Структуры.

Структурой называется совокупность логически связанных переменных различных типов, сгруппированных под одним именем для удобства дальнейшей обработки.

Структура – тип данных задаваемый пользователем.

Этапы работы со структурами:

1. **объявление и определение типа структуры**
2. **объявление структурной переменной**
3. **инициализация структурной переменной**
4. **использование структурной переменной**

Определение типа структуры

```
struct ID
{
    <тип> <имя 1-го элемента>;
    <тип> <имя 2-го элемента>;
    .....
    <тип> <имя последнего элемента>;
};
```

Объявление структурной переменной:

1. struct ID var1;
2. ID var1; // служебное слово struct при объявлении структурной переменной можно не использовать

Здесь ID – пользовательский тип данных

var1 – имя структурной переменной

Пример объявления структурной переменной

1.

```
struct list
      { char name[20]; //поле структуры
        char first_name[40]; // поле структуры
        int age;  // поле структуры
      };
struct list student;
```

2.

```
struct list
      { char name[20];
        char first_name[40];
        int age;
      } student;
```

Работа со струтурной переменной обычно сводится к работе с отдельными полями структуры.

Доступ к полю структуры осуществляется с помощью операции. (точка) посредством конструкции вида :

имя_структуры . имя_поля_структуры

Например, нашу структурную переменную можно определить следующим образом:

struct list student={"Ivanov", "Petr",1995};

Далее значение полей структуры можно вывести на экран монитора
cout<<student.name<<” ‘’<< student.first_name<<” ‘’student.age;

Массивы структур.

Структуры часто образуют массивы. Чтобы объявить массив структур, вначале необходимо определить структуру (то есть тип данных), а затем объявить переменную массива этого же типа. Например, чтобы объявить 25-элементный массив структур типа list, который был определен ранее, напишим следующее:

struct list student[25];

Здесь student – имя массива
list- тип данных элементов массива
Обращение к каждому полю структуры осуществляется по имени массива и по имени поля

```
for (int i=0; i<3; i++)
    {cin>>student[i].name;
    cin>>student[i].first_name;
    }
```

Приведем пример програмы

В группе 20 студентов. Они сдавали экзамены по физике и математике. Надо ввести данные о студентах с клавиатуры, сосчитать средний балл каждого студента и вывести на экран имя студента с минимальным баллом и сам балл

```
#include <iostream>
using namespace std;

struct student {       // описание пользовательского типа
struct student
    char name[10];
    char first_name[20];
    int fisic;
    int matematic;
};

int main() {
    student exsam [20];  // объявление массива exsam
    float sr_ball[20];   // объявление массива sr_ball
    for (int i = 0; i < 20; i++) {
        cin >> exsam[i].name;   // ввод имени студента
        cin >> exsam[i].first_name;  // ввод  фамилии
студента
        cin >> exsam[i].fisic;  // оценка по физике
        cin >> exsam[i].matematic; // оценка по математике
        sr_ball[i] = (exsam[i].fisic + exsam[i].matematic) /
2.; //ср. балл
    }
    for (int i = 0; i < 20; i++)
        cout << exsam[i].name << "  " <<
exsam[i].first_name << "   " << exsam[i].fisic << "  " <<
```

```
exsam[i].matematic << endl; //вывод значений элементов массивов на экран монитора

    for (int i = 0; i < 20; i++)
        cout << sr_ball[i] << endl; //вывод ср. баллов всех студентов

     float min = 25;
     int k = 0;
     for (int j = 0; j < 20; j++)
           if (sr_ball[j] < min) {
               min = sr_ball[j];
               k = j; //номер эл-та массива с минимальным средним баллом
           }
       cout <<exsam[ k].name << "  "<< sr_ball[k] << "  ";
    return 0;}
```