

指针变量

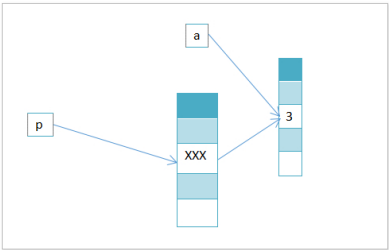
- 1、指针的概念

指针是一个变量。和普通变量不同的是，指针变量里存储的数据是一个内存地址，就好像一个指示器，指引着你去该内存地址开始的一块内存区域存取数据。

- 2、指针的定义和使用

指针变量的定义格式为：
数据类型 *指针变量

- 1、普通变量定义：int a=3;
- 2、指针变量定义：int *p=NULL;(其实就是0，表示特殊的空地址)
- 3、给指针变量p赋值：P=&a;(*p等价于a)



设有指向整型变量 p，如要把整型变量 a 的地址赋予 p 可以有以下两种方式：

- 1、指针变量初始化的方法
int a;int *p=&a;
- 2、赋值语句的方法
int a;int *p;p=&a;

| 指针的几个相关操作说明表 | |
|-----------------------|----------------------------|
| 说明 | 样例 |
| 指针定义： 类型说明符 *指针变量名 | int a=10; int *p; |
| 取地址运算符： & | P=&a; |
| 间接运算符： * | *p=20; |
| 指针变量直接存取的是内存地址 | cout<<p; 结果可能是：0x4097ce |
| 间接存取的才是储存类型的值 | cout<<*p; 结果是：20 |

例 1：输入两个不同的数，通过指针对两个数进行相加和相乘，并输出。

```
#include<bits/stdc++.h>
using namespace std;

int main(){
    int a,b,s,t;
    int *pa,*pb;

    pa=&a;pb=&b;

    a=10;b=20;

    s=*pa*pb;
    t=*pa**pb;

    printf("a=%d,b=%d\n",*pa,*pb);
    printf("s=%d,t=%d\n",s,t);

    return 0;
}
```

输出：
a=10,b=20
s=30,t=200

- 3、指针的引用与运算

一般的，我们可以这样看指针（int *p）与普通变量（int a）的对应关系

```
p—>&a
*p—a
*p=3—a=3
```

①指针变量的初始化

| 指针的几个初始化操作说明表 | | |
|---------------|------------------|--------------------|
| | 方法 | 说明 |
| 1 | Int *p=NULL; | NULL 是特殊的地址 0，叫零指针 |
| 2 | int a;int *p=&a; | p 初始化为 a 的地址 |
| 3 | int *p=new(int); | 申请一个空间给 p，*p 内容不确定 |

②指针变量的 +、- 运算

例 2: 输入 n 个数，使用指针变量访问输出。

```
#include<bits/stdc++.h>
using namespace std;
int a[101],n;

int main(){
    scanf("%d",&n);
    for(int i=1;i<=n;i++){
        scanf("%d",&a[i]);

        int p=&a[1];
        for(int i=1;i<=n;i++){
            printf("%d",*p);
            p++;
        }

        return 0;
    }
```

输入: 4
2 1 6 0
输出: 2 1 6 0

说明:
c"p+ "+"即刚好"跳过"一个整数的空间，达到下一个整数。
c"p- -"即向前"跳过"一个整数的空间，达到前一个整数。
c"(p+3) 就是指向后第3个整数的地址。

③无类型指针

例 3: 无类型指针运用举例。

```
#include<bits/stdc++.h>
using namespace std;

int a=10;
double b=3.5;
void *p;

int main(){
    p=&a;
    cout<<*(int *)p<<endl;

    p=&b;
    cout<<*(double *)p<<endl;

    return 0;
}
```

输出: 10
3.5

④多重指针

例 4: 多重指针运用举例

```
#include<bits/stdc++.h>
using namespace std;

int a=10;
int *p;
int **pp;

int main(){
    p=&a;
    pp=&p;
    printf("%d=%d\n",a,**pp);

    return 0;
}
```

输出:
10=10=10

全文完

