

# 第3章 auto占位符

《现代C++语言核心特性解析》 谢丙堃

## auto关键字的新定义

- 声明变量时根据初始化表达式自动推断该变量的类型、声明函数时函数返回值的占位符

```
auto i = 5; // 推断为int
auto str = "hello auto"; // 推断为const char*
auto sum(int a1, int a2)->int // 返回类型后置, auto为返回值占位符
{
    return a1+a2;
}
```

## auto关键字的新定义

- 从左往右的推到

```
auto x = 1, y = 4.2;
```

- 使用表达能力更强的类型

```
auto x = true ? 1 : 4.2;  
static_assert(std::is_same<decltype(x), double>::value);
```

- 不能声明非静态成员变量
- C++20之前无法声明形参

# 推导规则

- 按值初始化

```
const int i = 5;  
auto j = i;
```

- 按引用初始化

```
int i = 5;  
int &j = i;  
auto m = j;
```

- 万能引用

```
int i = 5;  
auto&& m = i;  
auto&& j = 5;
```

- 数组或者函数

```
int i[5];  
auto m = i;
```

- 列表初始化

## auto占位符的使用

- 返回类型推导： 要求统一返回类型

```
auto sum(int a1, int a2) { return a1 + a2; }
```

- lambda表达式的形参中使用auto

```
auto l = [](auto a1, auto a2) { return a1 + a2; };  
auto retval = l(5, 5.0);
```

- 非类型模板形参占位符

```
template<auto N>  
void f() {  
    std::cout << N << std::endl;  
}
```



感谢聆听  
欢迎关注