

Хранение связанной информации в структурах

ОСНОВНЫЕ КОНЦЕПЦИИ

- Структуры позволяют вашим программам группировать в одной переменной связанные данные, типы которых могут различаться.
- Структура состоит из одной или нескольких частей данных, называемых элементами.
- Для определения структуры внутри программы следует указать имя структуры и ее элементы.
- Каждый элемент структуры имеет тип, например *char*, *int* и *float*, и имя каждого элемента должно быть уникальным.
- После того как ваша программа определит структуру, она может объявить переменные типа этой структуры.
- Для изменения элементов структуры внутри функции ваши программы должны передать структуру в функцию с помощью адреса.
- Структура определяет шаблон, с помощью которого ваша программа может позднее объявить одну или несколько переменных.

- Структура объявляется следующим образом:

struct имя структуры

{

поля структуры

}

- В данном случае определение не объявляет какие-либо переменные типа этой структуры. После того как вы определите структуру, ваша программа может объявить переменные типа этой структуры, используя имя структуры (иногда называемое *структурным тэгом*).
- Структура позволяет вашим программам группировать информацию, называемую элементами, в одной переменной. Чтобы присвоить значение элементу или обратиться к значению элемента, используйте оператор C++ *точку* (.).

СТРУКТУРЫ И ФУНКЦИИ

- Если функция не изменяет структуру, вы можете передать структуру в функцию по имени.
- Как вы знаете, если функция изменяет параметр, вам следует передавать этот параметр в функцию с помощью адреса. Если функция изменяет элемент структуры, вы должны передавать эту структуру в функцию с помощью адреса. Для передачи переменной типа структуры с помощью адреса вы просто предваряете имя переменной оператором адреса C++ (&), как показано ниже:

```
some_function(&worker);
```

- Внутри функции, которая изменяет один или несколько элементов, вам следует работать с указателем. Если вы используете указатель на структуру, легче всего обращаться к элементам структуры, используя следующий синтаксис:

```
pointer_variable->member = some_value;
```

Пример.

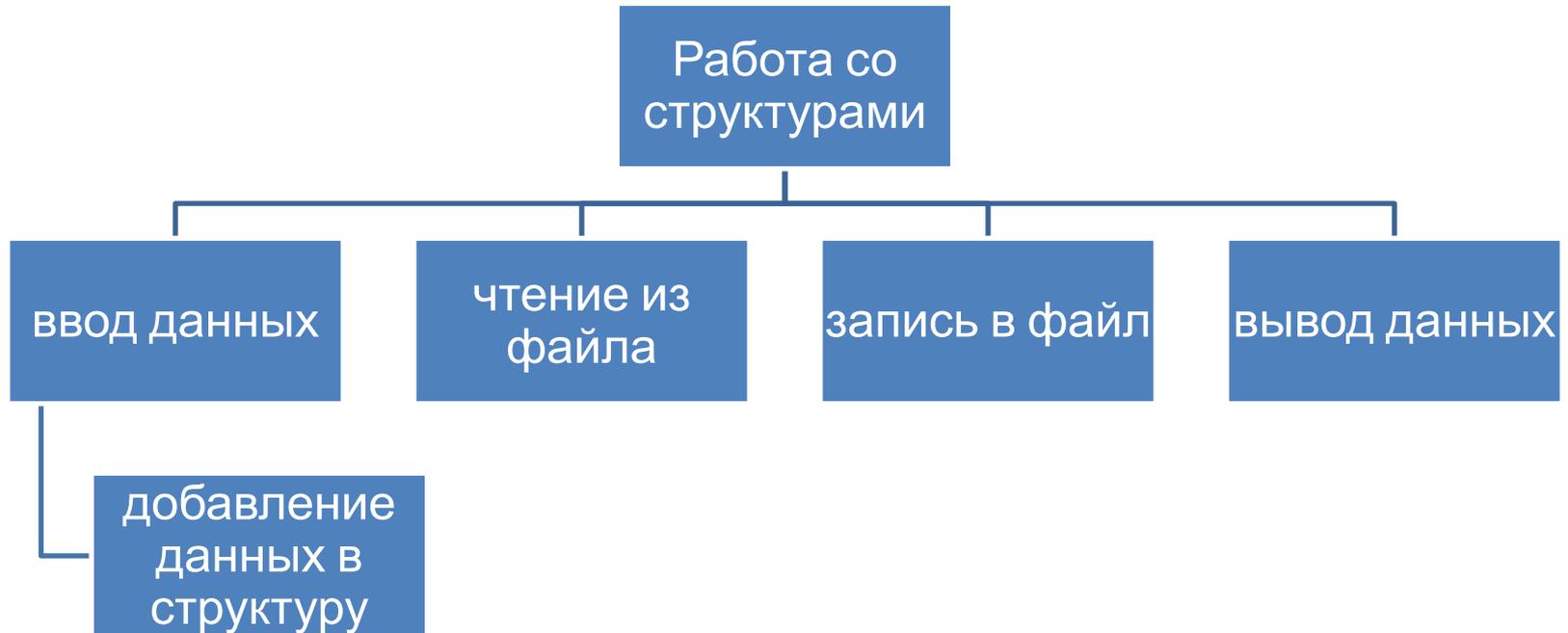
Задание. Разработать программу со следующими функциями:

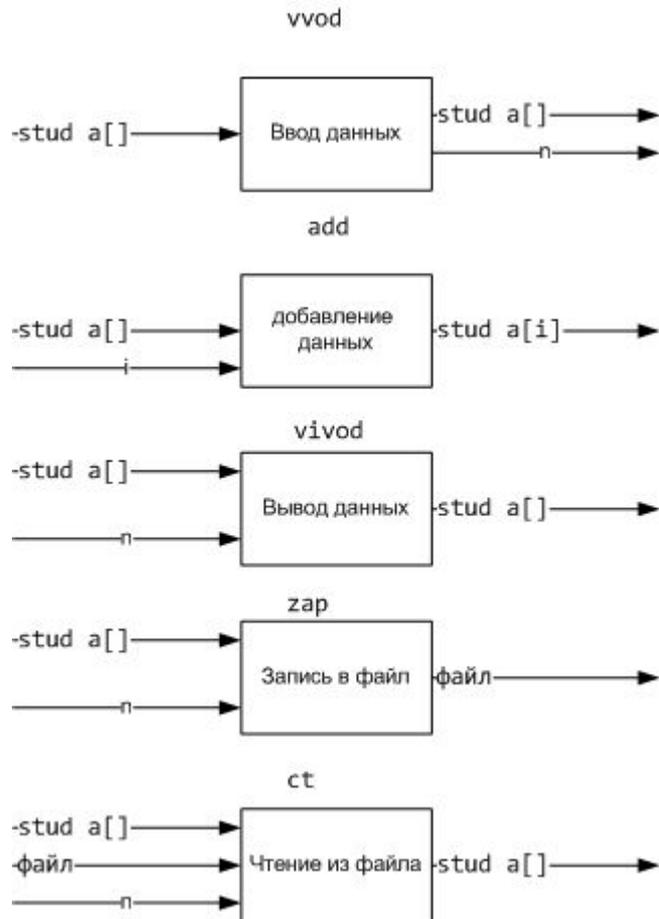
- ввод данных структуры;
- сохранения данных в файл;
- чтение данных из файла;
- вывод на экран таблицы с данными;

Справочник пропусков студентов.

- ФИО студента.
- Группа.
- Количество пропусков.
- Количество неаттестаций.

| ФИО | Группа | Количество пропусков | Количество неаттестаций |
|--------|--------|----------------------|-------------------------|
| строка | строка | целое | целое |





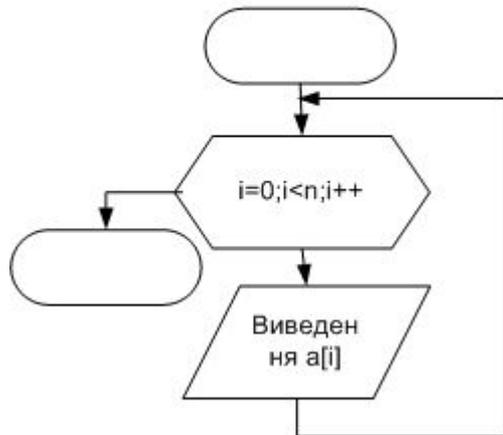
| Вхід | Дії | Вихід |
|-------------------------|--|---|
| а – структура типа stud | 1. ex=1 2. Пока ex=1 2.1. добавление i-го элемента в структуру 2.2. i++ 3. n=i | а – структура типа stud n – количество элементов в структуре |

| Вхід | Дії | Вихід |
|--------------------------------------|---|------------------------------------|
| а – структура типа stud i - целое | ex=1 Пока ex=1 добавление i-го элемента в структуру | а[i] – элемент структуры типа stud |

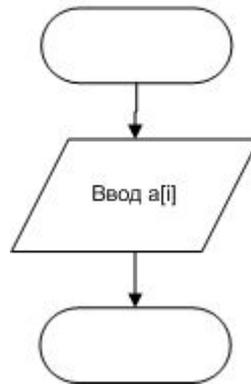
| Вхід | Дії | Вихід |
|---|--------------------------------|----------|
| а – структура типа stud n – количество элементов в структуре | i=0..n Запись данных в файл | f – файл |

| Вхід | Дії | Вихід |
|---|---------------------------------------|----------------------------|
| а – структура типа stud n – количество элементов в структуре f – файл | $i=0..n$ Чтение данных из файла | а – структура типа stud |

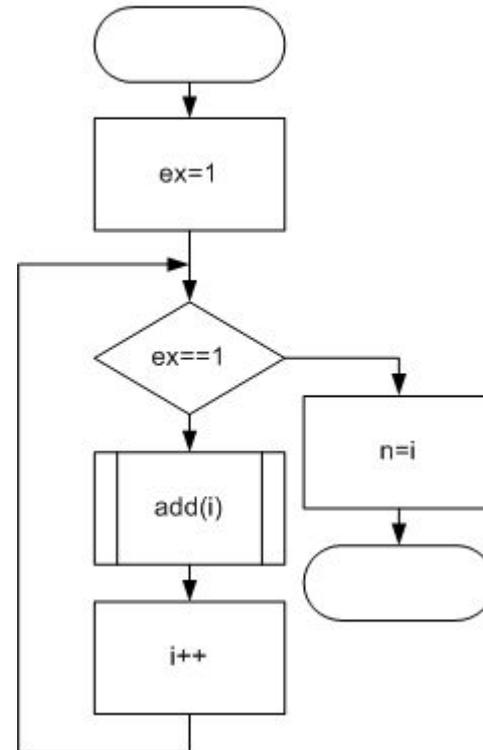
void vivod()

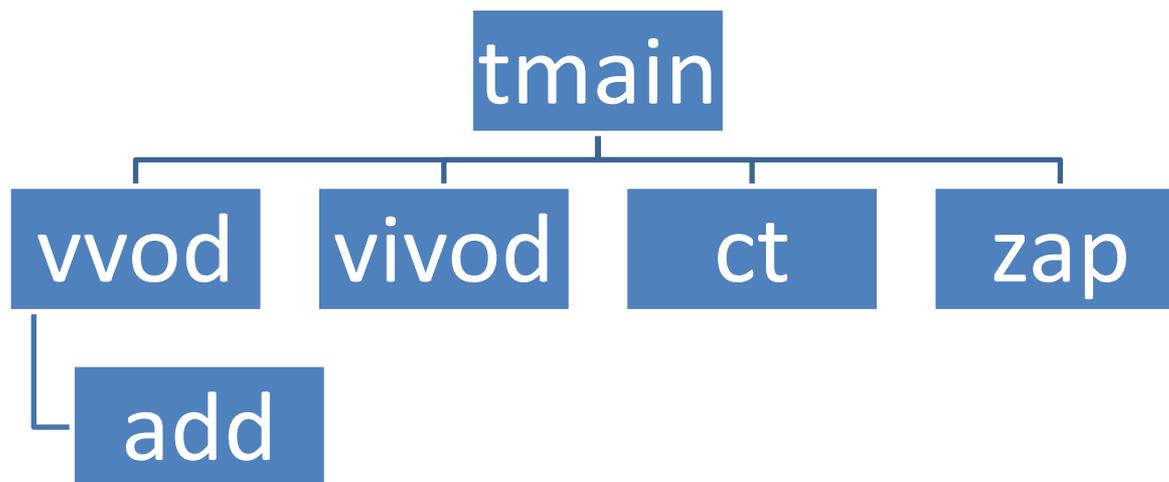
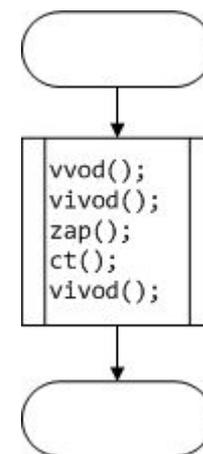
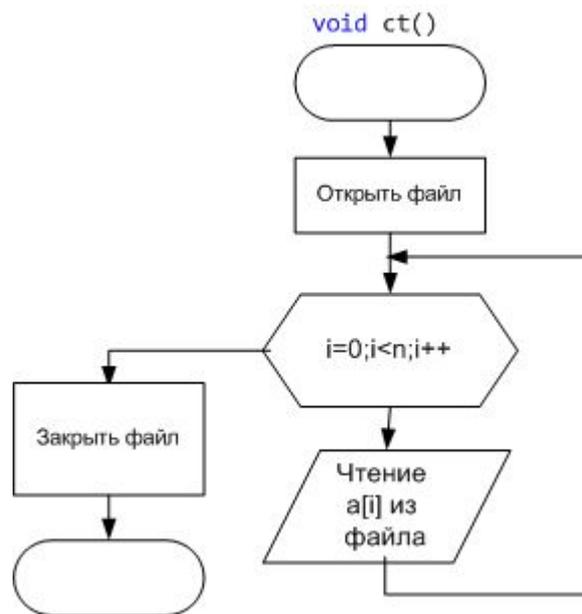
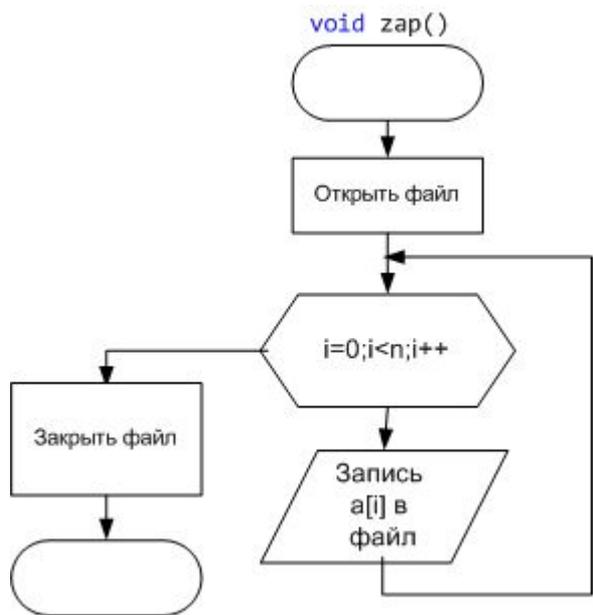


void add (int i)



void vvod()





```

// prim1_lek8.cpp: определяет точку входа для консольного приложения.
//
#include "stdafx.h"
#include "stdafx.h"
#include<iostream>
#include<fstream>
#include<stdio.h>
#include<conio.h>
#include <iomanip>
using namespace std;
struct stud
{
    char fio[25];
    char gruppa[25];
    int kol_p;
    int kol_na;
};
stud a[100];
int n=0;
void vivod()
{
    for (int i=0;i<n;i++)
        cout<<setw(4)<<a[i].fio<<"|"<<setw(4)<<a[i].gruppa<<"|"<<setw(2)<<a[i].kol_p<<"|"<<setw(2)<<a[i].kol_na<<endl;
}
void add (int i)
{
    cout<<"FIO=";
    cin>>a[i].fio;
    cout<<"Gruppa=";
    cin>>a[i].gruppa;
    cout<<"Kol-vo prop.=";
    cin>>a[i].kol_p;
    cout<<"kol-vo ne at.=";
    cin>>a[i].kol_na;
}
}

```

```
void vvod()
{int i=0;
  int ex=1;

  while (ex==1)
  {
    cout<<"Vvedite zapis"<<endl;

    add(i);
    i++;
    cout<<"Continue? YES-1 No-0";
    cin>>ex;
  }
  n=i;
}
void zap()
{
  const char *FName="c:\\1.txt"; //Путь к файлу. Вписывайте свой.
  // открываем файл для записи
  ofstream f(FName,ios::binary|ios::out);
  for (int i=0;i<n;i++)
    f.write((char*)&a[i],sizeof a[i]); //Записали всю структуру целиком
  f.close();
}
```

```
void ct()
{
    const char *FName="c:\\1.txt"; //Путь к файлу. Вписывайте свой.
    //открываем файл для чтения
    ifstream in(FName,ios::binary|ios::in);
    for (int i=0;i<n;i++)
        in.read((char*)&a[i],sizeof a[i]); //Читаем структуру целиком сразу
    in.close();
}
int _tmain(int argc, _TCHAR* argv[])
{
    vvod();
    vivod();
    zap();
    ct();
    vivod();
    _getch();
    return 0;
}
```

```

Uvedite zapis
FIO=Ivanov
Gruppa=PNK-21
Kol-vo prop.=10
kol-vo ne at.=4
Continue? YES-1 No-01
Uvedite zapis
FIO=Petrov
Gruppa=PNK-21
Kol-vo prop.=8
kol-vo ne at.=8
Continue? YES-1 No-01
Uvedite zapis
FIO=Sidorov
Gruppa=PNK-21
Kol-vo prop.=1
kol-vo ne at.=1
Continue? YES-1 No-00
Ivanov!PNK-21!10! 4
Petrov!PNK-21! 8! 8
Sidorov!PNK-21! 1! 1
Ivanov!PNK-21!10! 4
Petrov!PNK-21! 8! 8
Sidorov!PNK-21! 1! 1

```

```

IvanovBBBBBBBBBBBBBBBBBBBBPNK-21BBBBBBBBBBBBBBBBBBBBPetrovBBBBBBBBB
BBBBBBBBBBBBPNK-21BBBBBBBBBBBBBBBBBBBBSSidorovBBBBBBBBBBBBBBBBPNK-2
1BBBBBBBBBBBBBBBBBBBB

```