

4_布尔类型和引用

- 布尔类型是C++ 中的基本数据类型
 - C++在C语言的基础类型系统之上增加了bool
 - C++中的bool可取的值只有true和false
 - 理论上bool只占用一个字节

注意:

true代表真值, 编译器内部用1来表示

false代表非真值, 编译器内部用0来表示

bool类型只有true (非0) 和false (0) 两个值
C++编译器将非0值转换为true, 0值转换为false

- 可以定义 bool 类型的全局变量
- 可以定义 bool 类型的 const常量
- 可以定义 bool 类型的指针
- 可以定义 bool 类型的数组

lesson_4-1.cpp

```
1  #include <stdio.h>
2
3  int main(void)
4  {
5      //bool b = -1;
6      bool b;
7      scanf("%c", &b);
8      printf("b = %d\n", b);
9
10     b++;
11     printf("b = %d\n", b);
12
13     b = b - 3;
14     printf("b = %d\n", b);
15
16     return 0;
17 }
```

输出

```
1  root@ubuntu:~/exp/DT_CPP/part01# ./a.out
2  4
3  b = 52
```

```
4 b = 1
5 b = 1
```

lesson_4-2.cpp

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     bool b = false;
6     int a = b;
7
8     printf("sizeof(b) = %zd\n", sizeof(b));
9     printf("a = %d, b = %d\n", a ,b);
10
11     b = 3;
12     a = b;
13     printf("a = %d, b = %d\n", a ,b);
14
15     b = -5;
16     a = b;
17     printf("a = %d, b = %d\n", a ,b);
18
19     a = 10;
20     b = a;
21     printf("a = %d, b = %d\n", a ,b);
22
23     a = 0;
24     b = a;
25     printf("a = %d, b = %d\n", a ,b);
26
27
28     return 0;
29 }
```

输出

```
1 sizeof(b) = 1
2 a = 0, b = 0
3 a = 1, b = 1
4 a = 1, b = 1
5 a = 10, b = 1
6 a = 0, b = 0
```

三目运算符

```
cond ? expr1 : expr2
```

- C语言中的三目运算符返回的是 **变量的值**
 - 不能作为左值使用
- C++中的三目运算符可直接返回 **变量本身**
 - 既可以作为左值使用，还可以作为右值使用

若 `expr1` 和 `expr2` 其中有一个是常量值，则不能作为左值使用

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     int a = 1;
6     int b = 2;
7
8     printf("a = %d, b = %d\n", a, b);
9
10    (a < b ? a : 9) = 5;
11
12    return 0;
13 }
```

输出

```
1 lesson_4-4.cpp:10:23: error: lvalue required as left operand of assignment
2     10 |     (a < b ? a : 9) = 5;
3         |                   ^
4
```

- C++ 对三目运算符的升级

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     int a = 3;
6     int b = 5;
7
8     int c = 9;
9
10    printf("a = %d, b = %d, c = %d\n", a, b, c);
11
12    (a < b ? a : b) = c;
13
14    printf("a = %d, b = %d, c = %d\n", a, b, c);
```

```
15
16
17     return 0;
18 }
```

输出

```
1 a = 3, b = 5, c = 9
2 a = 9, b = 5, c = 9
```

以上三目运算符在C语言中是不能成为左值的。但在C++是可以的。

引用

- 引用 可以看作一个已定义 变量的别名
- 语法: `Type& name = var;`

普通引用在定义时就必须要初始化，且需要用**同类型的变量**进行初始化

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     int a = 4;
6     int& b = a;
7
8     printf("a = %d\n", a);
9     printf("b = %d\n", b);
10    printf("&a = %p\n", &a);
11    printf("&b = %p\n", &b);
12
13    return 0;
14 }
```

输出

```
1 a = 4
2 b = 4
3 &a = 0x7fff615a7d1c
4 &b = 0x7fff615a7d1c
```

引用b 保存 变量a 的地址

对 引用b 取地址，也就是 直接从 b的内存空间中取值。由于 引用b 保存 变量a 的地址，所以取出来的就是 变量a 的地址