

## C+ +11中auto与declare推导出来的类型区别



Jun  
资深码农，新时代农民工，知名大厂待过多年，也在初创公司干过

+ 关注他

auto与delare都可以从现有的变量类型来推断新的变量类型。

首先语法区别：

```
auto varname = value;
decltype(exp) varname = value;
```

我们接着来看看auto使用时的变量类型推导：

```
int a = 1;
int &b = a;    // b变量为int引用类型
int *c = &a;  // c变量为int指针类型
auto m = 1;   // m变量为int类型
auto n = a;   // n 变量也为int类型
auto q = b;   // q变量为int类型，注意，此处并不是和b变量相同的引用类型
auto r = c;   // r变量为int指针类型
```

使用declare的情况下：

```
int a = 1;
int &b = a;    // b变量为int引用类型
int *c = &a;  // c变量为int指针类型
declare(a) m = 1;    // m变量和a变量类型相同，为int类型
declare(b) n = b;    // n变量和b变量类型相同，为int引用类型
declare(c) q = c;    // q变量和c变量类型相同，为int指针类型
declare((b)) q = b;  // q变量和(c)变量类型不相同，为int引用类型
declare(a + b) x = (a + b);    // 变量x的类型为表达式a+b的类型，也就是int类型
```

简单的说，最大的区别是：

auto推导出的类型，是去掉引用属性的。

declare推导出的类型，除了直接的采用变量本身的声明，其他是带有引用属性的。

对于C++14版本，如果在使用auto的场合下，还想保留变量的引用类型，还可以使用declare(auto) 这种样式的声明。

```
int a = 1;
int &b = a;    // b变量为int引用类型

// 返回值为int类型
auto func(int &a)
{
    return a;
}

// 返回值为int引用类型
declare(auto) func(int &a)
{
    return a;
}
```

编辑于 2021-11-17 16:44

C++ Modern C++ C / C++

文章被以下专栏收录



C++笔记  
现代C++新特性与问题笔记，也是避坑指南

还没有评论

写下你的评论...

