



JAVA

Object类



Object类是Java中所有类的父类

Java中的每个类都是Object类的直接或间接子类

由于这种特殊地位，这个类中定义了所有对象都需要的状态和行为

Object类的所有方法都是public或protected的，能被子类继承



Object类中的常用方法

Object

Object类中的方法分为两类

一类可以被子类重写

一类是最终方法，不能被子类重写



Object类中的常用方法

Object — 类中可以被重写的方法

- protected Object clone()

- 生成当前对象的一个备份

- public boolean equals(Object obj)

- 比较当前对象和另外一个Object对象是否相等

- public String toString()

- 返回对象有关内存的信息



Object类中的常用方法

Object 类中的最终方法

- `public final Class getClass()`
 - 获得当前所属类的信息
- `public final void notify()`
 - 唤醒一个在等待的单个线程
- `public final void notifyAll()`
 - 唤醒在等待的所有线程。
- `public final void wait() throws InterruptedException`
 - 让当前线程挂起，直到另外一个线程用 `notify()` 方法或 `notifyAll()` 方法唤起



Object类常用方法举例

- `public boolean equals(Object obj)`
 - 比较当前对象的引用是否与参数obj的引用指向同一个对象
 - 这个方法如果不重写，意义不是很大
 - 很多类都重写了这个方法，以满足实际的开发需求
 - 录屏讲解TestEquals.java



Object类常用方法举例

■ public String toString()

- 返回当前对象的有关信息，用字符串表示，表达的内容因具体的对象而异
- 子类一般要重写该方法
- 录屏讲解TestToString.java



Object类常用方法举例

■ public final Class getClass()

- 返回一个Class类型的对象，该对象包含了当前对象的类信息



Object类常用方法举例

```
class Pair{  
    int x, y;  
    Pair(int x, int y){  
        this.x = x;  
        this.y = y;  
    }  
}
```

```
public class TestGetClass{  
    public static void main(String[] args){  
        Pair pair = new Pair(3, 4);  
        Class info = pair.getClass();  
        System.out.println("类名: " + info.getName());  
        System.out.println("包名: " + info.getPackage());  
        System.out.println("父类名: " + info.getSuperclass().getName());  
    }  
}
```

```
类名: Pair  
包名: null  
父类名: java.lang.Object
```