## Зачем нужны структуры

Использование структур в простых программах не делают процесс программирования более удобным. Программа, в некоторых случаях, становится более понятна, но все же достаточно громоздка.

Структуры больше всего нужны при работе с функциями. Структурные переменные (объекты) и указатели на них используются как атрибуты функций и как возвращаемые значения функций.

**Указатель как атрибут структуры (объекта)**

Пусть имеется двое часов, которые показывают дату, час и минуты. Показания часов и минут могут различаться, а дата на часах не зависит от часов и минут и должна быть одинаковой.

Для описания подобной ситуации разумно использовать указатель на переменную с датой.

// Начало мантры

#include <stdio.h>

#include <stdlib.h>

#include <math.h> // Конец мантры

typedef struct ClockFace {

char min; // минуты

char h; // часы

// этот атрибут будет иметь смысл только если ему

// передать адрес существующей переменной для даты

int \*dayOfmonth; // адрес переменной день месяца

} Ctime;

int main(){

//объявляем переменные типа Ctime

Ctime timer1, timer2; // объявляем переменную для даты

int date = 21; // Присвоим значения атрибутам timer1

timer1.h = 11; timer1.min = 10; // присваиваем адрес переменной date // до этого реального адреса в dayOfmonth не было

timer1.dayOfmonth = &date; // Присвоим значения атрибутам timer1 timer2.h = 3; timer2.min = 45; // присваиваем адрес переменной date

// до этого реального адреса в dayOfmonth не было

timer2.dayOfmonth = &date; // изменим значение даты и напечатаем результат

date++; // печать

// \*(timer1.dayOfmonth) - значение по адресу timer1.dayOfmonth printf("timer1: %02d %02hhd:%02hhd\n", \*(timer1.dayOfmonth), timer1.h, timer1.min);

printf("timer2: %02d %02hhd:%02hhd\n", \*(timer2.dayOfmonth), timer2.h, timer2.min);

return 0;

}

## Простая передача функции атрибута-объекта (структуры)

### FS3. Функция печати даты.

Дата представлена как структура:

typedef struct Dt

{

unsigned char day;

unsigned char month;

unsigned int year;

} Date;

Написать функцию void printDate(Date day);, которая печатает время в привычном нам формате yyyy-mm-hh.

Решение.

#include <stdio.h>

#include <stdlib.h>

// Допустим описана подобная структура

typedef struct Termin

{

unsigned char day; // день месяца

unsigned char month; // месяц года

unsigned int year; // год

} Date;

// Функция печати даты

// функция принимает объект struct Termin

// вместо struct Termin можно использовать

// синоним Date

void printDate(struct Termin a){

// при вызове функции с параметром, значения всех

// полей копируется в соответствующие поля объекта a

printf("%04u-%02hhu-%02hhu\n", a.year, a.month, a.day);

};

int main(){

Date dt; // объект дата

// получение значений атрибутов с консоли

scanf("%u%hhu%hhu", &(dt.year), &(dt.month), &(dt.day));

// вызов функции printDate(), параметр функции dt

printDate(dt);

return 0;

}

Передача указателя на объект (структуру)

FS3. Функция получения даты с консоли.

Дата представлена как структура:

typedef struct Dt

{

unsigned char day;

unsigned char month;

unsigned int year;

} Date;

Написать функцию void getDate(Date \* day);, которая считывает время в привычном нам формате yyyy-mm-hh.

Решение.

#include <stdio.h>

#include <stdlib.h>

// Допустим описана подобная структура

typedef struct Termin

{

unsigned char day; // день месяца

unsigned char month; // месяц года

unsigned int year; // год

} Date;

// Функция печати даты

// функция принимает объект struct Termin

// вместо struct Termin можно использовать

// синоним Date

void printDate(struct Termin day){

// при вызове функции с параметром, значения всех

// полей копируется в соответствующие поля объекта a

printf("%04u-%02hhu-%02hhu\n", day.year, day.month, day.day);

};

// Функция получения даты

// функция принимает указатель на объект типа Date

// Это позволит изменять значения переменной по адресу

void getDate(Date\* a){

// при вызове функции с параметром указателем

// в переменную-указатель копируется адрес объекта

// Таким образом scanf работает с адресом

// a->year - получение ЗНАЧЕНИЯ атрибута ПО АДРЕСУ объекта

// &(a->year) - получение адреса атрибута,

// можно писать &a->year, приоритет оператора -> выше, чем у &

scanf("%u%hhu%hhu", &(a->year), &(a->month), &(a->day));

};

int main(){

Date dt; // объект дата

// получение значений атрибутов с консоли

// парметр - адрес dt

// значение dt в main() меняется функцией getDate()

getDate(&dt);

// вызов функции printDate(), параметр функции - dt

// значение dt копируется в локальную переменную функции

// значение dt в main() не меняется

printDate(dt);

return 0;

}