

# Битовые поля

```
#define HIDE      0
#define SHOW     1
#define BORDER   1
#define CAPTION  2
#define RED       1
#define GREEN    2
#define BLUE     4
```

```
struct window
{
    ...
    unsigned char flags;
};
```

Пусть в поле flags

бит 0 — видимо/скрыто (SHOW/HIDE)

биты 1 – 2 — стиль окна (BORDER, CAPTION)

биты 3 – 5 — цвет окна (RED, GREEN, BLUE)

```
struct window w;
unsigned char part;

w.flags = SHOW | (BORDER << 1) | ((RED | BLUE) << 3);

part = (w.flags >> 1) & 0x3;
if (part & BORDER)
    printf("BORDER is present\n");
else
    printf("BORDER is not present\n");

if ((w.flags >> 3) & GREEN)
    printf("GREEN is present\n");
else
    printf("GREEN is not present\n");

if ((w.flags >> 3) & BLUE)
    printf("BLUE is present\n");
else
    printf("BLUE is not present\n");
```

# Битовые поля

*Битовое поле* - особый тип структуры, определяющей, какую длину имеет каждый член в битах.

Стандартный вид объявления битовых полей следующий:

```
struct имя_структуры
{
    тип имя1: длина;
    тип имя2: длина;
    ...
    тип имяN: длина;
};
```

Битовые поля должны объявляться как целые, unsigned или signed.

```
struct wnd_flags
{
    unsigned char show    : 1;
    unsigned char style   : 2;
    unsigned char color   : 3;
};
```

```
struct window
{
    struct wnd_flags flags;
};
```

...

```
w.flags = SHOW | (BORDER << 1) | ((RED | BLUE) << 3);
```

```
w.flags.show    = SHOW;
w.flags.style   = BORDER;
w.flags.color   = RED | BLUE;
```

```
part = (w.flags >> 1) & 0x3;
if (part & BORDER)
    printf("BORDER is present\n");
else
    printf("BORDER is not present\n");
if (w.flags.style & BORDER)
    printf("BORDER is present\n");
else
    printf("BORDER is not present\n");

if ((w.flags >> 3) & GREEN)
    printf("GREEN is present\n");
else
    printf("GREEN is not present\n");
if (w.flags.color & GREEN)
    printf("GREEN is present\n");
else
    printf("GREEN is not present\n");
```

# Битовые поля

```
w.flags = 5; // ошибка компиляции
```

```
unsigned char f;
```

```
...
```

```
f = w.flags; // ошибка компиляции
```

```
struct window
```

```
{
```

```
    // другие обычные поля
```

```
    unsigned char show : 1;
```

```
    unsigned char style : 2;
```

```
    unsigned char color : 3;
```

```
};
```