

day03-多态、内部类、API

多态

- 概念 — 同类型的对象，调用同一个行为，表现出不同的行为特征
- 形式
 - 父类 对象名称 = new 子类构造器;
 - 接口 对象名称 = new 实现类构造器
- 多态的识别技巧
 - 对于方法调用：编译看左边，运行看右边
 - 对于变量调用：编译看左边，运行也看左边
- 多态的使用前提
 - 必须继承或实现关系
 - 必须父类类型指向子类对象
 - 存在方法重写
- 多态的优势
 - 右边对象解耦合
 - 父类类型变量作为方法的参数，可以接收一切子类对象
- 多态的劣势 — 多态下无法调用子类独有功能
- 多态的类型转换
 - 自动类型转换：Animal a = new Dog();
 - 强制类型转换： Dog d = (Dog)a;
 - 出现类型转换异常
 - 有继承关系的两个类就可以强制转换，编译不报错
 - 运行可能出错 — Cat c = (Cat)a;
 - Java建议强制转换前先判断真实类型
 - a instanceof Dog
 - a instanceof Cat

内部类

- 一个类里面的类就是内部类
- 静态内部类
- 成员内部类
- 局部内部类
- 匿名内部类
 - 方便构建子类对象，最终为了简化代码
 - 格式：new 类|抽象类|接口() { 方法重写 }
 - 特点
 - 是一个没有名字的内部类
 - 本身也是一个对象，是自己匿名内部类的对象
 - 这个对象相当于是一个子类对象
 - 使用形式
 - 作为方法的参数，完成对象回调
 - 最终可以简化代码（详细讲解）

常用API(理解+记忆)

参见代码和PPT