

20220529-元编程

1.学习内容

1.1 元编程

类型元函数

2.结果描述

1.学习内容

1.1 元编程

元编程是一种典型的函数式编程，这里的函数更接近数学意义上的函数——无副作用的映射或变换：在输入相同的前提下，多次调用同一个函数，得到的结果也是相同的。

元函数会在编译期被调用与执行。在编译阶段，编译器只能构造常量作为其中间结果，无法构造并维护可以记录系统状态并随之改变的量，因此编译期可以使用的函数（即元函数）只能是无副作用的函数。



C++ | 复制代码

```
1  constexpr int fun(int a)
2  {
3      return a+1
4  }
```

上面这段代码定义了一个函数，可以作为元函数使用。其中constexpr这个关键字表明函数可以在编译期被调用。如果去掉，则函数只能用于运行期，虽然具有无副作用的性质，但无法在编译期被调用。



C++ | 复制代码

```
1  static int call_count=3;
2  constexpr int fun(int a)
3  {
4      return a+(call_count++);
5  }
```

上面这段代码无法通过编译，因为函数内部的逻辑丧失了无副作用的性质——相同的输入会产生不同的输出。而关键字constexpr试图保持函数的无副作用特性，导致冲突。

类型元函数

▼ C++ 复制代码

```
1  template <typename T>
2  struct Fun_ {using type=T;};
3
4  template <>
5  struct Fun_<int>{using type=unsigned int;};
6
7  template <>
8  struct Fun_<long>{using type=unsigned long;};
9
10  Fun_<int>::type h=3
```

1-8行已经定义了一个函数Fun_，其具备一个元函数所需要的全部性质：

- 输入为某个类型信息T，以模板参数的形式传递到Fun_模板中；
- 输出为Fun_模板的内部类型type，即Fun_<T>::type；
- 映射体现为模板通过特化实现的转换逻辑：若输入类型为int，则输出类型为unsigned int。

2.结果描述

今天坐高铁回了广东。在班车开出校园的那一刻，内心多少有点感慨。7年前一个人背着书包拖着行李箱去到上海，没想到离开也是相似的模样。再见了上海，再见了学生时光。