

# 编程手记

杨志鹏

2015 年 1 月 15 日

## 目录

<b>1</b>	<b>C/C++</b>	<b>2</b>
1.1	类 . . . . .	2
1.2	指针 . . . . .	3
1.3	vector(C++) . . . . .	3
1.4	多线程 . . . . .	3
<b>2</b>	<b>C#</b>	<b>4</b>
<b>3</b>	<b>Java</b>	<b>4</b>
<b>4</b>	<b>L<sup>A</sup>T<sub>E</sub>X</b>	<b>4</b>
<b>5</b>	<b>PHP</b>	<b>5</b>
<b>6</b>	<b>MySQL</b>	<b>5</b>
<b>7</b>	<b>Javascript</b>	<b>5</b>
<b>8</b>	<b>Python</b>	<b>5</b>
<b>9</b>	<b>Ocaml</b>	<b>5</b>

## 1 C/C++

### 1.1 类

#### • 虚函数

在基类中声明虚函数的话，用基类的指针指向基类的对象的时候会调用基类的函数，指向派生类的对象的时候会调用派生类的函数。（一旦在基类中声明虚函数，派生类中相同的函数同时也成为虚函数）

虚函数在基类中使用 `virtual` 定义，如：

```
virtual void function(){...}
```

#### • 纯虚函数

纯虚函数在基类中用 `virtual` 和 `=0` 定义，有纯虚函数的类都是抽象类，如：

```
virtual void function()=0
```

#### • 虚基类

设 class A,B,C,D。A 中有变量 a，且 B、C 继承 A，D 继承 B、C。

如果有 D d。此时若 `cout<<d.a<<endl;` 会出错。因为 d.a 可能是 B 继承的 a，也可能是 C 继承的 a。

所以在 B、C 继承 A 的时候，要加 `virtual`，声明 B、C 用虚基类的方式继承 A（必须 B、C 同时声明）。那么当 D 继承 B、C 的时候，B 和 C 只保留一份 a 的副本。也就是说 `d.A::a`、`d.B::a`、`d.C::a`、`d.a` 在内存中均指向一个地址。

※ 类 B、C、D 中不能再定义 a

虚基类使用如下方法定义：

```
class A{
public:
    int a;
};
class B: virtual public A{
    ...
};
class C: virtual public A{
    ...
};
```

```
};  
class D: public B, public C{  
    ...  
};
```

## 1.2 指针

### • 普通函数指针

使用如下方法定义一个指向返回值为 `int*`，参数为 `int*` 的函数的指针：

```
int (*(*pf)(int*));
```

### • 参数为函数指针且返回函数指针的函数

使用如下方法定义一个参数为函数指针且返回函数指针的函数：

```
int* (*func2(int (*(*pf)(int*)), int *p))(int*){}
```

或 (C++11)

```
int* (*func2(int (*(*pf)(int*)), int *p))(int*){}
```

```
auto func3(int (*(*pf)(int*)), int *p) -> int *(*)(int*)
```

### • 使用 typedef 定义函数指针

使用如下方法定义一个指向返回值为 `int*`，参数为 `int*` 的函数的指针：

```
typedef int *(*PF)(int*);
```

## 1.3 vector(C++)

头文件：

```
#include <vector>
```

## 1.4 多线程

在最新的 C++11 中，可以通过 `#include <thread>` 使用多线程。

```
#include<thread>
#include<iostream>
using namespace std;
void Task0(int a){
    cout<<"Task"<<a<<endl;
}
void Task1(int a){
    cout<<"Task"<<a<<endl;
}
int main(int args, char* argc[]){
    thread t0(Task0,0);
    thread t1(Task1,1);

    t0.join();
    t1.join();

    system("pause");
    return 0;
}
```

## 2 C#

## 3 Java

- 按位右移，并补 0（下列例子结果是 000000010，如果用 »，则是 0000010）：

8>>>2

## 4 L<sup>A</sup>T<sub>E</sub>X

- 定义字体颜色：

`{\color{red/blue/black..} TEXT}`

或

`{\color[rgb]{0.0/0.0/0.0} TEXT}`

**5 PHP**

**6 MySQL**

**7 Javascript**

**8 Python**

**9 Ocaml**