

编程入门基础

—— this 引用

主讲教师：耿宇航



this引用

- this就是正在执行本方法的当前对象的指针
- 方法的执行依赖于对象

this的要点

- this是形参变量
 - this的生命期很短：从方法调用到结束
 - 同一方法的两次调用，this是不同的变量，其值可能相同或不同
- this是引用类型的变量
 - this的值与调用时的实参变量相同
- a.f(x)
 - 则，a为实参（引用类型）
 - this为形参（引用类型）
 - 当f执行的时候，this与a指向了同一个对象

到底什么是方法？

- 方法的描述是公共的
 - 方法的描述语句是公共的，不是特定到某一个具体的对象
- 方法的执行是特定的
 - 方法一旦执行，总是特定到某个具体的对象
 - 除非.....

对象的内存图景

```
class Age  
{  
    private int n;  
    public Age() { n=1;}  
    public void grow() { n++;}  
    public int get() { return n;}  
}
```

```
Age a = new Age();  
a.grow();  
Age b = new Age();  
b.grow();  
Age c = new Age();  
c.grow();  
c.grow();
```

对象的内存图景

所有对象公用的方法

```
Age.grow(Age this)  
{  
  this.n ++;  
}  
  
int Age.get(Age this)  
{  
  return this.n;  
}  
  
Age.Age(Age this){  
  this.n = 1 }
```

Age a



int n

Age b



int n

Age c



int n

变量的生存期

- `class A`
 - `{`
 - `private int x;`
 - `public void f(){ x++; }`
 - `}`
-
- 上面的x并非永存！只有在`new A()` 的时候，x才跟随对象出生！
 - 当创建多个对象的时候，x会存在多个实例
-
- f 方法中 x 从哪里来？实质是 `this.x++`
 - 而this是从f方法的参数列表中传来的（被语法上隐藏了）
 - 实际：`public void f(A this) { this.x++; }`

谢谢！