

函数重载

范 懿

目录

1 概念

函数重载的动机

重载是 C++ 实现多态 (polymorphism) 的一种方式。

在 C++ 中, 多个函数可以共享一个函数名, 只要参数列表不同。

共享名字的多个函数被称作是重载的, 这个过程称作函数重载。

例

考虑 C 语言 `math.h` 头文件中的 `abs()`, `labs()` 和 `fabs()`。

- `abs()` 返回 `int` 型变量的绝对值
- `labs()` 返回 `long` 型变量的绝对值
- `fabs()` 返回 `double` 型变量的绝对值

☞ 三者都返回绝对值, 却要使用不同的名字, 简单问题复杂化。

```
1  /*在C++中，你可以只用一个名字来表达三个函数*/
2  #include <iostream>
3  using namespace std;
4  // abs以3种方式被重载
5  int abs(int i);
6  double abs(double d);
7  long abs(long l);
8
9  int main(){
10     cout << abs(-10) << "\n";
11     cout << abs(-11.0) << "\n";
12     cout << abs(-9L) << "\n";
13     return 0;
14 }
15
16 int abs(int i){
17     cout << "Using integer abs()\n";
18     return i<0 ? -i : i;
19 }
20 double abs(double d){
21     cout << "Using double abs()\n";
```

```
22     return d<0.0 ? -d : d;
23 }
24 long abs(long l)
25 {
26     cout << "Using long abs()\n";
27     return l<0 ? -l : l;
28 }
29 /*输出示例
30 Using integer abs()
31 10
32 Using double abs()
33 11
34 Using long abs()
35 9*/
36
37 /*程序定义了3个相似但不同的函数，
38 每个都返回输入参数的绝对值，
39 编译器根据输入参数的类型知道每种情况下调用的是哪个
   函数*/
40
41 /*abs()代表一个正在被执行的通用动作，
```

```
42 编译器会确定具体调用哪个方法，  
43 程序员只要记住通用的动作就可以了。*/  
44  
45 /*因为多态，原来要记三个东西，后来只要记住一个就可  
    以。*/
```

函数重载的方法

只需声明不同的版本，但不同版本的参数列表必须不同。

- 参数列表的长度不同
- 参数列表中某个分量的类型不同

🔊 仅仅在返回类型上不同是不够的。

```
1 #include <iostream>
2 #include <cstdio>
3 #include <cstring>
4 using namespace std;
5
6 // 以下两个函数同名，但第二个参数的类型不一样
7 void str_add(char *s1, char *s2);
8 void str_add(char *s1, int i);
9
10 int main()
11 {
12     char str[80];
13     strcpy(str, "Hello ");
14     str_add(str, "there");
15     cout << str << "\n";
16     str_add(str, 100);
17     cout << str << "\n";
18     return 0;
19 }
20
21
```



```
22
23
24 // 连接两个字符串
25 void str_add(char *s1, char *s2)
26 {
27     strcat(s1, s2);
28 }
29
30 // 连接一个字符串和一个字符化的数字
31 void str_add(char *s1, int i)
32 {
33     char temp[80];
34     sprintf(temp, "%d", i);
35     strcat(s1, temp);
36 }
```

- stradd() 被重载，有两个版本
- 一个版本把另一个字符串，连接到一个字符串末端
- 一个版本把一个数字，连接到一个字符串末端

 建议只重载紧密相关的函数。