

1. 继承于现有的异常结构：RuntimeException 、Exception
2. 提供全局常量：serialVersionUID
3. 提供重载的构造器
4. 注意
  1. 一般地，用户自定义异常类都是 RuntimeException 的子类。
  2. 自定义异常类通常需要编写几个 重载的构造 器 。
  3. 自定义异常需要提供 serialVersionUID
  4. 自定义的异常通过 throw 抛出 。
  5. 自定义异常最重要的是异常类的名字，当异常出现时，可以根据名字判断异常类型。

```
1  class MyException extends Exception{
2      static final long serialVersionUID = -7034897193246939L;
3      public MyException(){
4
5      }
6
7      public MyException(String msg){
8          super(msg);
9      }
10     }
11     public class StudentTest {
12
13     public static void main(String[] args) {
14         try {
15             Student s = new Student();
16             s.regist(-1001);
17             System.out.println(s);
18         } catch (Exception e) {
19
20             System.out.println(e.getMessage());
21         }
22     }
23 }
24
```

```
25 class Student{
26     private int id;
27     public void regist(int id) throws Exception {
28         if(id > 0){
29             this.id = id;
30         }else{
31             throw new MyException("不能输入负数");
32         }
33     }
34
35     @Override
36     public String toString() {
37         return "Student [id=" + id + "]";
38     }
39 }
```