

C/C++程序设计案例实战

——编程世界里的快递员

华中农业大学信息学院 章 英

问题引入

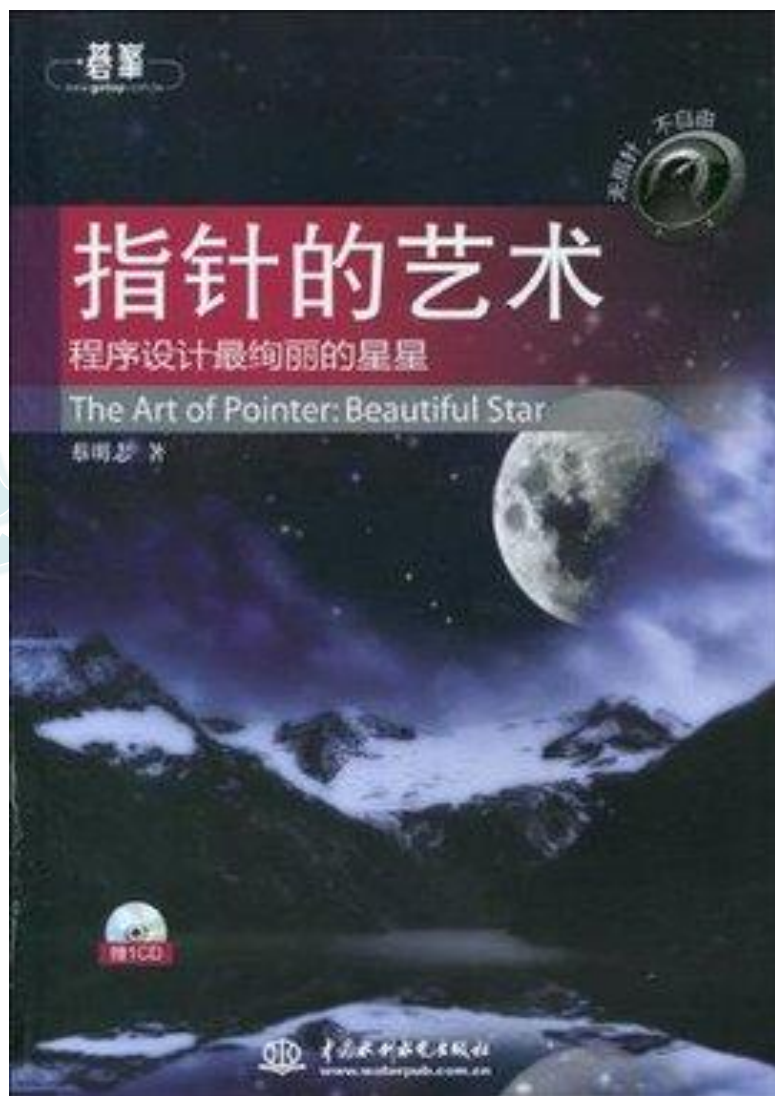
热门游戏

追求更强，征服全场

AMD 锐龙 7 3700X 处理器具备更强性能，能流畅运行全球热门游戏大作，战斗过程更流畅，让画面远离掉帧、卡顿，感受更丝滑的沉浸式体验。



问题引入



```
int p ;
```

```
int *p ;
```

```
int p[3];
```

```
int *p[3] ;
```

```
int (*p)[3] ;
```

```
int **p ;
```

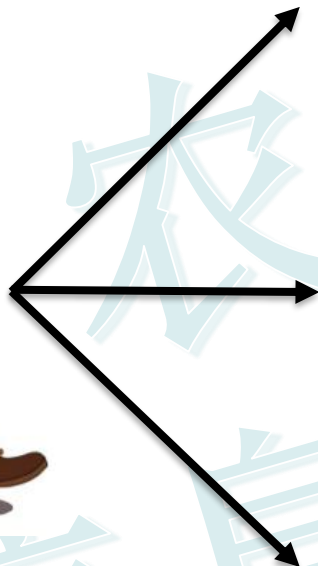
```
int p(int) ;
```

```
int (*p)(int) ;
```

案例分析

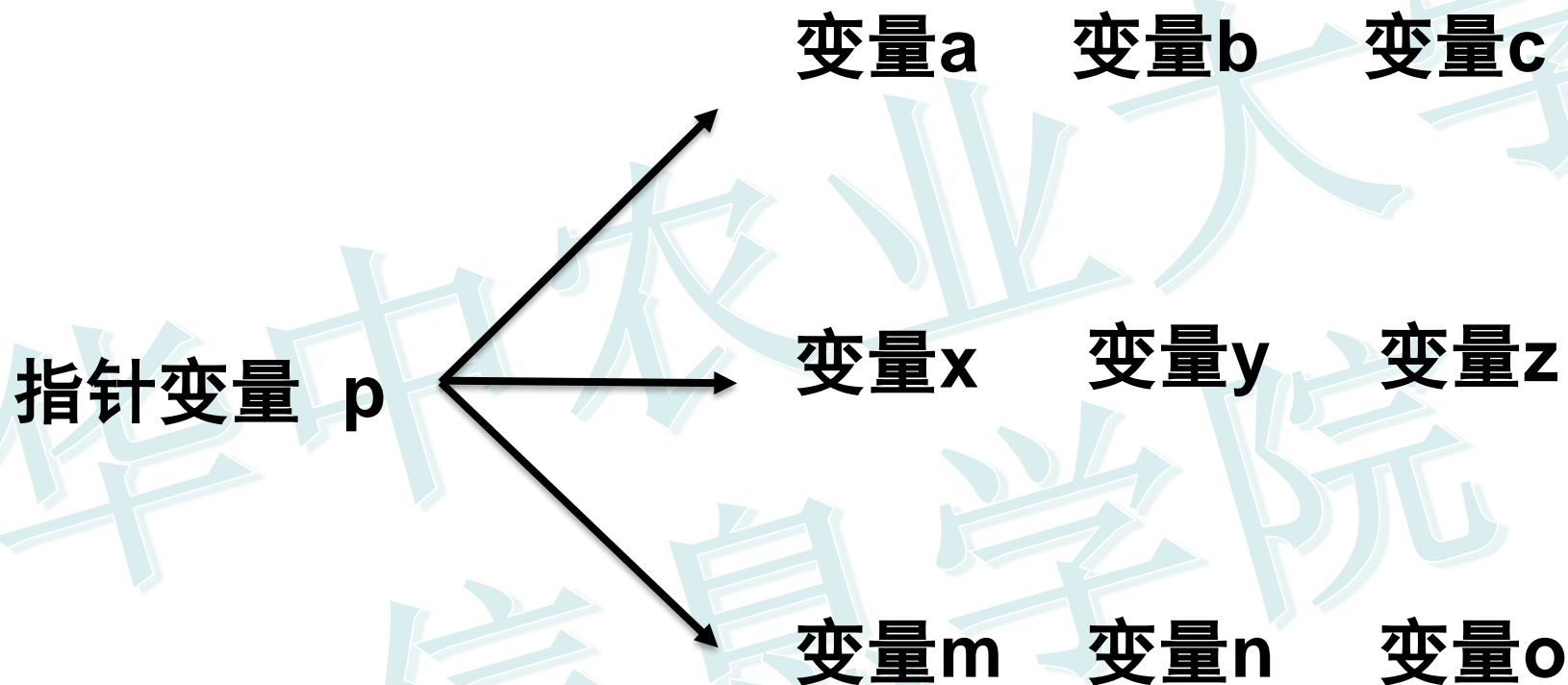


快递员



现实空间

编程世界里的快递员



指针

内存空间



庄小帅

地址：某公司

地址：华中农大荟园5栋209寝室

指针变量 p

整形变量 a

0x0037b000

20

10

地址：0x00472000

地址：0x0037b000



庄小帅

地址：某公司

地址：华中农大荟园5栋209寝室

指针变量 p

0x0037b000

整形变量 a

20

地址：0x00472000

地址：0x0037b000

案例——代码实现

```
1  #include<iostream>
2  using namespace std;
3  int main()
4  {
5      int *p, a = 10;
6
7
8
9
10 }
```

指针变量 **p**



快递员上岗

案例——代码实现

```
1  #include<iostream>
2  using namespace std;
3  int main()
4  {
5      int *p, a = 10;
6      p = &a; //p=&c; error!
7
8
9      指针变量 p
10 }
```

获得客户地址

0x0037b000

整形变量 a

10

地址: 0x0037b000

案例——代码实现

```
1  #include<iostream>
2  using namespace std;
3  int main()
4  {
5      int *p, a = 10;
6      p = &a; //p=&c; error!
7      *p = 20; //a = 20;
8
9      指针变量 p
10 }
```

0x0037b000

整形变量 a

20

地址: 0x0037b000

礼物送达

案例——代码实现

```
1  #include<iostream>
2  using namespace std;
3  int main()
4  {
5      int *p, a = 10;
6      p = &a;    //p = &c;
7      *p = 20;   //a = 20;
8      cout<<a<<" "<<*p;
9      return 0;
10 } //运行后输出20 20
```

对比：第5行和第7、8行的*p

```
1  #include<iostream>
2  using namespace std;
3  int main()
4  {
5      int *p, a = 10;
6      p = &a; //p = &c;
7      *p = 20; //a = 20;
8      cout<<a<<" "<<*p;
9      return 0;
10 }
```

指针定义符

间接引用操作符 or 取内容、取值操作符



庄小帅

地址：华中农大荟园5栋209寝室



王新宇

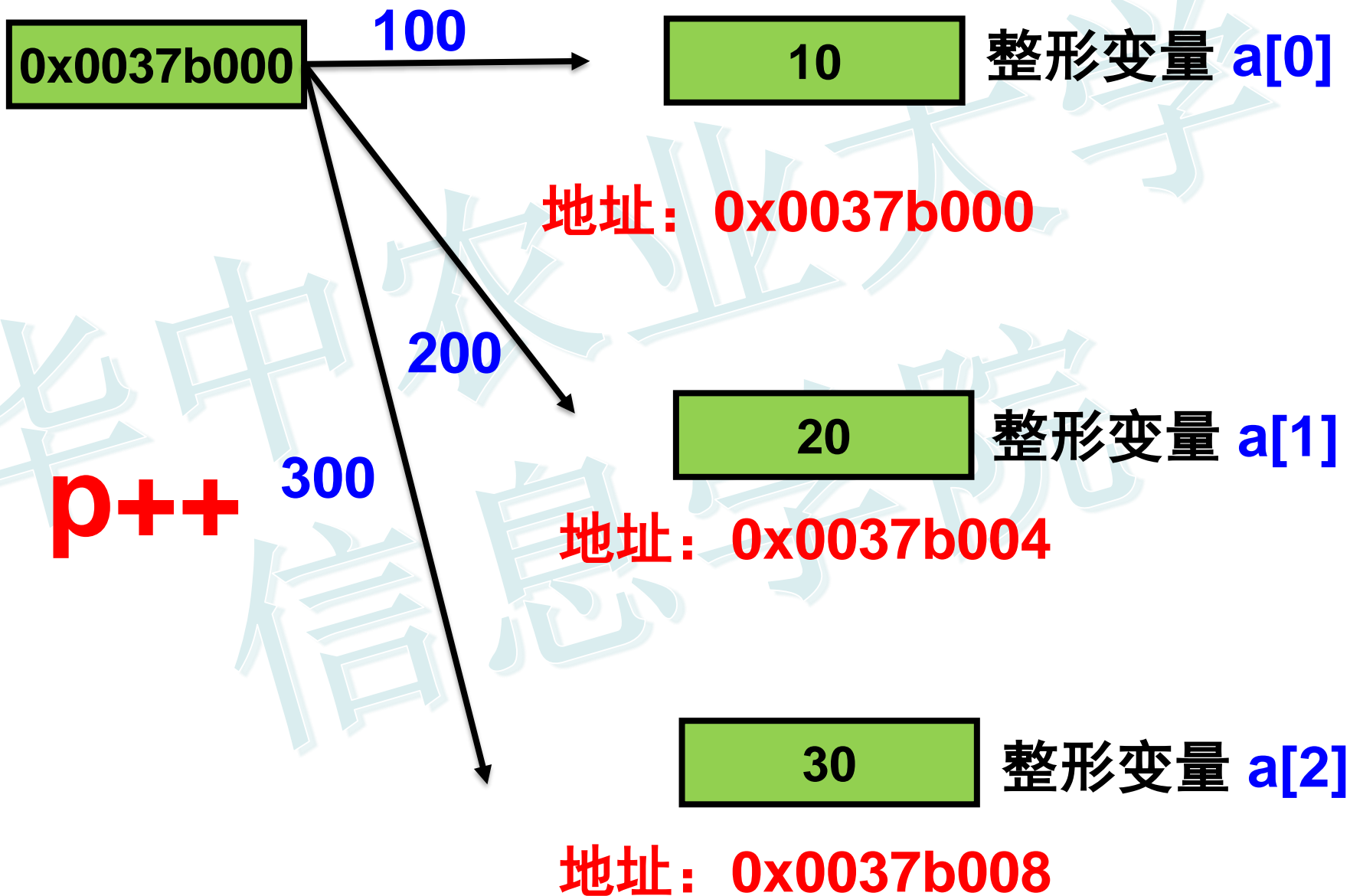
地址：华中农大荟园5栋210寝室




詹晔康

地址：华中农大荟园5栋211寝室

指针变量 p



进阶案例——代码实现

```
1  #include<iostream>
2  using namespace std;
3  int main()
4  {
5      int *p, a[3] = {10,20,30};
6
7
8
9      指针变量 p 
10
11
12
13 }
```

快递员上岗

进阶案例——代码实现

```
1  #include<iostream>
2  using namespace std;
3  int main()
4  {
5      int *p, a[3] = {10, 20, 30};
6      p = a;      //p=*a; error!
7      // p=&a[0];
8
9      指针变量 p  0x0037b000
10
11      数组名代表数组的内存起始地址
12
13 }
```

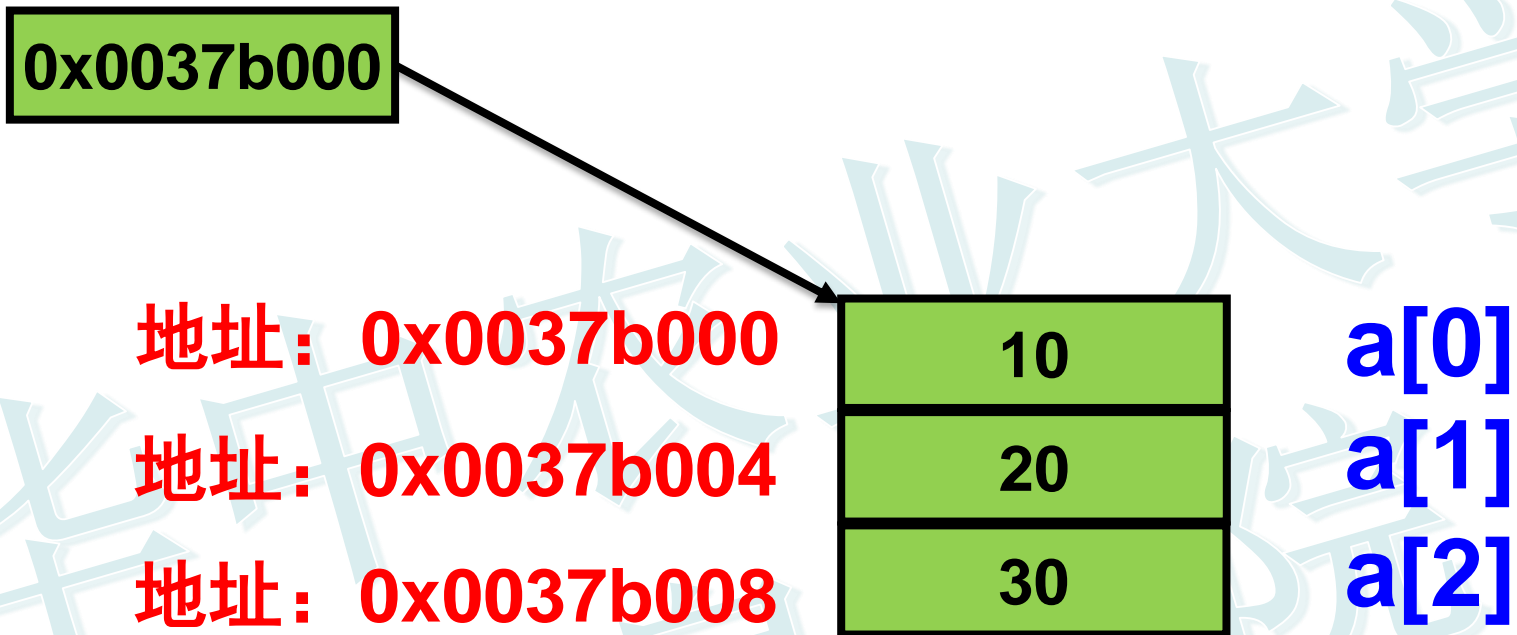
指针变量 p

0x0037b000

地址: 0x0037b000

地址: 0x0037b004

地址: 0x0037b008



10
20
30

a[0]

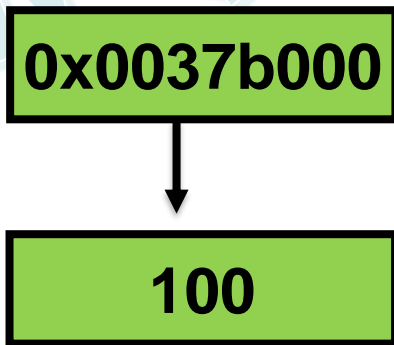
a[1]

a[2]

p = a; 等价于 p = &a[0];

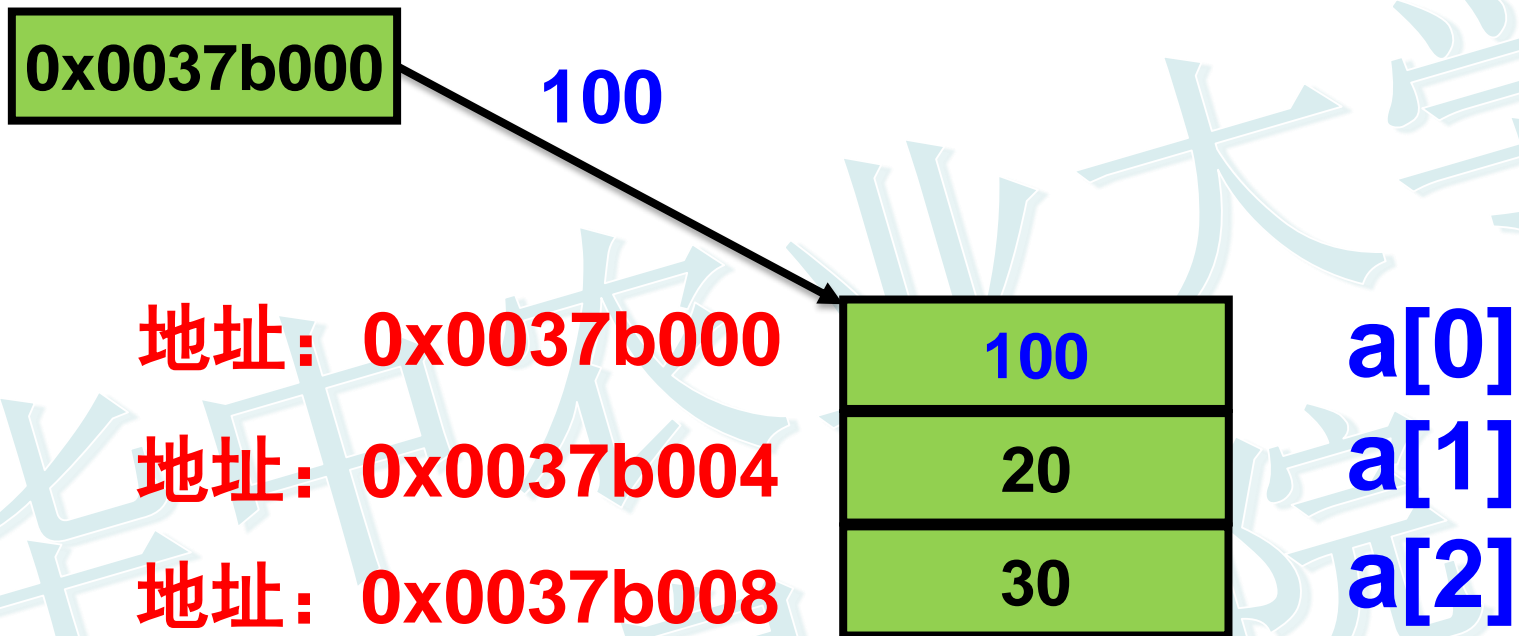
进阶案例——代码实现

```
1  #include<iostream>
2  using namespace std;
3  int main()
4  {
5      int *p, a[3] = {10, 20, 30};
6      p = a; //p=*a; error!
7      *p = 100; //a[0]=100;
8
9      指针变量 p
10
11
12  整形变量 a[0]
13 }
```



第1份礼物送达

指针变量 p



***p = 100;** 等价于 **a[0] = 100;**

进阶案例——代码实现

```
1  #include<iostream>
2  using namespace std;
3  int main()
4  {
5      int *p, a[3] = {10,20,30};
6      p = a;    //p=*a; error!
7      *p = 100; //a[0]=100;
8      p++;
9
10     指针变量 p  0x0037b004
11
12
13 }
```

进阶案例——代码实现

```
1  #include<iostream>
2  using namespace std;
3  int main()
4  {
5      int *p, a[3] = {10,20,30};
6      p = a;    //p=*a; error!
7      *p = 100; //a[0]=100;
8      p++;
9      *p = 200; //a[1]=200;
10
11
12      整形变量 a[1]
13 }
```

200

第2份礼物送达

指针变量 p

0x0037b004

200

p++;

地址: 0x0037b000

地址: 0x0037b004

地址: 0x0037b008

100
200
30

a[0]

a[1]

a[2]

*p = 200; 等价于 a[1] = 200;

进阶案例——代码实现

```
1  #include<iostream>
2  using namespace std;
3  int main()
4  {
5      int *p, a[3] = {10,20,30};
6      p = a;    //p=*a; error!
7      *p = 100; //a[0]=100;
8      p++;
9      *p = 200; //a[1]=200;
10     *(p+1) = 300; //a[2]=300;
11     cout<<*p<<" "<<*(p+1);
12     return 0;
13 } //输出200 300
```

指针变量 p

0x0037b004

地址: 0x0037b000

地址: 0x0037b004

地址: 0x0037b008

100

a[0]

200

a[1]

300

a[2]

*(p+1) 等价于 *(0x0037b008)

等价于 a[2]

小结

- (1) 能够灵活使用取地址运算符和取值运算符
- (2) 能够运用指针指向简单变量或一维数组
- (3) 能够使用指针加加运算访问数组元素

延伸

编程实现快递员根据包裹的优先级对某一小区邻居的包裹进行送货。