



ВЫСШАЯ ШКОЛА ЭКОНОМИКИ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

Департамент программной инженерии
Алгоритмы и структуры данных - 2

Семинар 2.2. Структура (тип) Union

2020-2021

Union - Объединение

- **Union** – «структура данных» для размещения нескольких переменных любого типа по одному адресу.
- Элементы union – поля.
- Обращение к полям – через точку

<Имя переменной>.<Имя поля>

Описание

```
union <имя_типа>
{
    <тип1> <имя_переменной1>;
    <тип2> <имя_переменной2>;
    ...
    <типN> <имя_переменнойN>;
};
<имя_типа> <имя_переменной>;
```

Размещение в памяти

По одному адресу, выравнивание по нулевому байту

➤ Переменная 2 – `unsigned int`

➤ `mas` – массив из 4 элементов типа `unsigned char`

Байт 3	Байт 2	Байт 1	Байт 0
			Переменная1
Переменная 2			
		Переменная 3	
<code>mas[3]</code>	<code>mas[2]</code>	<code>mas[1]</code>	<code>mas[0]</code>

См. дополнительный материал «Представление чисел в комп.»

Пример описания и адреса полей

```
union UN
{
    int a;
    unsigned int b;
    short int c;
    float d;
};
UN un;
int *pa = &un.a;
unsigned int *pb = &un.b;
short *pc = &un.c;
float *pd = &un.d;
cout<<'\n'<<" адреса полей "<<'\n'
<< pa<<'\n'<< pb<<'\n'<< pc<<'\n'<< pd<<'\n';
```

```
адреса полей
0046FAAC
0046FAAC
0046FAAC
0046FAAC
```

Байт 3	Байт 2	Байт 1	Байт 0
int a			
unsigned int b			
		short int c	
float d			

Пример использования_1

```
un.a = 0;
cout<<'\\n'<<"un.a = 0"<<'\\n';
cout<<"a="<<un.a<<" b="<<un.b<<" c="<<un.c<<" d="<<un.d<<'\\n';
```

```
un.a = -2147483647;
cout<<'\\n'<<"un.a = -2147483647"<<'\\n';
cout<<"a="<<un.a<<" b="<<un.b<<" c="<<un.c<<" d="<<un.d<<'\\n';
```

```
un.a = -1;
cout<<'\\n'<<"un.a = -1"<<'\\n';
cout<<"a="<<un.a<<" b="<<un.b<<" c="<<un.c<<" d="<<un.d<<'\\n';
```

```
un.a = 0
a=0 b=0 c=0 d=0
```

```
un.a = -2147483647
a=-2147483647 b=2147483649 c=1 d=-1.4013e-045
```

```
un.a = -1
a=-1 b=4294967295 c=-1 d=-1.#QNAN
```

Байт 3	Байт 2	Байт 1	Байт 0
1000 0000 0000 0000 0000 0000 0000 0001 (int)			
1000 0000 0000 0000 0000 0000 0000 0001 (unsigned int)			
	(short int)	0000 0000 0000 0001	
1000 0000 0000 0000 0000 0000 0000 0001 (float)			

Пример использования_2

```
un.b = 1;  
cout<<'\\n'<<"un.b = 1"<<'\\n';  
cout <<"a="<<un.a<<" b="<<un.b<<" c="<<un.c<<" d="<<un.d<<'\\n';
```

```
un.b = 4294967295;  
cout<<'\\n'<<"un.b = 4294967295"<<'\\n';  
cout <<"a="<<un.a<<" b="<<un.b<<" c="<<un.c<<" d="<<un.d<<'\\n';
```

```
un.b = 0;  
//ВАЖНО!! иначе старшие байты остаются с прежними значениями !!!
```

```
un.b = 1  
a=1 b=1 c=1 d=1.4013e-045
```

```
un.b = 4294967295  
a=-1 b=4294967295 c=-1 d=-1.4013e-045
```

3	Байт 2	Байт 1	Байт 0
1111 1111 1111 1111 1111 1111 1111 1111	(int)		
1111 1111 1111 1111 1111 1111 1111 1111	(unsigned int)		
	(short int)	1111 1111 1111 1111	
1111 1111 1111 1111 1111 1111 1111 1111	(float)		

Пример использования_3

```
un.b = 0; un.c = 1;
cout<<'\\n'<<"un.c = 1"<<'\\n';
cout <<"a="<<un.a<<" b="<<un.b<<" c="<<un.c<<" d="<<un.d<<'\\n';
un.c = -1;
cout<<'\\n'<<"un.c = -1"<<'\\n';
cout <<"a="<<un.a<<" b="<<un.b<<" c="<<un.c<<" d="<<un.d<<'\\n';
un.c = 32767;
cout<<'\\n'<<"un.c = 32767;"<<'\\n';
cout <<"a="<<un.a<<" b="<<un.b<<" c="<<un.c<<" d="<<un.d<<'\\n';
un.c = -32768;
cout<<'\\n'<<"un.c = -32768"<<'\\n';
un.c = 1;
cout <<"a="<<un.a<<" b="<<un.b<<" c="<<un.c<<" d="<<un.d<<'\\n';
```

```
a=1 b=1 c=1 d=1.4013e-045
```

```
un.c = -1
a=65535 b=65535 c=-1 d=9.18341e-041
```

```
un.c = 32767;
a=32767 b=32767 c=32767 d=4.59163e-041
```

```
un.c = -32768
a=32768 b=32768 c=-32768 d=4.59177e-041
```

Байт 3	Байт 2	Байт 1	Байт 0
0000 0000 0000 0000 1111 1111 1111 1111 (int)			
0000 0000 0000 0000 1111 1111 1111 1111(unsigned int)			
	(short int)	1111 1111 1111 1111	
0000 0000 0000 0000 1111 1111 1111 1111 (float)			

Пример использования_4

```
un.d = 1;
cout<<'\\n'<<"un.d = 1"<<'\\n';
cout <<"a="<<un.a<<" b="<<un.b<<" c="<<un.c<<" d="<<un.d<<'\\n';
un.d = 0.5;
cout<<'\\n'<<"un.d = 0.5"<<'\\n';
cout <<"a="<<un.a<<" b="<<un.b<<" c="<<un.c<<" d="<<un.d<<'\\n';
```

```
un.d = 1
a=1065353216 b=1065353216 c=0 d=1
un.d = 0.5
a=1056964608 b=1056964608 c=0 d=0.5
```

Байт 3	Байт 2	Байт 1	Байт 0
0011 1111 1000 0000 0000 0000 0000 0000 (int)			
0011 1111 1000 0000 0000 0000 0000 0000 (unsigned int)			
	(short int)	0000 0000 0000 0000	
0011 1111 1000 0000 0000 0000 0000 0000 (float)			

Упражнение

Вывести двоичное представление в памяти вещественного числа.

Подсказки –

- используйте объединение вещественного числа и целого без знака.
- для вывода очередной цифры используйте логическое поразрядное И числа с 1,
- для перехода к следующей цифре – сдвиг целого числа вправо на один разряд и поразрядное И числа с 1

