

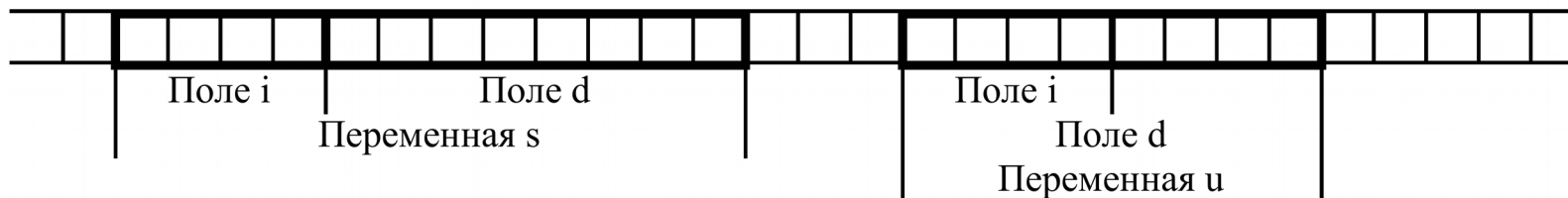
Объединения (введение).

Объединения

Объединение, как и структура, содержит одно или несколько полей различного типа. Однако все поля объединения разделяют одну и ту же область памяти.

```
struct
{
    int i;
    double d;
} s;
```

```
union
{
    int i;
    double d;
} u;
```



Объединения

Присвоение значения одному члену объединения изменит значение других членов.

```
printf("u.i %d, u.d %g\n", u.i, u.d);  
// u.i 2293664, u.d 1.7926e-307
```

```
u.i = 5;  
printf("u.i %d, u.d %g\n", u.i, u.d);  
// u.i 5, u.d 1.79255e-307
```

```
u.d = 5.25;  
printf("u.i %d, u.d %g\n", u.i, u.d);  
// u.i 0, u.d 5.25
```

Инициализация объединений

```
struct s_t  
{  
    int i;  
    double d;  
};
```

...

```
struct s_t s = {1, 5.25};
```

```
union u_t  
{  
    int i;  
    double d;  
};
```

...

```
union u_t u_1 = {1};
```

// только c99

```
union u_t u_2 = { .d = 5.25 };
```

Использование объединений

```
#define NAME_LEN      20
#define TITLE_LEN     20
#define PUBLISHER_LEN 10

struct book_t {
    char author[NAME_LEN+1];
    char title[TITLE_LEN+1];
    char publisher[PUBLISHER_LEN+1];
    int  year;
};

struct magazine_t {
    char title[TITLE_LEN+1];
    int  year;
    int  volume;
};
```

```
union item_t {
    struct book_t      book;
    struct magazine_t  magazine;
};

typedef enum
    {KIND_BOOK, KIND_MAGAZINE}
    kind_item_t;

struct library_item_t {
    int          number;
    kind_item_t  kind;
    union item_t item;
};
```