

您的潜力，我们的动力

**Microsoft**  
微软(中国)有限公司

# VC++ 2005：指针与对象模型

李建忠

上海祝成科技 高级讲师

2005年5月26日

- CLI对象模型概览
- 追踪句柄 tracking handle
- 内部指针 interior pointer
- 固定指针 pinning pointer
- 讲座总结
- Q&A

# CLI对象模型简介

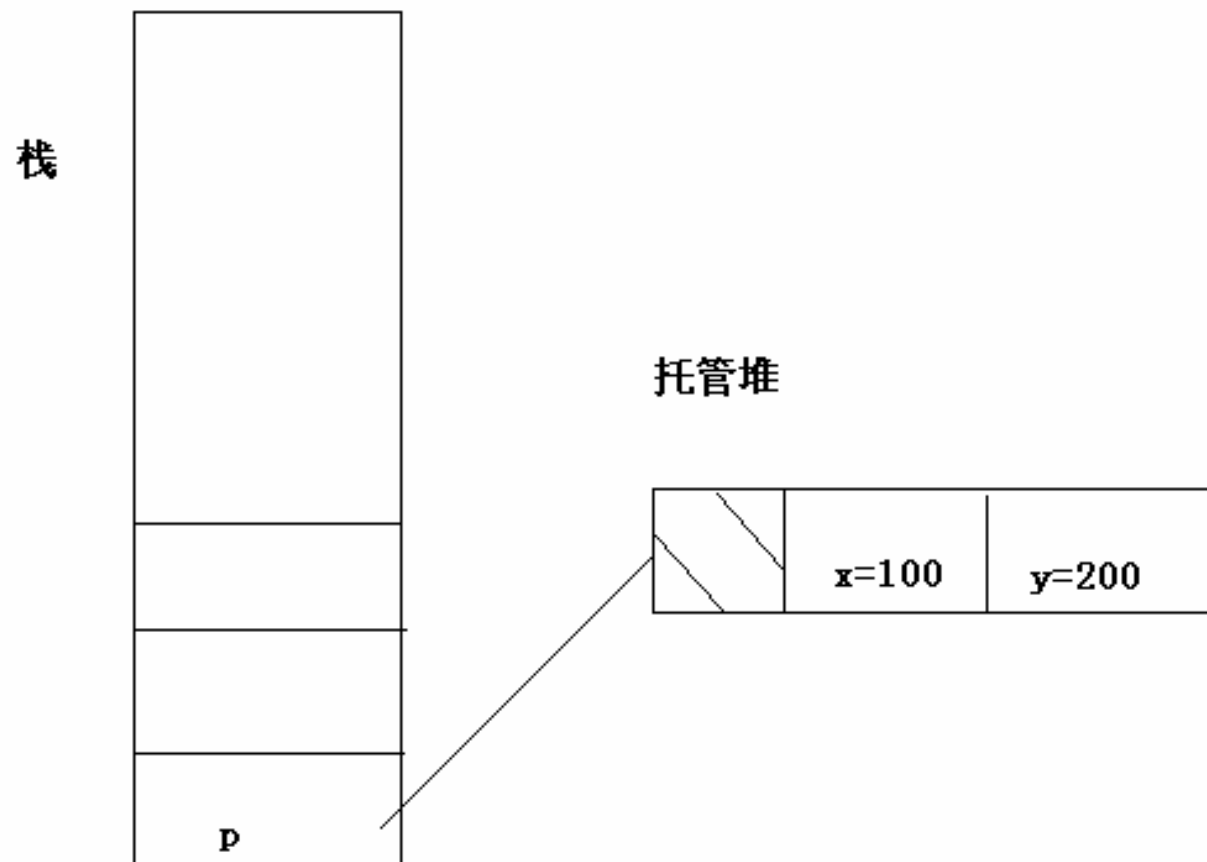
- 对象模型是类型实例在机器（虚拟机）中的表示
- 对象模型是深刻认识现代面向对象/面向组件编程语言之根本。
- 掌握C++/CLI的关键是掌握托管CLI对象模型，以及本地C++对象模型

# CLI对象模型(1)

```
public ref class Point {  
public:  
    int x;  
    int y;  
};
```

```
int main() {  
    Point^ p = gcnew Point;  
    p->x = 100;  
    p->y = 200;  
}
```

# CLI对象模型(1)



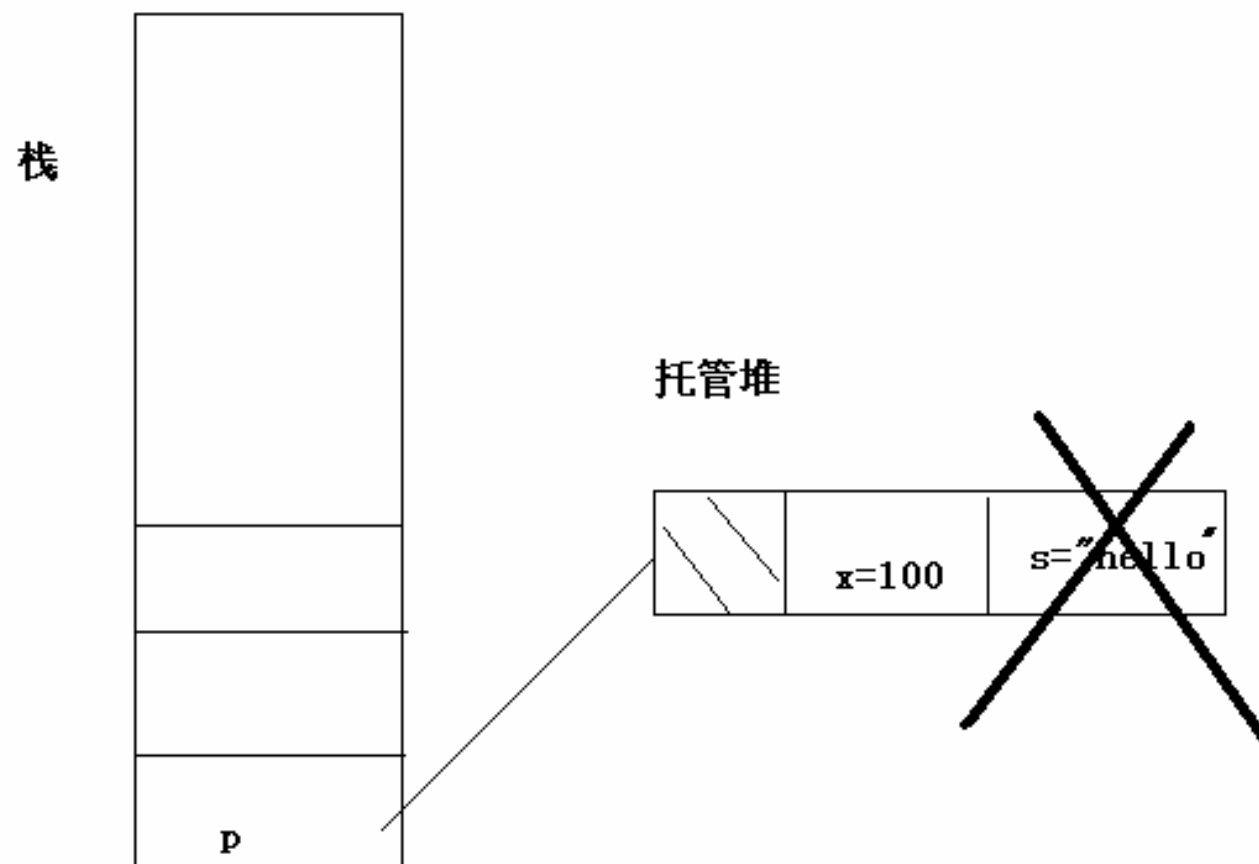
# CLI对象模型(2)

```
public ref class Point {  
public:  
    int x;  
    System::String^ s;  
};
```

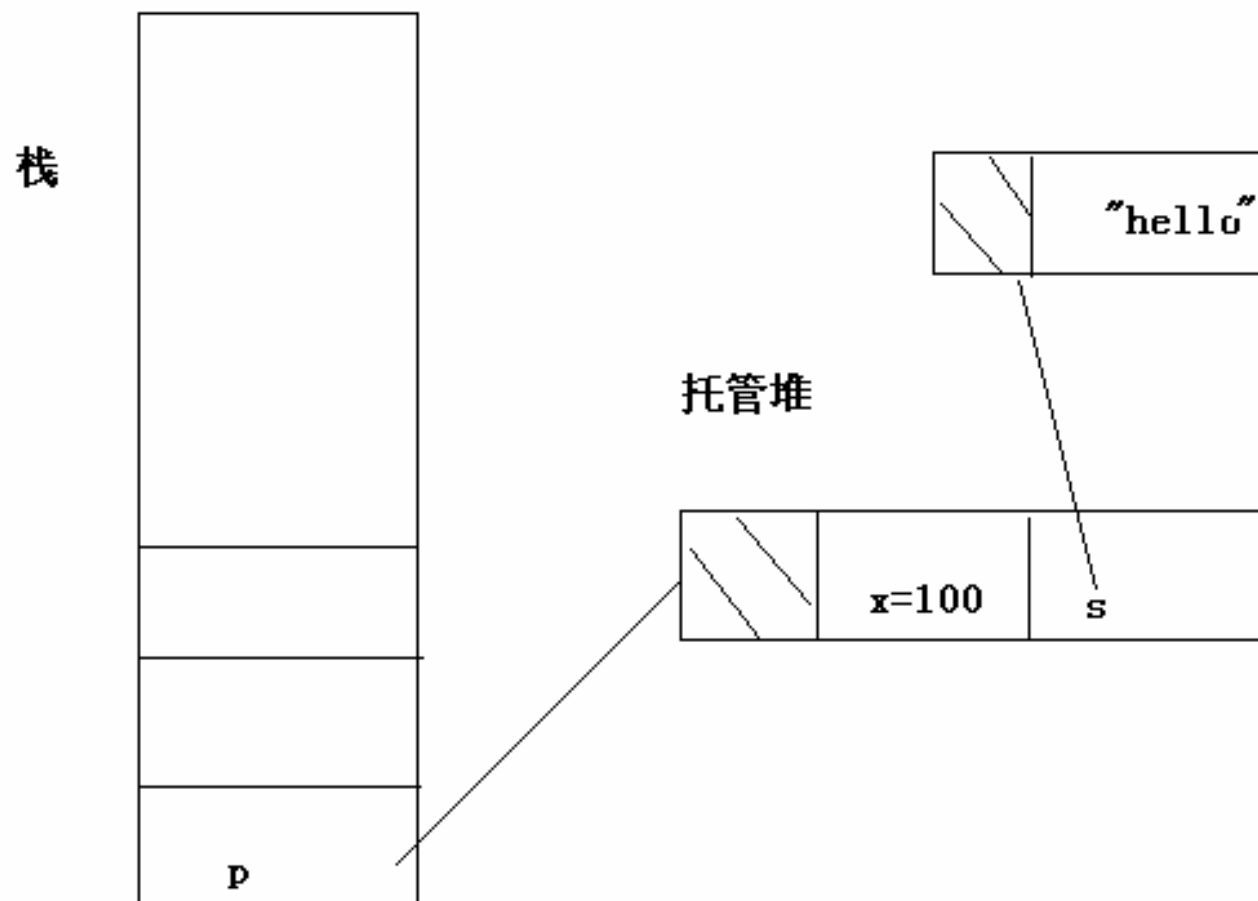
```
int main() {  
    Point^ p = gcnew Point;  
    p->x = 100;  
    p->s = "hello";  
}
```



# CLI对象模型(2)



# CLI对象模型(2)

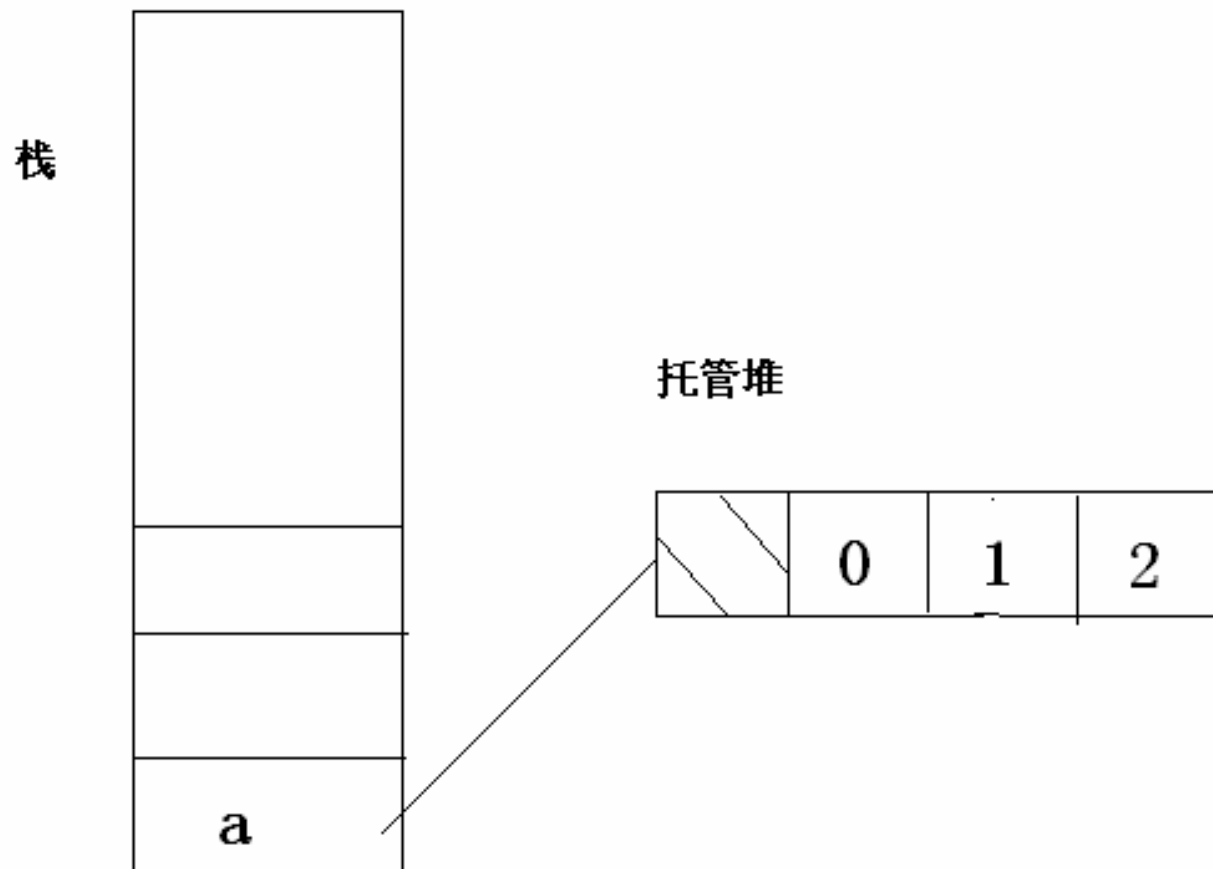




# CLI对象模型(3)

```
int main() {  
  
    array<int>^ a =  
        gcnew array<int>(3){0,1,2};  
  
    for each(int i in a)  
    {  
        System::Console::WriteLine(i);  
    }  
}
```

# CLI对象模型(3)



# Agenda

- CLI对象模型概览
- 追踪句柄 tracking handle
- 内部指针 interior pointer
- 固定指针 pinning pointer
- 讲座总结
- Q&A

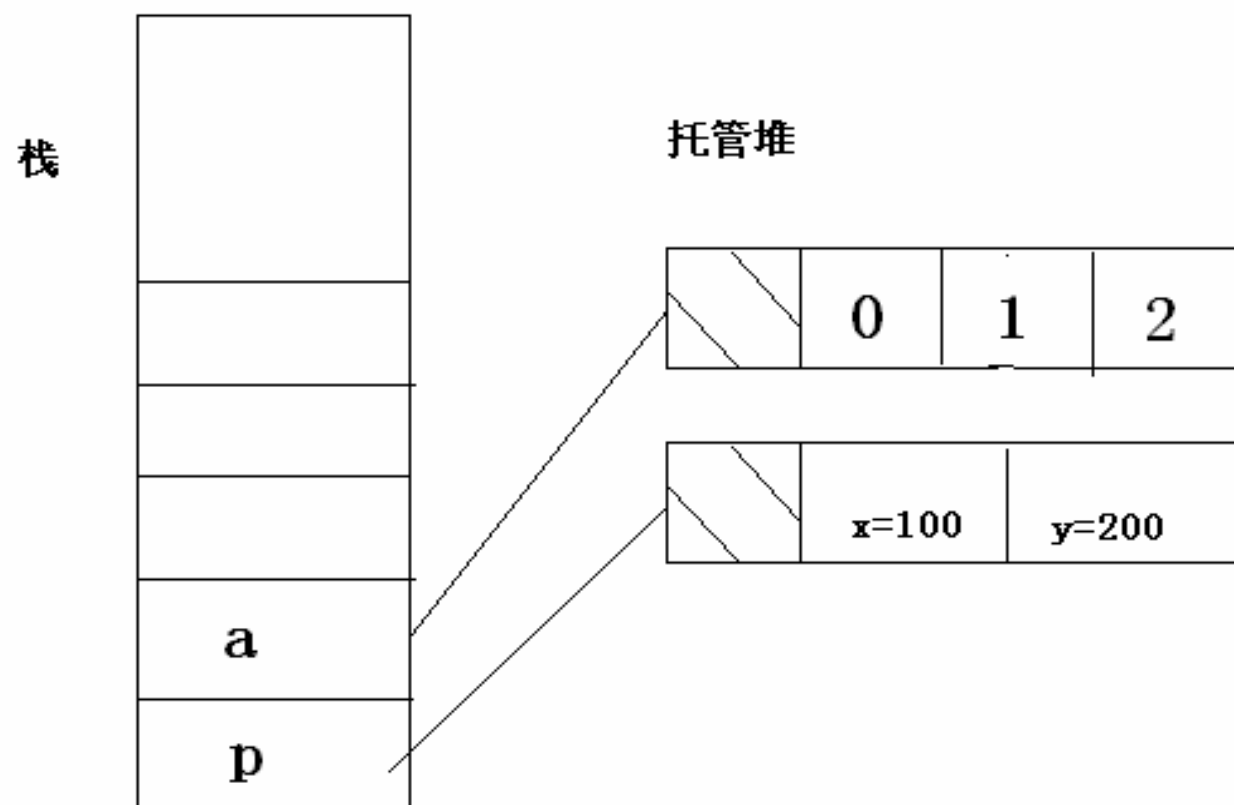
# 追踪句柄 tracking handle

```
int main() {  
    Point^ p=gcnew Point;  
    array<int>^ a=  
        gcnew array<int>(3){0,1,2};  
}
```

追踪句柄——指向“托管堆上一个完整的对象（whole object）”的指针

**Microsoft®**  
微软(中国)有限公司

# 追踪句柄的对象模型



# Agenda

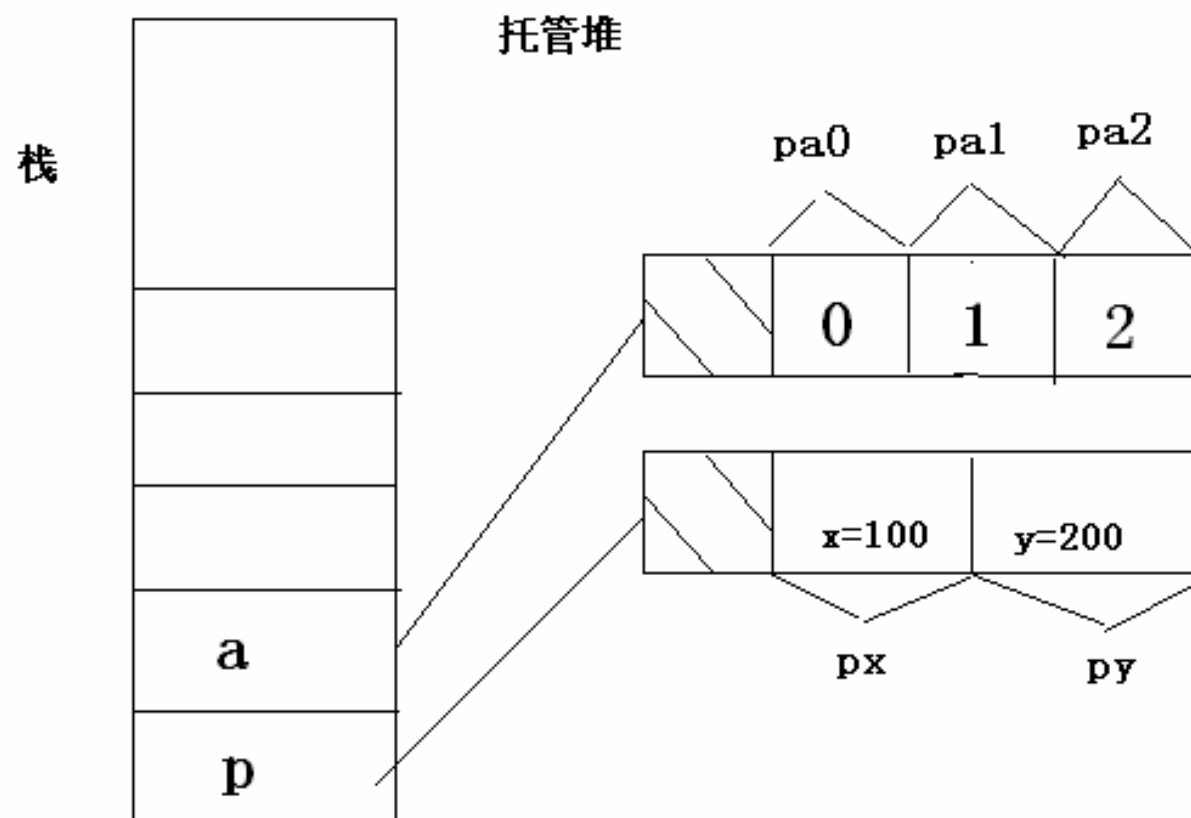
- CLI对象模型概览
- 追踪句柄 tracking handle
- 内部指针 interior pointer
- 固定指针 pinning pointer
- 讲座总结
- Q&A



# 内部指针 interior pointer

```
int main() {  
    Point^ p=gnew Point();  
    array<int>^ a=  
        gnew array<int>(3){0,1,2};  
  
    interior_ptr<int> px=&p->x;  
    interior_ptr<int> py=&p->y;  
  
    interior_ptr<int> pa0=&a[0];  
    interior_ptr<int> pa1=&a[1];  
    interior_ptr<int> pa2=&a[2];  
}
```

# 内部指针的对象模型



# 内部指针的适用对象

- 包含本地指针，本地指针的超集
- 可以指向：
  - 引用句柄
  - 值类型或者装箱值类型句柄
  - 托管类型内部成员
  - 托管数组内部元素

# Agenda

- CLI对象模型概览
- 追踪句柄 tracking handle
- 内部指针 interior pointer
- 固定指针 pinning pointer
- 讲座总结
- Q&A

# 固定指针 pinning pointer

```
void native_function(int* p) ;
```

```
int main() {
```

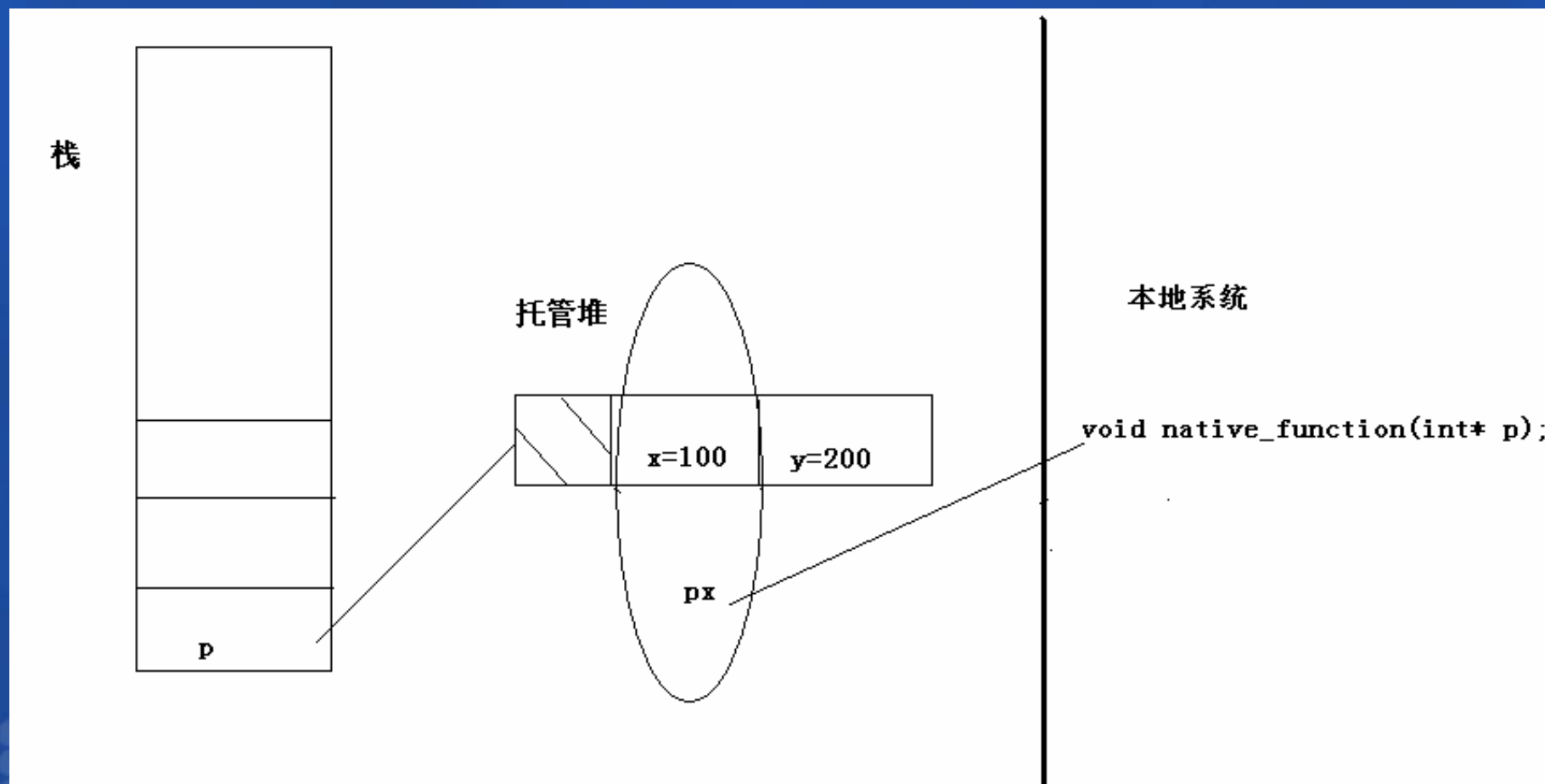
```
    Point^ p=gnew Point();
```

```
    pin_ptr<int> px=&p->x;
```

```
    native_function(px);
```

```
}
```

# 固定指针的对象模型





# 固定指针的适用对象

- 可以指向：
  - 引用句柄
  - 值类型或者装箱值类型句柄
  - 托管类型内部成员
  - 托管数组内部元素

# Agenda

- CLI对象模型概览
- 追踪句柄 tracking handle
- 内部指针 interior pointer
- 固定指针 pinning pointer
- 讲座总结
- Q&A

# 讲座总结

- C++/CLI由于垃圾收集器的引入，大大改变了对象模型
- 有什么样的对象模型，便有什么样的指针类型
- C++/CLI最复杂的集成是托管对象模型和本地对象模型的集成

# Agenda

- CLI对象模型概览
- 追踪句柄 tracking handle
- 内部指针 interior pointer
- 固定指针 pinning pointer
- 讲座总结
- Q&A

您的潜力，我们的动力

**Microsoft**<sup>®</sup>  
微软(中国)有限公司

# Q & A



# 资源链接

- [msdn.microsoft.com/visualc](http://msdn.microsoft.com/visualc)
- [comp.lang.c++.moderated](http://comp.lang.c++.moderated)
- [blogs.msdn.com/slippman](http://blogs.msdn.com/slippman)
- [pluralsight.com/blogs/hsutter](http://pluralsight.com/blogs/hsutter)
- [blog.joycode.com/lijianzhong](http://blog.joycode.com/lijianzhong)
- [www.chinaitclub.org/forums/](http://www.chinaitclub.org/forums/)



您的潜力，我们的动力

**Microsoft®**  
微软(中国)有限公司

**Microsoft®**

msdn  


**MSDN Webcasts**