

32 | 答疑（三）：如何选择合适的异常处理方式？

2019-07-22 景霄

Python核心技术与实战

[进入课程 >](#)



讲述：冯永吉

时长 05:40 大小 5.21M



你好，我是景霄。

不知不觉中，我们又一起完成了第三大章规范篇的学习。我非常高兴看到很多同学一直在坚持积极地学习，并且留下了很多高质量的留言，值得我们互相思考交流。也有一些同学反复推敲，指出了文章中一些表达不严谨或是不当的地方，我也表示十分感谢。

大部分留言，我都在相对应的文章中回复过了。而一些手机上不方便回复，或是很有价值很典型的问题，我专门摘录了出来，作为今天的答疑内容，集中回复。

问题一：应该使用哪种异常处理方式？



code2

在代码中对数据进行检测并直接做出处理与抛出异常，在异常处理代码中进行处理，哪一种风格更有效，更优雅？

2019-07-09



第一个问题是 code2 同学的疑惑。下面这两种处理的风格，哪一种风格更有效、更优雅？

第一种，在代码中对数据进行检测，并直接处理与抛出异常。

第二种，在异常处理代码中进行处理。

其实，第一种方法，可以翻译成下面的 “if...elif...” 语句：

复制代码

```
1 if [condition1]:
2     raise Exception1('exception 1')
3 elif [condition2]:
4     raise Exception2('exception 2')
5 ...
```

而第二种方法，则对应着下面异常处理的代码：

复制代码

```
1 try:
2     ...
3 except Exception as e:
4     ...
```

这两种方法很大的一个区别是，第一种方法一旦抛出异常，那么程序就会终止；而在第二种方法中，如果抛出异常，会被程序捕获（catch），程序还会继续运行。这也是我们选择这两种方法的重要依据。当然，在实际工作中，到底使用哪一种方法，还是取决于具体的场景。

比方说，一个模块的功能是对输入进行检测，如果输入不合法，则弹出对话框进行提示，并终止程序。那么，这种情况下，使用第一种方法更加合理。

但是，如果换成一个产品的服务器端，它需要应对各种可能发生的情况，以保证服务器不崩溃。比如在连接数据库时，如果网络异常，无法连接，那就需要捕获（catch）这个异常（exception），进行记录，并同时保证其他功能不受影响。这种情况下，我们通常会选择第二种方式。

问题二：先写出能跑起来的代码，后期再优化可以吗？



夜路破晓

听很多程序员讲过，开始他们的关注点大多数是先写出能跑起来的代码，后期当优化成为他们的瓶颈和需求时再来关注代码规范之类的问题。

对于初学者而言，想要实现弯道超车，就需要下大力气把基础夯实，而代码规范正是其中重要的一项。

不可否认，担心学了半天能写漂亮但跑不起来代码的大有人在。如何权衡呢？

分享一个认知：写漂亮代码与写能跑起来的代码之间不存在因果关系，两者都是你需要花费时间精力学习的内容。

越早认识到这一点，越能合理而高效地安排自己的学习计划。

2019-07-09



3

第二个问题，夜路破晓同学提到了很多程序员传授的“经验之谈”，即先写出能跑起来的代码，后期再优化。很明显，这种认知是错误的。我们从一开始写代码时，就必须对功能和规范这两者双管齐下。

代码功能完整和规范完整的优先级是不分先后的，应该是同时进行的。如果你一开始只注重代码的功能完整，而不关注其质量、规范，那么规范问题很容易越积越多。这样就会导致产品的 bug 越来越多，相应的代码库越发难以维护，到最后不得已只能推倒重来。

我在 Facebook 工作时就遇到过这样的情况，参与过类似的项目。当时，某些功能模块因为赶时间，code review 很宽松，代码写得很不规范，留下了隐患。时间一长，bug 越来越多，legacy 越来越多。到最后，万分无奈的情况下，我们几个工程师专门立项，花了三个多月时间，重写了这一模块的代码，才解决了这个问题。

问题三：代码中写多少注释才合适？



小侠龙旋风

交接代码的时候，同事还问我，你写的代码有没有70%的注释。70%注释？写文章吗？我觉得命名规范、见名知意，这样更好吧。

2019-07-12



第三个问题，小侠龙旋风同学留言说，自己的同事要求代码中有 70% 的注释，这显然有点过了。但是反过来说，如果你的代码中没有注释或者注释很少，仅凭规范的变量名肯定是远远不够的。

通常来说，我们会在类的开头、函数的开头或者是某一个功能块的开头加上一段描述性的注释，来说明这段代码的功能，并指明所有的输入和输出。除此之外，我们也要求在一些比较 tricky 的代码上方加上注释，帮助阅读者理解代码的含义。

总的来说，代码中到底需要有多少注释，其实并没有一个统一的要求，还是要根据代码量和代码的复杂度来决定。不过，我们平常书写时，只要满足这样的规范就可以了。

另外，必须提醒一点，如果在写好之后修改了代码，那么代码对应的注释一定也要做出相应的修改，不然很容易造成“文不对题”的现象，给别人也给你自己带来困扰。

问题四：项目的 API 文档重要吗？



未来已来

除了开发过程中的规范，项目的 API 文档也挺重要的。这点对一个刚入职的员工，如何最快地理清公司项目，并对项目需求、各模块、函数功能有个大概的了解，也是很关键

2019-07-10



第四个问题，是未来已来同学的留言。他提到了项目的 API 文档的问题，这一点说得非常好，在这里我也简单介绍一下。

我在专栏中主要讲的是代码的规范问题，但很多情况下，光有规范的代码还是远远不够的。因为一个系统，一个产品，甚至一个功能模块的代码，都有可能非常复杂。少则几千行，动

辄几十万行，尤其是对于刚加入的新人来说，在 ramp up 阶段光看代码可能就是一个噩梦了。

因此，在这方面做得比较规范的公司，通常也会要求书写文档。项目的文档，主要是对相应的系统、产品或是功能模块做一个概述，有助于后人理解。以一个 service 为例，其对应的文档通常会包括下面几部分：

第一点，系统的概述，包括各个组成部分以及工作流程的介绍；

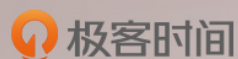
第二点，每个组成部分的具体介绍，包括必要性、设计原理等等；

第三点，系统的 performance，包括 latency 等等参数；

第四点主要说明如何对系统的各个部分进行修改，主要给出相应的 code pointer 及对应的测试方案。

这些内容，也希望屏幕前的你能够牢记。

今天我主要回答这些问题，同时也欢迎你继续在留言区写下疑问和感想，我会持续不断地解答。希望每一次的留言和答疑，都能给你带来新的收获和价值。



Python 核心技术与实战

系统提升你的 Python 能力

景霄

Facebook 资深工程师



新版升级：点击「👤请朋友读」，20位好友免费读，邀请订阅更有**现金**奖励。

上一篇 31 | pdb & cProfile：调试和性能分析的法宝

下一篇 33 | 带你初探量化世界

精选留言 (8)

写留言

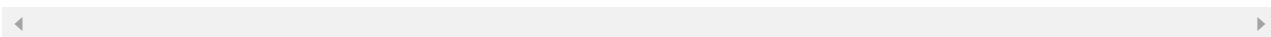


Geek_d848f7

2019-07-22

老师，结合我们这段时间的学习，有没有短小精悍的项目推荐，想学习别人怎么组织代码

作者回复: 这个你去github上找些popular的项目就可以啦



2

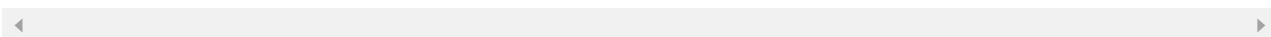


enjoylearning

2019-07-22

有没有web api的好工具推荐呢，虽然在用swagger，但感觉集成到项目侵入性太强，不优雅

作者回复: 测试的话postman就可以了。具体工程中python一般用dijando或者flask的框架



2

1

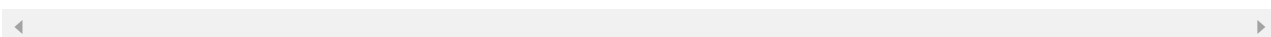


□

2019-07-22

老师，对于异常捕获选择异常条件有什么技巧吗？不太会去选择。对于服务端，我的异常打印到日志文件中，是全部异常都打印，还是筛选出具体异常的位置进行打印？

作者回复: 这个看具体需求，一般来说都会打印的，但是会downsample，比如每1000次打印一次，减小服务器压力



1

1





旗木卡卡
2019-07-27

问题二、产品设计的时候就应该同时考虑到功能规划和代码结构，设计的合理，代码结构自然也会好很多。



Michael
2019-07-24

项目中结合：isort, flake8(pylint), black(yapf), pytest, mypy 这些工具，写出来的代码在质量上肯定有不少的提升

展开 ∨



code2
2019-07-23

在C++中，构造函数中出现异常由编译器来处理，不需要程序员参与，在python重视如何处理的？由解释器处理吗？

展开 ∨



图·美克尔
2019-07-23

单元测试代码对于自己或者别人维护你之前的代码也是很重要的。



小侠龙旋风
2019-07-22

谢谢老师的问答，请问，有没有写得比较好的API案例推荐。

