1、在子类中可以根据需要对从基类中继承来的方法进行重写。

2、重写的方法和被重写的方法必须具有相同方法名称、参数列表和返回类型。

3、重写方法不能使用比被重写的方法更严格的访问权限

4.子类抛出的异常不能超过父类相应方法抛出的异常(子类异常不能大于父类异常)

重载发生在一个类中，同名的方法如果有不同的参数列表（参数类型不同、参数个数不同或者二者都不同）则视为重载；

重载的时候，方法名要一样，但是参数类型和个数不一样，返回值类型可以相同也可以不相同。无法以返回型别作为重载函数的区分标准。

**重载（Overload）和重写（Override）的区别。重载的方法能否根据返回类型进行区分？**   
答：方法的重载和重写都是实现多态的方式，区别在于前者实现的是编译时的多态性，而后者实现的是运行时的多态性。

重载发生在一个类中，同名的方法如果有不同的参数列表（参数类型不同、参数个数不同或者二者都不同）则视为重载；

重写发生在子类与父类之间，重写要求子类被重写方法与父类被重写方法有相同的返回类型，比父类被重写方法更好访问，不能比父类被重写方法声明更多的异常（里氏代换原则）。重载对返回类型没有特殊的要求。

只要编译器可以根据语境明确判断出语义，比如在int x =f()中，那么的确可以据此却分重载方法。不过，有时你并不关心方法的返回值，你想要的是方法调用的其他效果，这时你可能会调用方法而忽略其返回值。所以，如果像下面这样调用方法：f()；此时Java如何才能判断该调用哪一个f()呢？因此，根据方法的返回值来区分重载方法是行不通的。

太容易产生岐义了。  
int Fun(int a, int b);  
long Fun(int a, int b);  
如果代码里直接运行函数，并不将返回值存储起来，那么应该调用哪个？  
Fun(1, 2);  
即使有返回值存储，反推起来也很麻烦，会产生各种岐义。