

枚举

枚举顾名思义就是——列举（——举例出来）

把可能的取值列举

比如说：

一周的星期一到星期日是有限的7天，可以——列举

性别有：男、女、保密，也可以——列举

月份有12个月，也可以——列举
等等....

枚举类型定义

{ } 中的内容是枚举类型的可能取值，也叫 `枚举常量`

```
//枚举类型
enum Color {
    Red,
    Green,
    Blue
};
enum Day//星期
{
    Mon,
    Tues,
    Wed,
    Thur,
    Fri,
    Sat,
    Sun
};
enum Sex//性别
{
    MALE,
    FEMALE,
    SECRET
};
```

```
int main() {
    //创建变量
    enum Color co = Bule; //即co表示蓝色

    printf("%d\n", Red);
    printf("%d\n", Bule);
    printf("%d\n", Green);
    return 0;
}
```

```
int main() {
    //创建变量
    enum Color co = Bule; //即co表示蓝色

    printf("%d\n", Red);
    printf("%d\n", Bule);
    printf("%d\n", Green);
    return 0;
}
```

Microsoft Visual Studio 调试控制台

```
0
2
1
```

E:\Github_Gitee\C\enum_union\x64\Debug
按任意键关闭此窗口. . . CSDN @期邈云汉

结果表明这些每一个枚举可能取值都是常量，也称枚举常量，不能修改。但是这些可能取值都是有值的，默认从0开始，一次递增1，当然在定义的时候也可以赋初值来修改初始值。

```
Red = 5;
```

```
enum Color {
    Red = 5,
    Green,
    Bule
};

int main() {
    //创建变量
    enum Color co = Bule; //即co表示蓝色

    printf("%d\n", Red);
    printf("%d\n", Bule);
    printf("%d\n", Green);
    return 0;
}
```

Microsoft Visual Studio 调试控制台

```
5
7
6
```

CSDN @期邈云汉

★ 如何使用？

为什么使用枚举？

我们可以使用 #define 定义常量，为什么非要使用枚举？

枚举的优点：

1. 增加代码的可读性和可维护性
2. 和#define定义的标识符比较枚举有类型检查，更加严谨
3. 防止了命名污染（封装）
4. 便于调试
5. 使用方便，一次可以定义多个常量

```

int main() {
    //创建变量
    enum Color co = Bule; //即co表示蓝色

    printf("%d\n", Red);
    printf("%d\n", Bule);
    printf("%d\n", Green);
    printf("%d\n", sizeof(co));
    return 0;
}

```

```

int main() {
    //创建变量
    enum Color co = Bule; //即co表示蓝色

    printf("%d\n", Red);
    printf("%d\n", Bule);
    printf("%d\n", Green);
    printf("%d\n", sizeof(co));
    return 0;
}

```

Microsoft Visual Studio 调试控制台

5
7
6
4

E:\Github_Gitee\C\enum_union\x64\
按任意键关闭此窗口 CSDN.@期邈云汉

这里得到其大小为4，因为枚举常量只是保存一个具体枚举常量值，而不是整个枚举类型。

```

enum Color//颜色
{
    RED=1,
    GREEN=2,
    BLUE=4
};

enum Color clr = GREEN; //只能拿枚举常量给枚举变量赋值，才不会出现类型的差异。

clr = 5; //这样赋值显然是会报错的!
//要尽量避免这样的写法

```

联合（共用体）

联合也是一种特殊的自定义类型。

这种类型定义的变量也包含一系列的成员，特征是这些成员公用同一块空间（所以联合也叫共用体）。

✳ 定义联合体

```

//联合类型的声明
union Un
{
    char c;
    int i;
};
//联合变量的定义
union Un un;
//计算变量的大小
printf("%d\n", sizeof(un));

```

```

//联合类型的声明
union Un
{
    char c;
    int i;
};
int main() {
    //联合变量的定义
    union Un un;
    //计算变量的大小
    printf("%d\n", sizeof(un));
}

```

Microsoft Visual Studio 调试控制台

E:\Github Gitee\C\enum_union\x64\De
按任意键关闭此窗口. CSDN @期遯云汉

```

int main() {
    //联合变量的定义
    union Un un;
    //计算变量的大小
    printf("%p\n", &un);
    printf("%p\n", &(un.c));
    printf("%p\n", &(un.i));

    printf("%d\n", sizeof(un));
    printf("%d\n", sizeof(union Un));
}

```

```
union Un
{
    char c;
    int i;
};

int main() {
    //联合变量的定义
    union Un un;
    //计算变量的大小
    printf("%p\n", &un);
    printf("%p\n", &(un.c));
    printf("%p\n", &(un.i));

    printf("%d\n", sizeof(un));
    printf("%d\n", sizeof(union Un));
}
```

Microsoft Visual Studio 调试控制台

00000052F9FBFAD4
00000052F9FBFAD4
00000052F9FBFAD4
4
4
E:\Github_Gitee\C\enum_union\x64
按任意键关闭此窗 IQSDN @期邈云汉

上述例子表明联合的成员是共用同一块内存空间的，这样一个联合变量的大小，至少是最大成员的大小（因为联合至少得有能力保存最大的那个成员）

★ 考虑改进判断大小端实现

```
//改进大小端判断（考虑联合体）
int check() {
    union un {
        int i;
        char c;
    }u;
    u.i = 1;
    return u.c;
}

int main() {
    int ret = check();
    if (ret == 1)
        printf("小端! \n");
    else
        printf("大端! \n");
    return 0;
}
```

Microsoft Visual Studio 调试控制台

小端!
E:\Github_Gitee\C\enum_union\x64\Debug\enum_union.exe (进程 16584) 已退出,
按任意键关闭此窗口. . .
CSDN @期邈云汉

★ 联合大小的计算

- 联合的大小至少是最大成员的大小。
- 当最大成员大小不是最大对齐数的整数倍的时候，就要对齐到最大对齐数的整数倍。

比如：

```
union Un1
{
    char c[5];
    int i;
};
union Un2
{
    short c[7];
    int i;
};

printf("%d\n", sizeof(union Un1));
printf("%d\n", sizeof(union Un2));
```



```
union Un1
{
    char c[5];
    int i;
};
union Un2
{
    short c[7];
    int i;
};
//下面输出的结果是什么?
int main() {
    printf("%d\n", sizeof(union Un1));
    printf("%d\n", sizeof(union Un2)); //当最大成员大小不是最大对齐数的整数倍的时候，就要对齐到最大对齐数的整数倍。
    return 0;
}
```

Microsoft Visual Studio 调试控制台

8
16

E:\Github_Gitee\C\enum_union\x64\Debug\enum_union.exe (进程 632) 已退出，代码为 0。
按任意键关闭此窗口。 . . .

CSDN @期遑云汉

联合体公用同一块空间，short [7] 占用14个字节，那么最大对齐数为8；int 4个字节，最大对齐数8，最终对齐数为4，取所有类型对齐数最大的，那么根据规则，就会占用4的整数倍个字节。