**Lombok包下的@Data**

用法：为所有字段生成getter方法，为非final字段生成setter方法，以及构造函数，一个有用的同toString方法。Hashcode和equail的实现

等价于@Getter，@Setter，@ToString，@RequiredArgsConstructor（构造器的参数包涵final和不为空），@EqualsAndHashCode.

@NoArgsConstructor:生成无参构造器。

@AllArgsConstructor:生成全参构造器。

使用方法：引入相关的maven依赖：org.projectlombok Lombok 1.16.18

添加idea中对于Lombok的支持（添加Lombok的插件）：file—Settings—Lombok插件。

**Java中的异常的使用**：

Throwable类分为两类：Error和Exception（分为IOException和RuntiemException）

其中Error和RuntimeException未检查异常，就是在编译的时候，即使没有对异常进行处理：try。。。catch或者在方法的头部调用throws，也不会报错。

throw在方法体内，在程序的执行过程中，明确的抛出一个异常，比如说一个方法体，里面有个语句：if（满足一定条件）{throw new Exception（）} 那么这个方法就有抛出异常的可能（满足if里面的条件就会抛出，）如果调用这个方法，就要对这个可能抛出异常的方法进行处理：1.使用throws方法，在方法头部声明，2.使用try。。。catch（）{}将方法进行捕获。

**创建异常类**：重写构造函数，一个默认无参构造器，一个继承父类的构造器。

将异常进行捕获：try。。catch（）{}。。。finally{}其他的代码块，只有将所以异常进行catch后，才会执行其他代码块，如果仅仅是抛出，是不会执行其他代码块的。