

20220615-C++

1.学习内容

5.struct

5.1C中struct

5.2C++中struct

不适用typedef定义结构体别名

使用typedef定义结构体别名

1.学习内容

5.struct

5.1C中struct

- C中struct只单纯用作函数的复合类型，即只能将数据成员放在里面，而不能放函数
- C结构体不能使用C++访问修饰符（如public，protected，private）
- 在C中定义结构体变量，如果使用了，下面定义必须加struct
- C的结构体不能继承
- 若结构体的名字与函数名相同，可以正常运行并且调用

```
1  ▾ #include<stdio.h>
2  struct Base
3  ▾ {
4      int v1;
5      int v2;
6      int v3;
7  };
8
9  void Base()
10 ▾ {
11     printf("%s\n", "I am base");
12 }
13
14 int main()
15 ▾ {
16     struct Base baseI;
17     baseI.v1 = 1;
18     printf("%d\n", baseI.v1);
19     Base();
20     return 0;
21 }
```

5.2C++中struct

- C++结构体可以定义函数
- C++结构体可以使用访问修饰符
- C++结构体使用可以不带struct
- C++的结构体可以继承
- 若结构体的名字与函数名相同，可以正常运行并且调用，但是定义结构体变量的时候只能用带struct的

不适用typedef定义结构体别名

```
1  未添加同名函数前
2  struct Student
3  {
4  };
5  struct Student s;//OK
6  Student s;//OK
7  添加同名函数后
8  struct Student
9  {
10 };
11 Student(){};
12 struct Student s;//OK
13 Student s;//Error
```

使用typedef定义结构体别名

```
1  typedef struct Base
2  {
3      int v1;
4      int v3;
5  public:
6      int v2;
7      void print()
8      {
9          printf("%s", "Hello");
10     };
11 } B;
12 //void B() {};Error
13
14 int main()
15 {
16     struct Base base1;//OK
17     Base base2;
18     B base;
19     base.v1 = 1;
20     base1.v2 = 2;
21     base2.v3 = 3;
22     printf("%d\n", base.v1);
23     printf("%d\n", base1.v2);
24     printf("%d\n", base2.v3);
25     return 0;
26 }
```

```
1  struct Base
2  {
3      int v1;
4      int v3;
5  public:
6      int v2;
7      void print()
8      {
9          printf("%s", "Base");
10     };
11 };
12
13 struct Derived :Base
14 {
15 public:
16     int v2;
17     void print()
18     {
19         printf("%s", "Derived");
20     }
21 };
22
23 int main()
24 {
25     Base* b = new Derived();
26     b->print();//打印了Base
27     return 0;
28 }
```