



# ИННОВАЦИОННАЯ ОБРАЗОВАТЕЛЬНАЯ ПРОГРАММА



# Структуры

**Пример. Сведения о студентах группы.**

*Иллюстративный материал к  
лекциям по Информатике*

Автор Саблина Н.Г.

2011 г.





# Содержание



Постановка задачи

Метод решения

Блок-схема алгоритма записи данных

Текст программы ввода данных

Блок-схема алгоритма вычисления среднего  
возраста

Текст программы

Автор





# Постановка задачи.

Сохраним в файле следующие сведения о студентах некоторой группы: фамилию, имя, возраст, рост. Вычислим средний возраст студентов группы и определим самого высокого студента.

## Постановка задачи

- *Исходными данными* для этой программы являются сведения о студентах: фамилия, имя, возраст, рост. Все данные вводятся с клавиатуры.
- *Выходные данные* – файл, содержащий сведения, средний возраст студентов, фамилия, имя, рост самого высокого студента в группе.





# Метод решения.

- Для организации такого списка студентов создадим тип `student`, представляющий собой структуру с соответствующими полями:
  - фамилия,
  - имя - строковые поля;
  - возраст – поле целого положительного типа(см. [1 п. 8.5]),
  - рост в метрах– действительное число.
- Сведения о студентах сохраним в файле `group.dat`. Тип элементов этого файла – `student`.
- Разделим решение этой задачи на две независимые программы:
  - ввод данных о студентах и запись этих данных в файл;
  - вычисление среднего возраста и определение самого высокого студента.





# Метод решения. Задача 1

- Первая программа - диалог ввода исходных данных.
  - Сначала запрашивается количество студентов в группе (n).
  - Затем в цикле вводятся данные о каждом студенте.
- Эти сведения заносятся в поля переменной `person` типа `student`.
- Для обращения к отдельному полю структуры `person` используются составные имена, например,
  - `person.famil`,
  - `person.rost` и т.п.
- Полностью сформированная запись об *i*-м студенте сразу же записывается в файл с помощью одной операции записи.
- Использование структурированного типа `student` делают возможным осуществлять запись в файл всех сведений о студенте за одну операцию записи.



# Метод решения. Задача 2. (1)

- Исходные данные для второй задачи вводятся из файла group.dat.
- Для считывания записей из файла организован цикл с предусловием (while). Этот цикл выполняется до тех пор, пока не будет достигнут конец файла.
- В теле цикла выполняются следующие действия:
  - считывание данных об очередном студенте в переменную person, вывод на экран сведений о нем,
  - добавление его возраста к суммарному возрасту группы,
  - сравнение роста данного студента с ростом других уже просмотренных студентов.

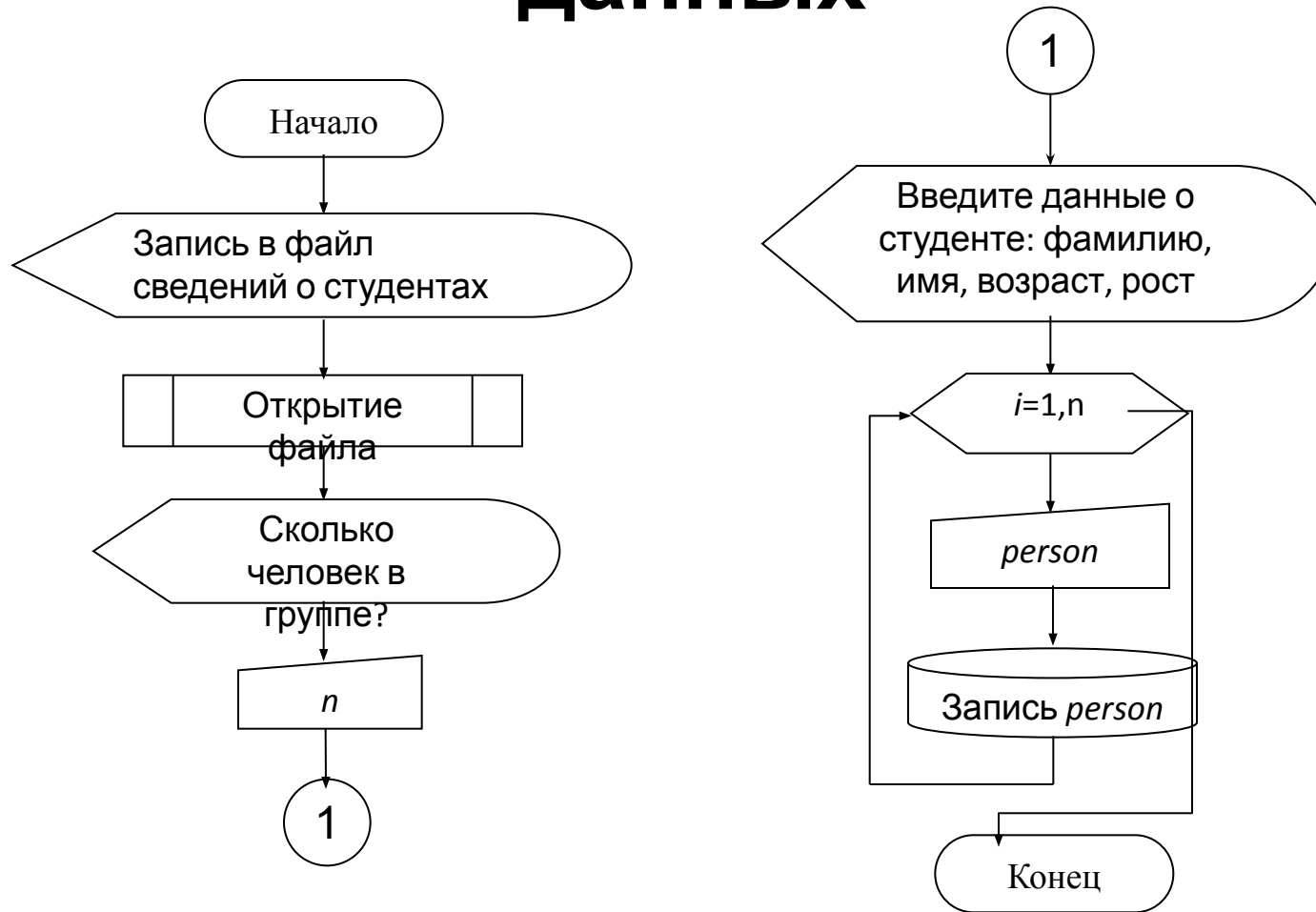


## Метод решения. Задача 2. (2)

- Номер записи  $k$ , содержащей сведения о самом высоком студенте, запоминается. Кроме того, имеется счетчик студентов  $i$ . Считывание записи из файла в переменную `person` осуществляется за одну операцию чтения (так же как и запись), а при выводе на экран нужно обратиться к каждому полю записи отдельно.
- После просмотра всех записей указатель файла переводится на запись с номером  $k$ , производится считывание и вывод на экран сведений о самом высоком студенте в группе.



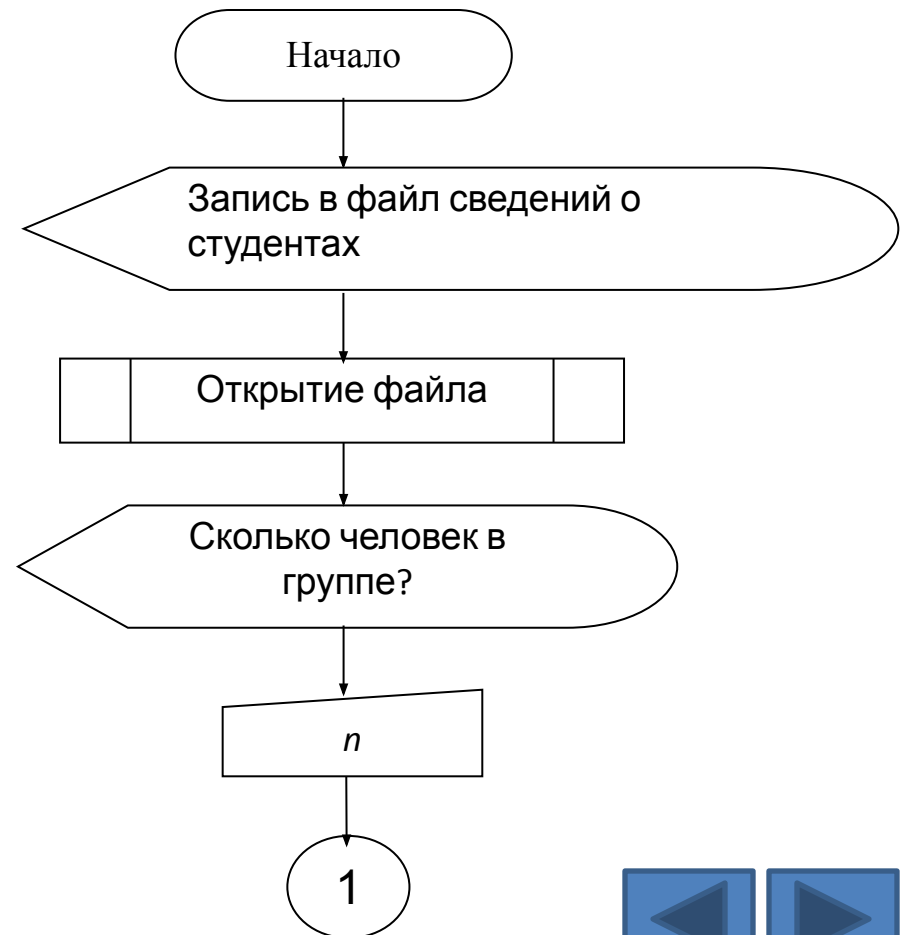
# Блок-схема алгоритма записи данных



# Текст программы ввода данных

```
int main()
{
    struct student{
        char famil[20];
        char name[15];
        unsigned int let;
        float rost;
    };

    int i,n;
    student person;
    FILE *group;
    char* file_name="gruppa.dat";
    group=fopen(file_name,"w");
    cout<<"Сколько человек в группе?";
    cin>>n;
```

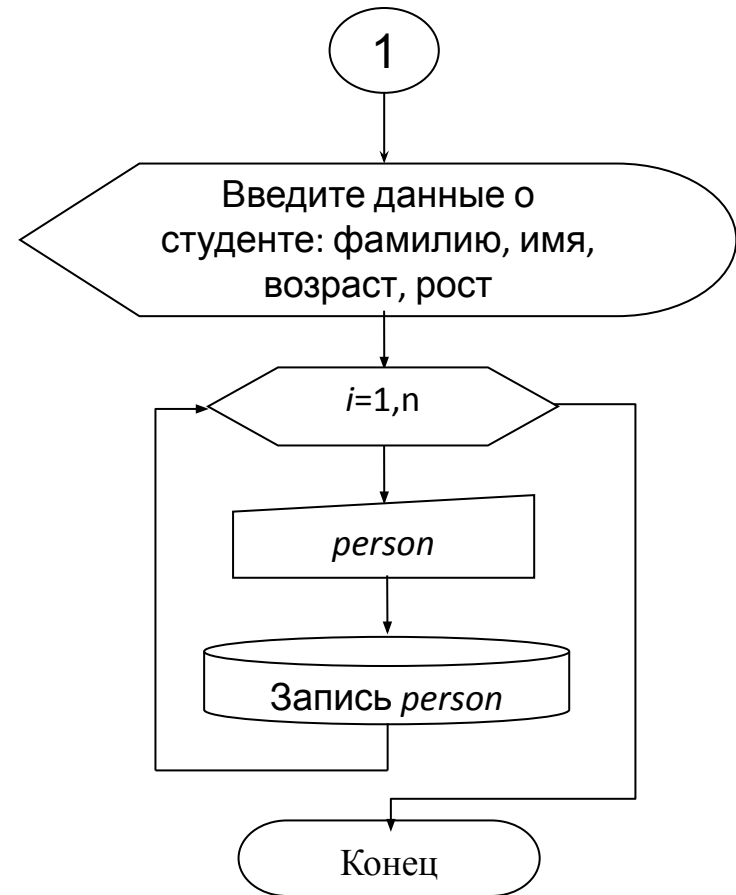


# Текст программы ввода данных

```
cout<<"Введите данные о студнте:";
cout<<"фамилия, имя,возраст, рост";
```

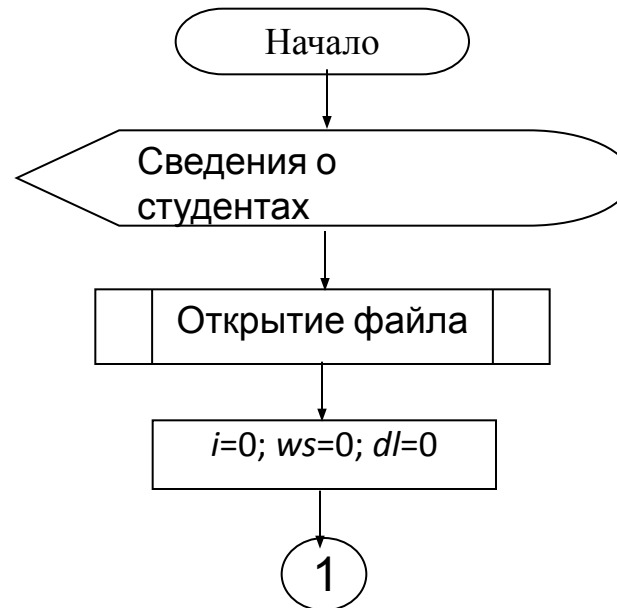
```
for(i=1;i<=n;i++)
{cout<<"\n"<<i<<" ";
cin>>person.famil;
cin>>person.name;
cin>>person.let;
cin>>person.rost;
    fwrite(&person, sizeof(student),1,group);
}

fclose(group);
}
```



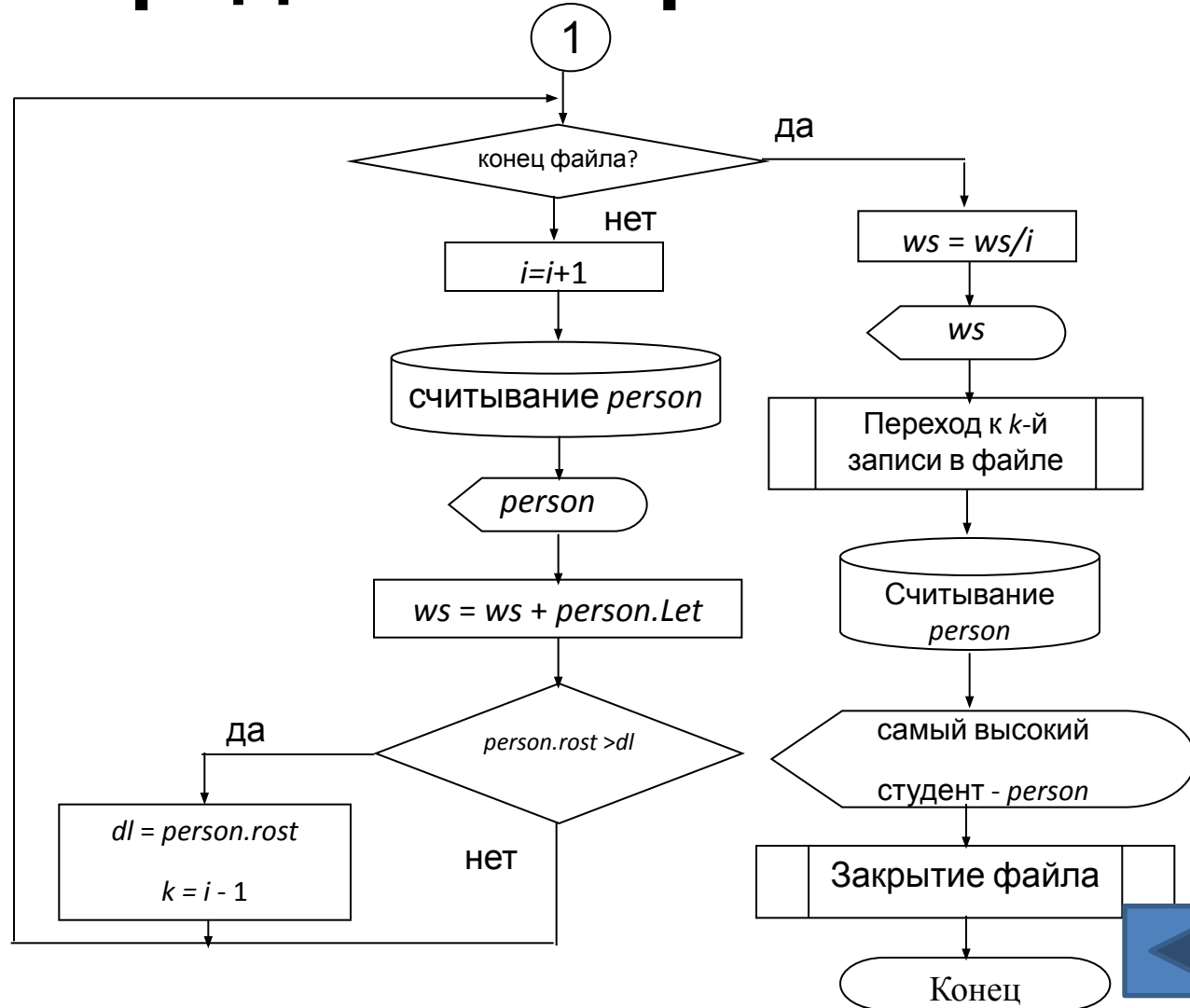


# Блок-схема алгоритма вычисления среднего возраста





# Блок-схема алгоритма вычисления среднего возраста





# Текст программы

```
#include <conio.h>
#include <stdio.h>
#include <io.h>
#include <iostream.h>
int main(){

    struct student{
        char famil[20]; //фамилия
        char name[15];  //имя
        unsigned int let;    //возраст}
        float rost;    // {рост}
    } ;
```





# Текст программы

```
student person;  
    FILE *group;  
    int i,k,ws;  
    float dl;  
cout<<"\n Сведения о студентах группы";  
//открытие файла для чтения  
group=fopen("gruppa.dat","r");  
i=0; dl=0; ws=0;
```





# Текст программы

```
while (!feof(group)) // достигнут конец файла
{
// считывание записи из файла
fread(&person,sizeof(person),1,group);
if (feof(group)) break;
++i; //счетчик студентов
ws=ws+person.let; //суммарный возраст
if (person.rost>dl)
    { dl=person.rost;
      k=i-1;
    } //вывод на экран
cout<<"\n"<<i<<) "<<person.famil<<" "<<person.name;
cout<<" , vozrast- "<<person.let;
cout<<" let, rost- "<<person.rost<<" м.";
}
```







# Текст программы

```
float sr=ws/(i);  
cout<<"\n Средний возраст- "<<sr<<" лет";  
long p=k*sizeof(student)+1;  
fseek (group,p,0);  
fread(&person,sizeof(student),1,group);  
cout<<"\n Самый высокий в группе - "<<person.famil<<"  
    "<<person.name;  
cout<<". Его рост- "<<person.rost<<" м.";  
fclose(group); //    {заккрытие файла}  
getch();  
return 0;
```





Автор:

аблина Наталья Григорьевна

Ст. преподаватель

каф. РТС УГТУ-УПИ

