

虚方法

虚方法就是java类在继承中，在上转型中，java类对象实际调用的方法是子类重写的方法（可以被覆写的方法）；也就是编译器和jvm调用的不是同一个类的方法；也可以理解为除了用static、final、private修饰之外（应该还有构造方法）的所有方法都是虚方法。

例子：

```
/* 文件名 : Employee.java */
public class Employee
{
    private String name;
    private String address;
    private int number;
    public Employee(String name, String address, int number)
    {
        System.out.println("Constructing an Employee");
        this.name = name;
        this.address = address;
        this.number = number;
    }
    public void mailCheck()
    {
        System.out.println("Mailing a check to " + this.name
            + " " + this.address);
    }
}
```

```
/* 文件名 : Salary.java */
public class Salary extends Employee
{
    private double salary; //Annual salary
    public Salary(String name, String address, int number, double
        salary)
```

```

{
    super(name, address, number);
    setSalary(salary);
}
public void mailCheck()
{
    System.out.println("Within mailCheck of Salary class ");
    System.out.println("Mailing check to " + getName()
        + " with salary " + salary);
}

```

```

/* 文件名 : VirtualDemo.java */
public class VirtualDemo
{
    public static void main(String [] args)
    {
        Salary s = new Salary("Mohd Mohtashim", "Ambehta, UP", 3,
3600.00);
        Employee e = new Salary("John Adams", "Boston, MA", 2,
2400.00);
        System.out.println("Call mailCheck using Salary reference --
");
        s.mailCheck();
        System.out.println("\n Call mailCheck using Employee
reference--");
        e.mailCheck();
    }
}

```

当编译器检查e.mailCheck()方法时，编译器检查到**Employee**类中的**mailCheck()**方法。

在编译的时候，编译器使用Employee类中的mailCheck()方法验证该语句，在运行的时候，Java虚拟机(JVM)调用的是**Salary**类中的**mailCheck()**方法。

该行为被称为虚拟方法调用，该方法被称为虚拟方法。