



动态内存分配

用new 运算符实现动态内存分配 (教材P109)

□ 第一种用法，分配一个变量：

P = new T;

T是任意类型名，P是类型为T * 的指针。

动态分配出一片大小为 sizeof(T) 字节的内存空间，并且将该内存空间的起始地址赋值给P。比如：

```
int * pn;
```

```
pn = new int;
```

```
* pn = 5;
```

用new 运算符实现动态内存分配 (教材P109)

□ 第二种用法, 分配一个数组:

P = new T[N];

T : 任意类型名

P : 类型为T * 的指针

N : 要分配的数组元素的个数, 可以是整型表达式

动态分配出一片大小为 sizeof(T) 字节的内存空间, 并且将该内存空间的起始地址赋值给P。

用new 运算符实现动态内存分配 (教材P109)

□ 动态分配数组示例:

```
int * pn;
```

```
int i = 5;
```

```
pn = new int[i * 20];
```

```
pn[0] = 20;
```

```
pn[100] = 30; //编译没问题。运行时导致数组越界
```

```
new T  
new T[n]  
both return T*  
int *p = new int
```

用delete运算符释放动态分配的内存

- 用“new”动态分配的内存空间，一定要用“delete”运算符进行释放

delete 指针; //该指针必须指向new出来的空间

```
int * p = new int;
```

```
* p = 5;
```

```
delete p;
```

```
delete p; //导致异常，一片空间不能被delete多次
```

用delete运算符释放动态分配的数组

□ 用“delete”释放动态分配的数组，要加“[]”

delete [] 指针； //该指针必须指向new出来的数组

```
int * p = new int[20];
```

```
p[0] = 1;
```

```
delete [] p;
```



下一小节： 内联函数、函数重载和函数缺省参数