

多态

2019年11月9日 15:35

多种形态：

编译时多态：方法重载（在同一个类中 方法名相同 参数列表）

在编译期，编译器就可以根据方法参数的个数以及数据类型来确定将要调用的是哪个方法；方法重载在编译期间就已经绑定了要调用的方法；

运行时多态：

向上造型 方法重写

向上造型：

创建子类对象并赋值给一个父类引用；

子类其实就是父类的一种；

方法重写：运行时多态的一种

注意：

- 1、编译期看左边变量的类型，如果左侧变量的类型包含了属性和方法，就可以直接调用；如果没有包含，编译时报错
- 2、运行时看右边对象的类型，调用的是右边对象的方法。
- 3、不能将父类对象直接赋值给子类的引用，编译会报错
- 4、可以通过强制类型将父类转成子类对象，前提是父类本身是子类对象，否则运行抛出异常

Exception in thread "main" java.lang.**ClassCastException**:
cn.tedu.duotai.Animal cannot be cast to cn.tedu.duotai.Dog

方法重写原则理解加深：

- 1、子类重写父类方法，要求子类的权限修饰符要大于等于父类的权限修饰符；两小；

```
class A{  
    protected void a(){  
    }  
}
```

```

}
class B extends A{
    public void a(){
    }
}
A a = new B();
a.a();

```

2、如果父类方法的返回值类型是引用数据类型，则子类重写方法的返回值类型必须是父类方法返回值类型的子类或者本身

```

class A{
}
class B extends A{
}
class C{
    public A m(){
    }
}
class D extends C{
    public B m(){
    }
}
C c = new D();
A a = c.m();
A a = B;

```