

# 第39章 用户自定义 推导指引

《现代C++语言核心特性解析》 谢丙堃

# 使用类模板推导存在的问题

- 实现一个精简版的std::pair类模板

```
template<typename T1, typename T2>
struct MyPair {
    MyPair(const T1& x, const T2& y)
        : first(x), second(y) {}
    T1 first;
    T2 second;
};
```

# 使用类模板推导存在的问题

- 能够正确实例化的情况:

```
MyPair p(5, 11.7);
```

- 有问题的情况:

```
MyPair p1(5, "hello");
```

# 使用类模板推导存在的问题

- STL解决方案:
  - 使用std::make\_pair
  - 使用用户自定义推导指引

```
template<typename _T1, typename _T2> pair(_T1, _T2) -> pair<_T1, _T2>;
```

- 用户自定义推导指引不必一定要有模板参数

```
MyPair(int, const char*)->MyPair<long long, std::string>;  
MyPair p7(5, "hello");
```

# 聚合类型类模板的推导指引

- C++20标准之前聚合类型的类模板是无法进行模板实参推导，使用用户自定义推导指引可以解决问题。

```
template<class T>
struct Wrap {
    T data;
};
```

```
template<class T> Wrap(T)->Wrap<T>;
```

```
Wrap w1{ 7 };
Wrap w2 = { 7 };
```



感谢您的观看  
欢迎关注