

结论：java中静态属性和静态方法可以被继承，但是没有被重写(overwrite)而是被隐藏.

原因：

1). 静态方法和属性是属于类的，调用的时候直接通过类名.方法名完成对，不需要继承机制及可以调用。如果子类里面定义了静态方法和属性，那么这时候父类的静态方法或属性称之为“隐藏”。如果你想要调用父类的静态方法和属性，直接通过父类名.方法或变量名完成，至于是否继承一说，子类是有继承静态方法和属性，但是跟实例方法和属性不太一样，存在“隐藏”的这种情况。

2). 多态之所以能够实现依赖于继承、接口和重写、重载（继承和重写最为关键）。有了继承和重写就可以实现父类的引用指向不同子类的对象。重写的功能是：“重写”后子类的优先级要高于父类的优先级，但是“隐藏”是没有这个优先级之分的。

3). 静态属性、静态方法和非静态的属性都可以被继承和隐藏而不能被重写，因此不能实现多态，不能实现父类的引用可以指向不同子类的对象。非静态方法可以被继承和重写，因此可以实现多态。

对于同时被static和final修饰的常量，必须在声明的时候就为其显式地赋值，否则编译时不通过；而只被final修饰的常量则既可以在声明时显式地为其赋值，也可以在类初始化时显式地为其赋值，总之，在使用前必须为其显式地赋值，系统不会为其赋予默认零值。