# 异常处理

不要自作主张处理异常。为了避免繁琐的异常声明。可以把 checked exception 转换为 unchecked exception。所以说，还是应该捕获异常，但只对异常进行转换，然后抛出。

异常继承RuntimeException,异常都抛出到controller上用AOP处理。

程序抛出异常是好事儿，越早抛出越好。不出问题，没人会关心日志。

# NPE

对于 null 检查。如果你认为有可能会出现 null 的情况，就检查；如果你认为不会出现 null，就不检查。如果你不检查 null，却抛出了 NPE 怎么办，那不是打脸吗？不是打脸，这是暴露出了程序中的问题，是好事儿。

我们的程序中经常需要检查 null（这跟我们的程序运行方式有关），也经常出现 NPE。有些人碰上 NPE 就通过检查 null 来避免程序 crash。一位老程序员坚决反对这么做。他让我们先了解为什么会出现 null。如果出现 null 是正常现象，可以通过检查 null 来避免 NPE。如果不应该出现 null 却出现了 null，那就说明程序有问题，就要解决问题，而不是通过检查 null 来掩盖问题

# javaEE编程习惯

## 异常处理

http://www.importnew.com/26858.html

不要随便加null判断（并不“健壮”，反而掩盖错误）：如果写了，就必须测试为空和不为空的两种情况

不捕获异常，将异常交给调用者处理：比起从日志中寻找错误，应在编码阶段处理，尽早让错误抛出。

## 接口定义

https://zhuanlan.zhihu.com/p/28708259

当前用户信息 都不应该出现参数里面，应该从当前会话里面获取

新增接口一般情况下应该返回新对象的id标识

接口返回格式统一，如ResultBean<T>，PageResultBean<T>

不要有复杂的输入参数

返回应该返回的数据

## Controller规范

https://zhuanlan.zhihu.com/p/28717374

1 所有函数返回统一的ResultBean/PageResultBean格式

2 ResultBean/PageResultBean是controller专用的，不允许往后传！

3 Controller做参数格式的转换，不允许把json，map这类对象传到services去，也不允许services返回json、map。

4参数中一般情况不允许出现Request，Response这些对象

5不需要打印日志，日志在AOP里打印

# 其他

一个方法的代码行数，不允许超过50行

if语句都用 // 注释

魔法数，定义成枚举