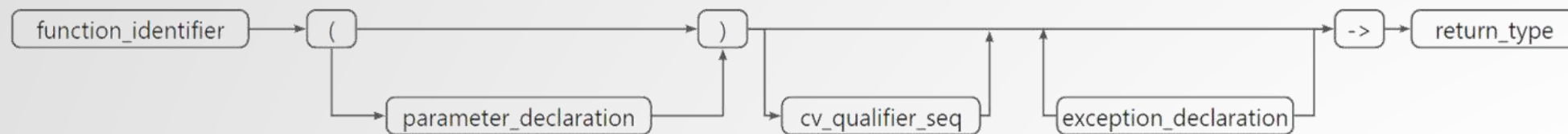


第5章 函数返回类型后置

《现代C++语言核心特性解析》 谢丙堃

使用函数返回类型后置声明函数

- 基本语法



```
auto foo()->int
{
    return 42;
}
```

使用函数返回类型后置声明函数

- 函数返回类型对比

```
int bar_impl(int x)
{
    return x;
}

typedef int(*bar)(int);
bar foo1()
{
    return bar_impl;
}
```

```
int bar_impl(int x)
{
    return x;
}

auto foo2()->int(*) (int)
{
    return bar_impl;
}
```

推导函数模板返回类型

- 结合decltype

```
template<class T1, class T2>
auto sum1(T1 t1, T2 t2)->decltype(t1 + t2)
{
    return t1 + t2;
}
```

```
template<class T1, class T2>
decltype(t1 + t2) auto sum2(T1 t1, T2 t2) {...} // 编译失败, 无法识别t1和t2
```

推导函数模板返回类型

- 其他变通方法
 - 有局限性或者不够优美

```
template<class T1, class T2>
decltype(T1() + T2()) sum3(T1 t1, T2 t2)
{
    return t1 + t2;
}
```

```
template<class T1, class T2>
decltype(*static_cast<T1 *>(nullptr) + *static_cast<T2 *>(nullptr)) sum4(T1 t1, T2 t2)
{
    return t1 + t2;
}
```



感谢聆听
欢迎关注