка**.

II. Для полученного списка программа выполняет следующие действия:

**вставить новую запись, задаваемую пользователем, перед каждой записью с данными о студенте из указанной группы**

# Распечатка программы

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

//очистка входной строки

#define CLR while (getchar()!='\n')

#define FCLR while (getc(f)!='\n')

//структура студентов,стек

struct Students{

struct

{

char name[80], surname[80], patronymic[80];

} info;

int group;

struct

{

unsigned int B[5];

float mark;

} achieve;

Students \*next;

};

//замена в строке символа '\n' на '\0'

void deln(char \*s)

{ for (; \*s && \*s!='\n'; s++);

\*s='\0';

}

//считывание из файла в стек

Students \*Add (FILE \*f)

{char s[80];

Students \*stack = NULL;

float sum;

while(!(feof(f)))

{ sum=0;

Students \*uk = new Students;//выделение памяти

uk->next = stack;

stack = uk;

fgets(s,80,f);//строка Student 1:

fgets(uk->info.surname,80,f); //Фамилия

deln(uk->info.surname);

fgets(uk->info.name,80,f);//Имя

deln(uk->info.name);

fgets(uk->info.patronymic,80,f);//отчество

deln(uk->info.patronymic);

fscanf(f,"%d", &(uk->group));//группа

if (!feof(f))

FCLR;

for(int i = 0; i<5; i++) //считывание оценок

{

fscanf(f,"%d", &(uk->achieve.B[i]));

sum += uk->achieve.B[i];

}

uk->achieve.mark = sum/5;//средняя оценка

if (!feof(f))

FCLR;

}

return (stack);

}

//вывод в файл

void Print(Students \*catalog, char \*filename)

{FILE \*fl;

int k = 1;

fl = fopen(filename,"w");

fputs("STUDENTS:\r\n",fl);

while(catalog!=NULL)

{ fprintf(fl, "\r\nStudent %d:\r\n", k++);

fprintf(fl, "FIO: "); fprintf(fl, "%s ", catalog->info.surname); fprintf(fl, "%s ", catalog->info.name); fprintf(fl, "%s\n", catalog->info.patronymic);

fprintf(fl, "Group: "); fprintf(fl, "%d \r\n", catalog->group);

fprintf(fl, "Marks: ");

for(int i = 0; i<5; i++)

fprintf(fl, "%d ", catalog->achieve.B[i]);

fprintf(fl, "\r\nAverage Mark: "); fprintf(fl, "%.1f \r\n", catalog->achieve.mark);

catalog = catalog->next;

}

fclose(fl);

}

//получение от пользователя новой записи

Students\* getStudent()

{char readingString[80];

int readingInt;

float sum;

Students \*newStudent = new Students;

printf("Enter surname:\r\n");

scanf("%s",&readingString);

strcpy(newStudent->info.surname,readingString);

printf("Enter name:\r\n");

scanf("%s",&readingString);

strcpy(newStudent->info.name,readingString);

printf("Enter patronymic:\r\n");

scanf("%s",&readingString);

strcpy(newStudent->info.patronymic,readingString);

printf("Enter group:\r\n");

scanf("%d",&readingInt);

newStudent->group = readingInt;

for (int i=0; i<5;i++)

{

printf("Enter mark#%d:\r\n",i);

scanf("%d",&readingInt);

newStudent->achieve.B[i] = readingInt;

sum += newStudent->achieve.B[i];

}

newStudent->achieve.mark = sum/5;

return (newStudent);

}

//копирование данных из одной структуры в другую

void copyStudent(Students \*source, Students \*dest)

{ strcpy(source->info.surname,dest->info.surname);

strcpy(source->info.name,dest->info.name);

strcpy(source->info.patronymic,dest->info.patronymic);

source->group=dest->group;

for (int i = 0; i<5; i++)

source->achieve.B[i]=dest->achieve.B[i];

source->achieve.mark=dest->achieve.mark;

}

/\*вставить новую запись, задаваемую пользователем,

перед каждой записью с данными о студенте из указанной группы\*/

Students\* changes(Students \*catalog, int group)

{Students \*uk = new Students;

Students \*addedStudent = getStudent();

uk=catalog;

while (uk)

{

if (uk->group == group) //проверка на группу

{

Students \*newStudent = new Students;

copyStudent(newStudent, addedStudent);

newStudent->next = uk->next;

uk->next = newStudent;

uk = uk->next;

}

uk = uk->next;

}

return (catalog);

}

int main()

{Students \*catalog = NULL;

FILE \*f;

char filename[20];

int group;

printf("Input file name: \r\n");

gets(filename);

if (!(f = fopen(filename,"r")))

{puts("File not found");

exit(1);

}

catalog = Add(f);

fclose(f);

Print(catalog, "output1.txt");

printf("\r\nEnter group:\r\n");

scanf("%d",&group);

catalog = changes(catalog, group);

Print(catalog, "output2.txt");

return 0;

}

# Тестовые примеры

|  |  |  |
| --- | --- | --- |
| **№** | **Исходные данные** | **Результат** |
| **1** | filename = nunu | File not found |
| **2** |  |  |
| **3** |  |  |