面向对象方法与C++程序设计

第1章　面向对象概述

大连理工大学
主讲人-赵小薇

函数


```
类型说明符 函数名( [形式参数表] ){
    声明部分
    执行语句部分
}
```

```
float max(float x,float y){
    float z;
    z= x>y?x:y;
    return z; }
```

```
类型说明符 函数名( [形式参数表]);
```

```
float max(float x,float y);
```

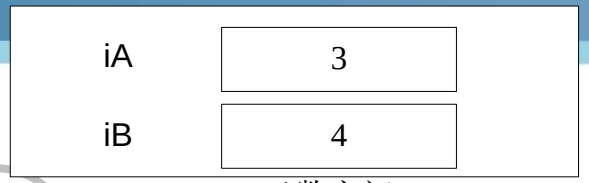
```
函数名( [实际参数表]);
```

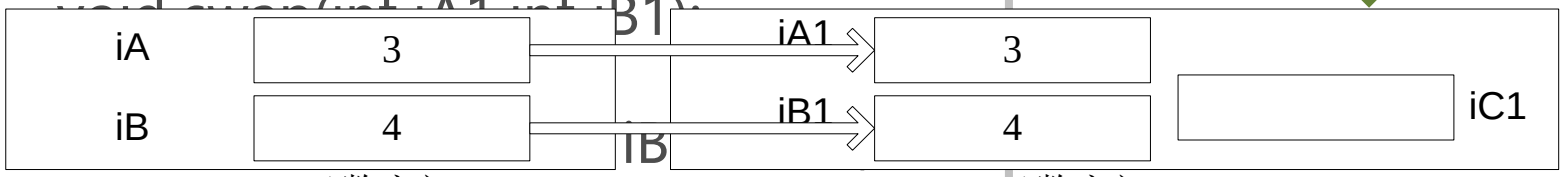
```
max(3.6,4.8);
```

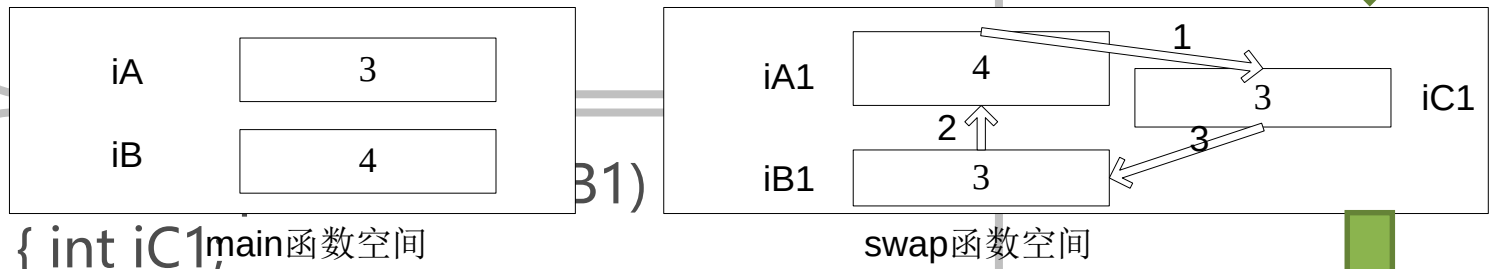
参数传递

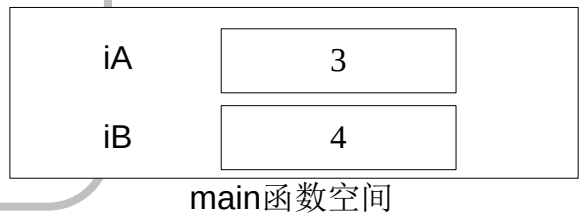


```
int main()
{ int iA=3,iB=4;
  void swap(int iA1,int iB1);
  swap(iA,iB);
  return 0;
}
void swap(int iA1,int iB1)
{ int iC1;
  iC1=iA1;
  iA1=iB1;
  iB1=iC1;
}
```







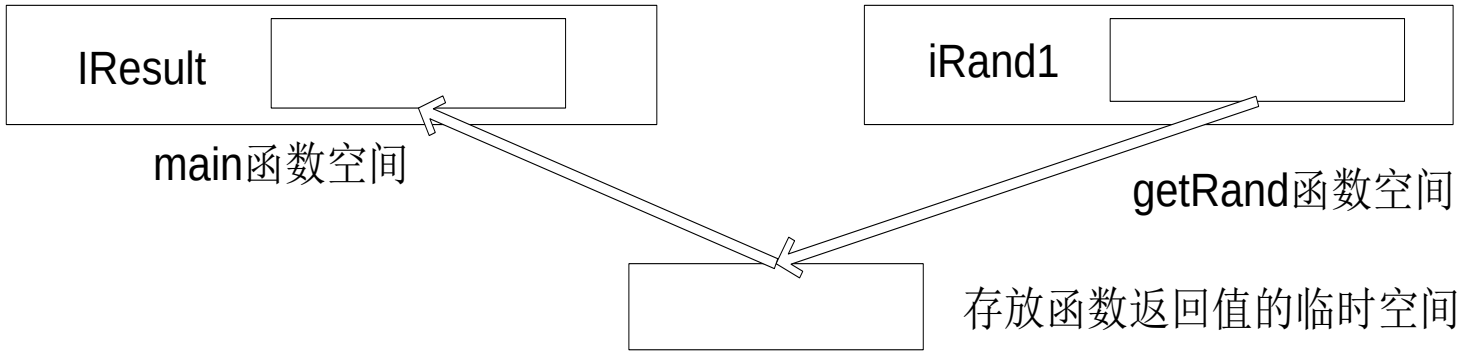


返回值



```
int main()
{
  int iResult;
  int getRand();
  iResult=getRand()%7;
  cout<<iResult;
  return 0;
}
```

```
int getRand()
{
  int iRand1;
  srand(time(0));
  iRand=rand()%100+1;
  return iRand;
}
```





- 在C语言中，为计算不同情形的最大值，需要定义多个不同名称的函数，如：

  对于计算两个整数的最大值需要定义函数：

  int maxTwoInt(int a,int b) {…}

  对于计算两个浮点数的最大值需要定义函数：

  float maxTwoFloat(float a, float b) {…}

  对于计算三个整数的最大值需要定义函数：

  int maxThreeInt (int a,int b,int c) {…}

  用户需要记住多个计算最大值的函数名，非常不方便

函数重载



- C++语言引进函数重载。
- 在同一作用范围中为多个函数定义（其功能通常是相近的）指定一个共同的函数名，委托编译器根据每一个单独函数的形参个数、类型和位置的差别进行名称区分，并选择合适的函数调用匹配称为函数重载。

重载函数定义举例



```
int max(int a,int b){…}
double max(double a, double b){…}
调用 max(1.2f, 3.4f) ;
```

- 会调用参数为double类型的函数，float类型会向double类型转换。

函数默认参数



- 默认参数也称为缺省参量，函数定义中的每一个参数都可以拥有一个默认值，如果在函数调用中没有为对应默认值的参量提供实参数据，系统就直接使用默认值。注意只能从右往左设定默认参数。

- 如函数定义：

```
int  f (int x,long y=10, double z=20) {...}
```

- 这样就可以采用几种等价地调用形式：

```
f (3);
f (3,10);
f (3,10,20);
```

函数默认参数举例



- 默认参数函数调用时，遵循参数调用顺序，

  自左到右逐个匹配，函数定义：

```
void mal(int a, int b=3, int c=5) {…}      //默认参数
mal(3, 8, 9 );                              //不使用默认参数
mal(3, 5); //按从左到右顺序调用，相当于mal(3,5,5);
mal(3);       //按从左到右顺序调用,相当于mal(3,3,5);
mal( );                                     //错误，因为a没有默认值
mal(3, , 9);          //错误，应按从左到右顺序逐个调用
```

函数调用二义性



- 当函数重载与默认参数同时使用容易出现二义性：

  重载函数定义：

  int max(int a,int b){…}

  int max(int a,int b,int c=5){…}

  调用：max(4,5)；//二义性错误

- 调用两个参数的、还是调用三个参数的，两种情形都符合规则，编译系统无法确定，因此出现二义性错误。因此，默认参数与重载函数尽量不要同时采用。