

点击上方 Java后端, 选择 设为星标

优质文章, 及时送达

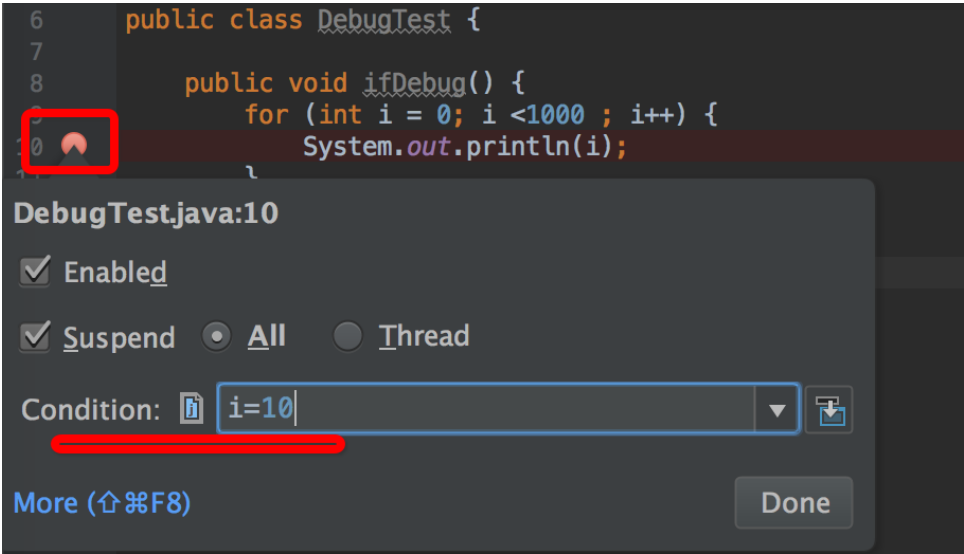
作者 | 菩提树下的杨过

来源 | www.cnblogs.com/yjmyzz

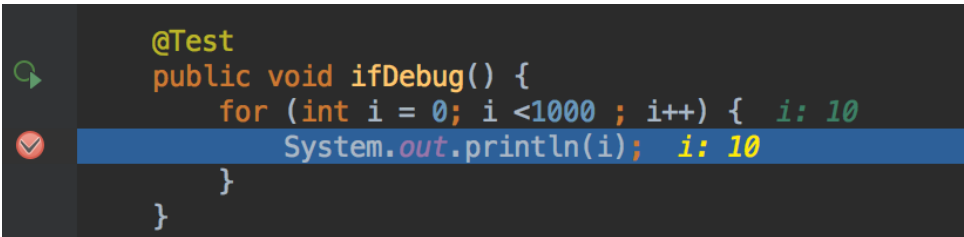
上篇 | [除了《颈椎康复指南》, 还有这 9 本书](#)

一、条件断点

循环中经常用到这个技巧，比如：遍历1个大List的过程中，想让断点停在某个特定值。

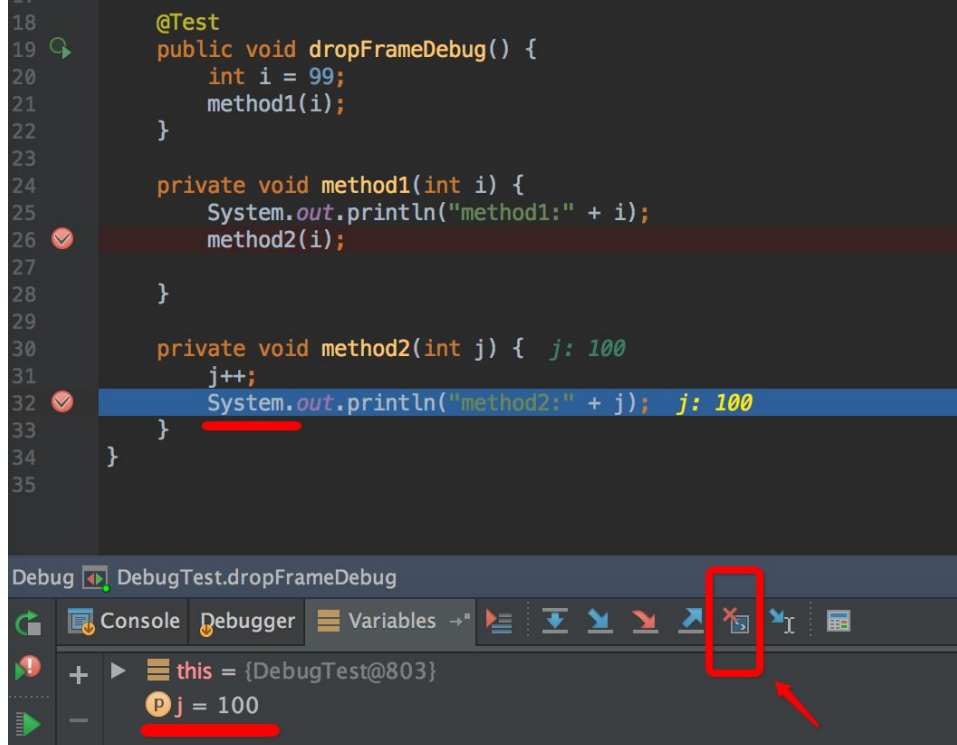


参考上图，在断点的位置，右击断点旁边的小红点，会出来一个界面，在Condition这里填入断点条件即可，这样调试时，就会自动停在i=10的位置

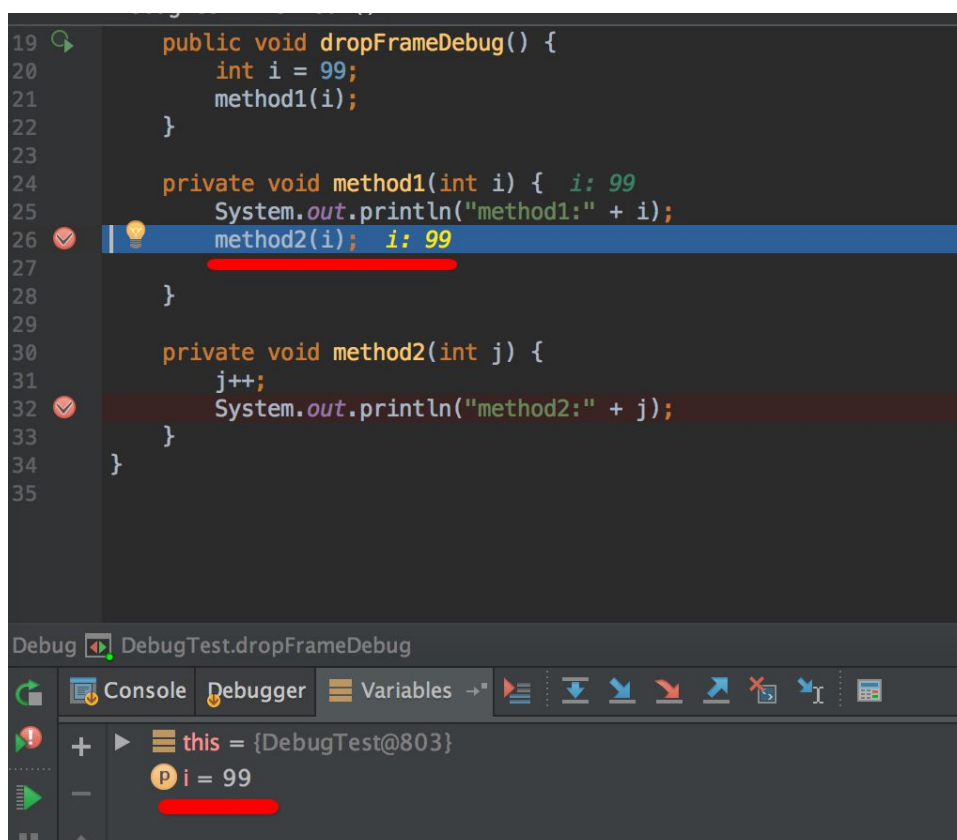


二、回到"上一步"

该技巧最适合特别复杂的方法套方法的场景，好不容易跑起来，一不小心手一抖，断点过去了，想回过头看看刚才的变量值，如果不知道该技巧，只能再跑一遍。



参考上图，method1方法调用method2，当前断点的位置j=100，点击上图红色箭头位置的Drop Frame图标后，时间穿越了



回到了method1刚开始调用的时候，变量i变成了99，没毛病吧，老铁们，是不是很6:)

注：好奇心是人类进步的阶梯，如果想知道为啥这个功能叫Drop Frame，而不是类似Back To Previous 之类的，可以去翻翻JVM的书，JVM内部以栈帧为单位保存线程的运行状态，drop frame即扔掉当前运行的栈帧，这样当前“指针”的位置，就自然到了上一帧的位置。

三、多线程调试

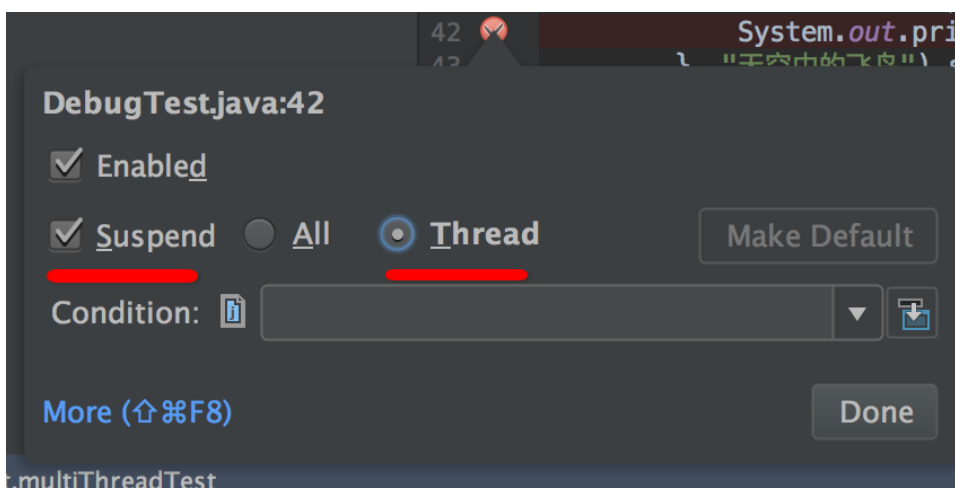
多线程同时运行时，谁先执行，谁后执行，完全是看CPU心情的，无法控制先后，运行时可能没什么问题，但是调试时就比较麻烦了，最明显的就是断点乱跳，一会儿停这个线程，一会儿停在另一个线程，比如下图：



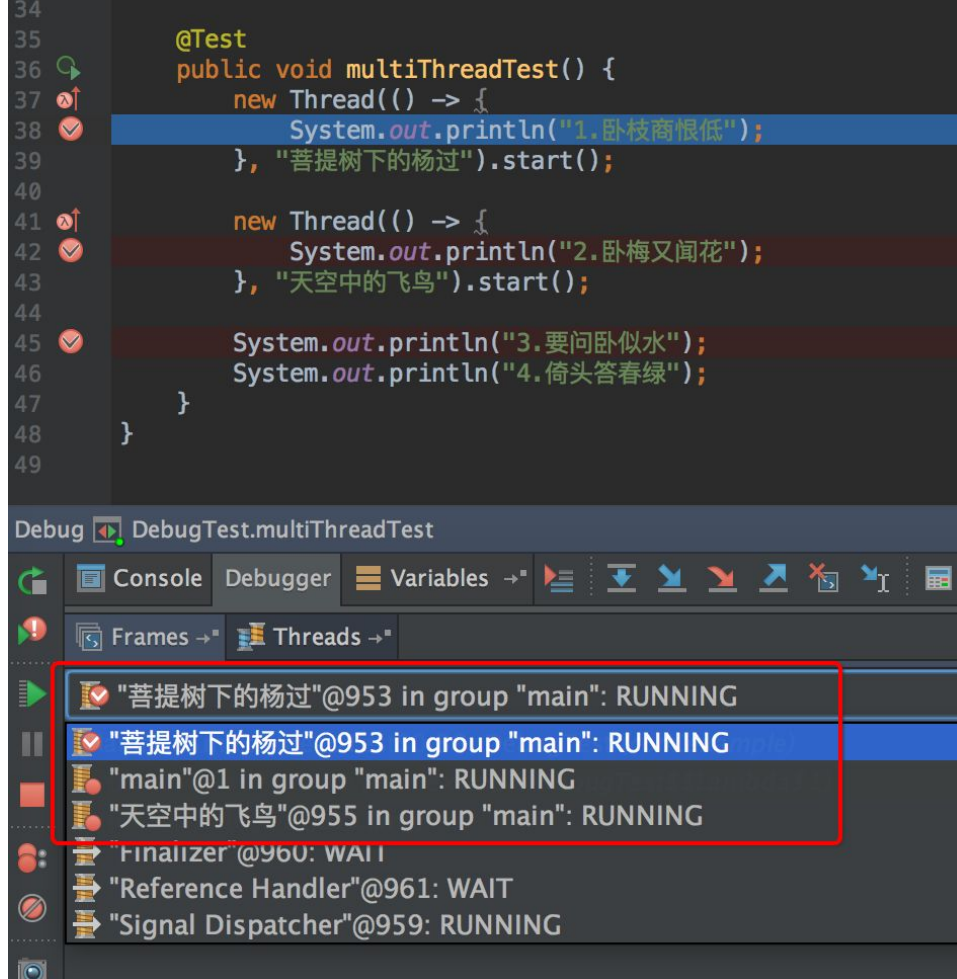
如果想希望下一个断点位置是第2句诗句，可能要失望了：



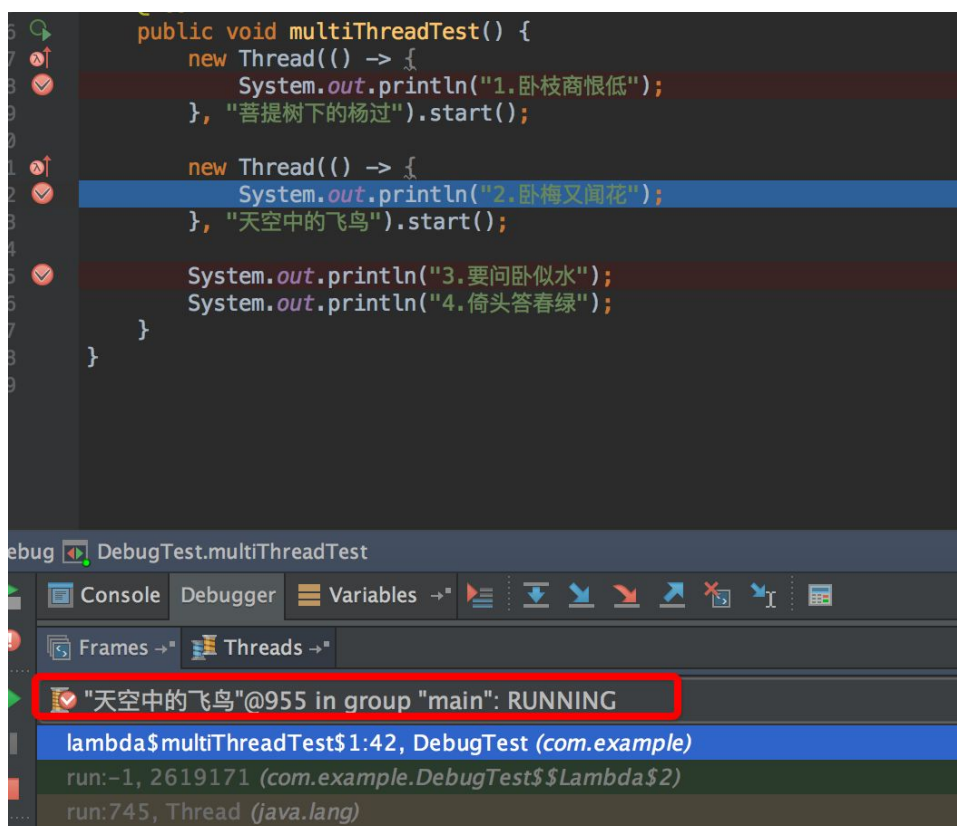
如果想让线程在调试时，想按自己的愿意来，让它停在哪个线程就停在哪个线程，可以在图中3个断点的小红点上右击，



即：Suspend挂起的条件是按每个线程来，而非All。把这3个断点都这么设置后，再来一发试试



注意上图中的红框位置，断点停下来时，这个下拉框可以看到各个线程（注：给线程起个容易识别的名字是个好习惯！），我们可以选择线程“天空中的飞鸟”



断点如愿停在了第2句诗。

四、远程调试

这也是一个装B的利器，本机不用启动项目，只要有源代码，可以在本机直接远程调试服务器上的代码，打开姿势如下：

4.1 项目启动时，先允许远程调试

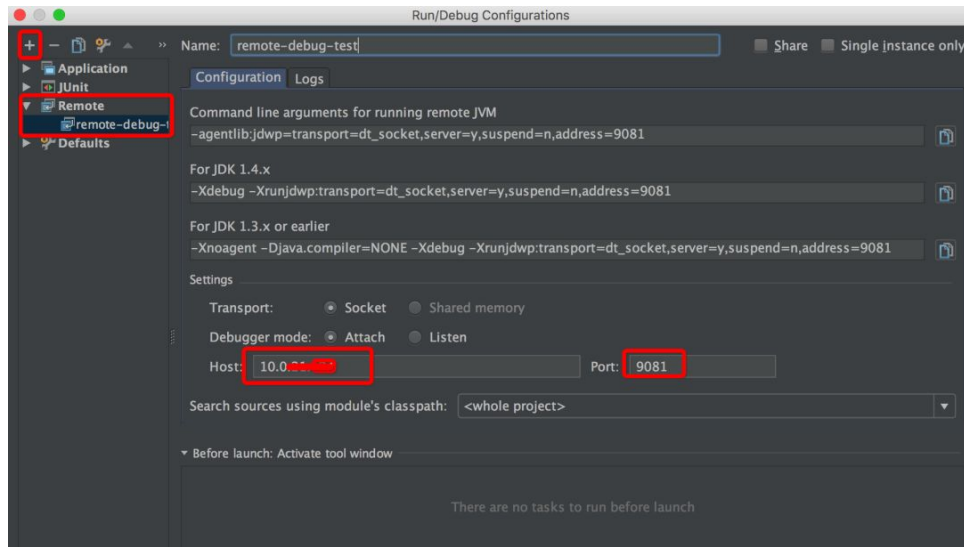
```
java -server -Xms512m -Xmx512m -Xdebug -Xnoagent -Djava.compiler=NONE -Xrunjdwp:transport=dt_socket,server=y,suspend=
```

起作用的就是

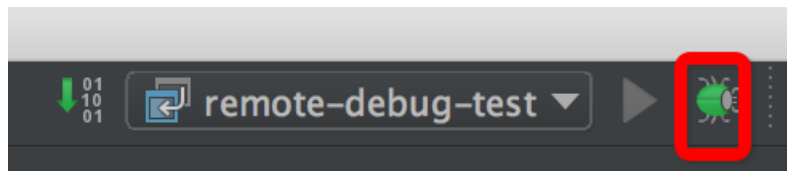
```
-Xdebug -Xnoagent -Djava.compiler=NONE -Xrunjdwp:transport=dt_socket,server=y,suspend=n,address=9081
```

注意：远程调试从技术上讲，就是在本机与远程建立socket通讯，所以端口不要冲突，而且本机要允许访问远程端口，另外这一段参数，要放在-jar 或 \${main_class}的前面

4.2 idea中设置远程调试



然后就可以调试了

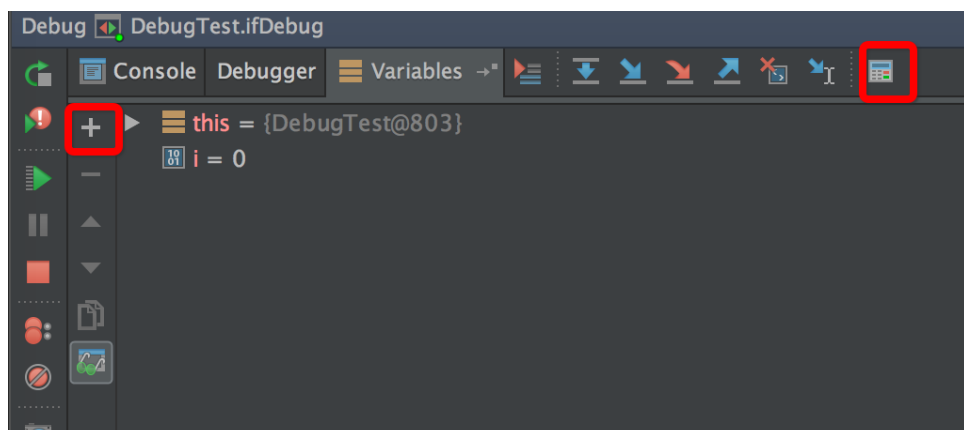


前提是本机有项目的源代码，在需要的地方打个断点，然后访问一个远程的url试试，断点就会停下来。

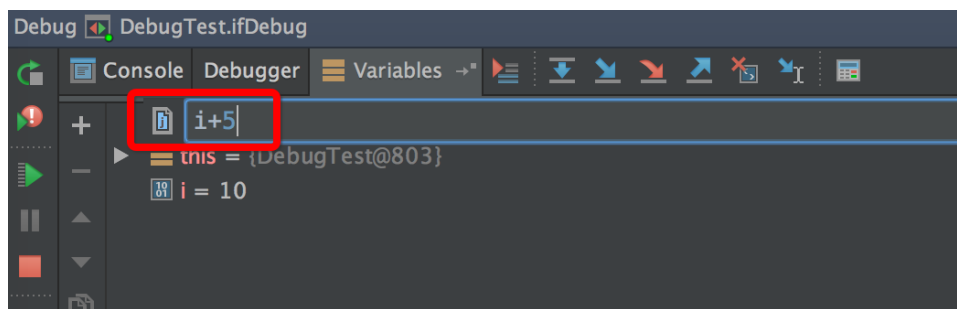
Tips：欢迎关注微信公众号：Java后端，每日技术博文推送。

五、临时执行表达式/修改变量的运行值

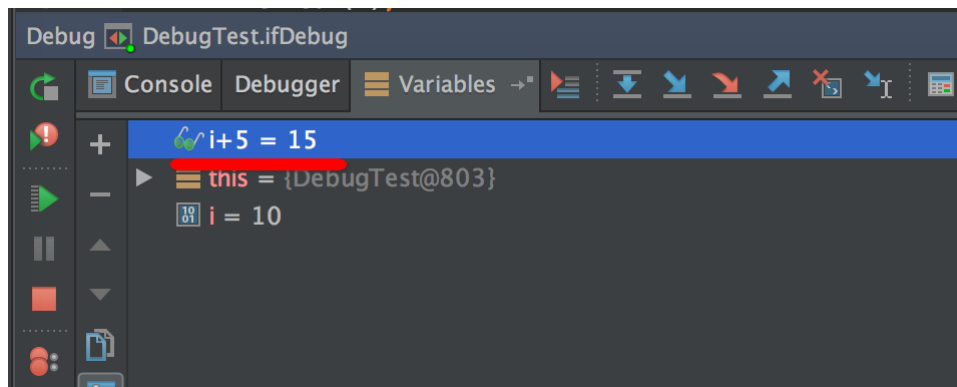
调试时，可以临时执行一些表达式，参考下图：点击这二个图标中的任何1个都可以



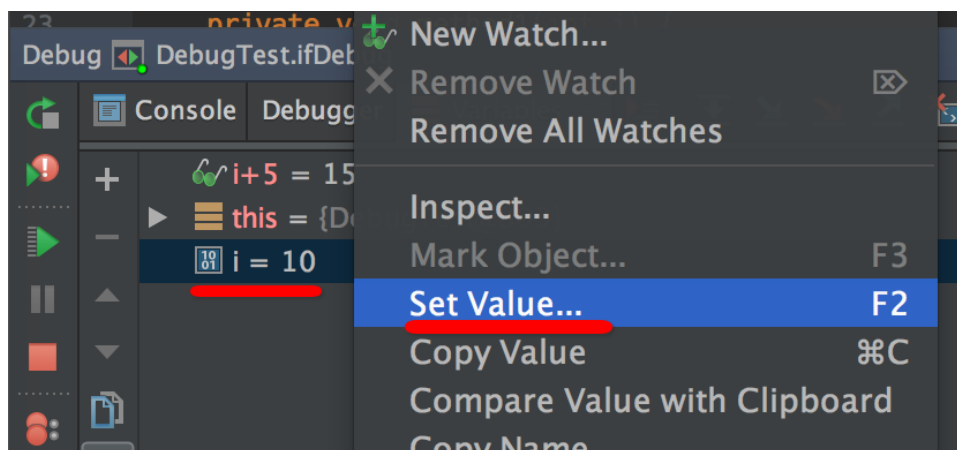
点击+号后，就可以在新出现的输入框里输入表达式，比如i+5



然后回车，马上就能看到结果



当然，如果调试时，想动态修改变量的值，很容易，在变量上右击，然后选择Set Value，剩下的事，地球人都知道。



善用上述调试技巧，相当大家撸起代码来会更有感觉，祝大家周末愉快！

- END -

如果看到这里，说明你喜欢这篇文章，请[转发](#)、[点赞](#)。微信搜索「web_resource」，关注后回复「进群」或者扫描下方二维码即可进入无广告交流群。

↓ 扫描二维码进群 ↓



推荐阅读

1. 你们心心念念的 GitHub 客户端终于来了！
2. Redis 实现「附近的人」这个功能
3. 一个秒杀系统的设计思考
4. 零基础认识 Spring Boot
5. 团队开发中 Git 最佳实践



喜欢文章, 点个在看 🍷

[阅读原文](#)

尽量避免 Bug 的一些手法

海岛 Java后端 2019-12-05

点击上方 Java后端, 选择 设为星标

优质文章, 及时送达

作者 | 海岛

来源 | blog.csdn.net/linsongbin1/article/details/84813468

最近参与了几个需求开发，BUG很少，有些需求没BUG，有些才一个BUG，搞的测试人员还发牢骚说，

大佬，你负责的项目，bug都少的可怜，叫俺怎么活？

哈哈，其实测试人员要感谢我才对，因为开发人员的代码质量高了，会极大的提升测试人员测试的速度，因为测试过程中非常顺畅，没啥阻碍的东西。设想一下，如果提测后，代码BUG满天飞，测试人员不断的提BUG单，开发人员不断的修复，一不小心还可能修复出其他BUG来呢，中间还穿插各种各样不必要的讨论，这些都严重影响了测试进度，当然也严重影响了测试人员和开发人员的心情。因此：

最好是在开发阶段就认真起来，把代码写好，以求后续流程的顺畅性。

最好是在开发阶段就认真起来，把代码写好，以求后续流程的顺畅性。

那么如何做到写代码的时候，尽量避免BUG呢？趁这个机会也跟大家分享一下我的做法。

与产品经理和经验丰富的测试人员多沟通

需求阶段

产品经理正式开需求会议之前，一般都会先把需求文档发出来，这个时候，开发人员一定要认真的看并仔细分析，每个细节都要多想想，有疑问的地方及时跟产品经理沟通。另外，看需求的时候，最好跟熟悉业务的测试人员多多沟通，测试人员是对以往需求最清楚的人，能看到其他人看不到的细节。像我自己就经常从测试人员那里，听到了一些要命的而我却忽略掉的需求细节。

代码设计阶段

我一般想好方案后，第一时间就会去找技术老大和熟悉业务的测试人员。能做到技术老大，他的思路一般都是比较广的，多听听他的意见是没错的。另外，也要去找测试人员，有些开发可能认为，技术方案怎么会去找测试人员商量呢？请不要忘记，部分测试人员是对整个公司的大部分应用和需求和业务都了如指掌的人，能清楚的知道系统与系统之间如何交互，整个链路是怎么走的，整个上下文又是怎么样的。当开发人员的设计方案出来后，表面上看起来，完美无瑕，其实可能存在影响上下游系统的隐患。而多与熟悉业务的测试人员沟通，则可以尽早把这些问题暴露出来，减少影响和损失。

代码开发阶段

必须写单元测试

单元测试的重要性，无论怎么强调都不为过。它是用于测试自己写的代码是否符合预期的极好的手段。尤其是在创业公司，需求都非常多，经常需要改代码，如果没有一套完整的单元测试来回归验证代码，分分钟由于新写了代码而破坏了原有的代码功能。

单元测试可以让开发人员放心大胆的改代码，无需担心影响之前的功能。

但是单元测试一定要认真负责的写，尽量覆盖主流程业务。那种随便写写，随便验证的单元测试，不写也罢，没啥意义，还浪费时间。写单元测试经常犯的另外一个错误是，由于急着改bug，忘记同时改单元测试了，导致之前跑过的单元测试，后面又跑不过了，这个是绝对不允许的，单元测试也必须持续维护的。

另外有一个点就是，开发人员提测后，理论上就可以接另外一个开发任务了，如果开发阶段BUG太多的话，则会影响下一个开发任务的进度。这个是开发经理不愿意看到的。

不断的重复的看自己的代码

代码提测前，要多看几次，有时候能看出一些隐藏的代码BUG的，有时候也会觉得，昨天写的代码，真垃圾，还是有蛮多代码要优化的。

在看代码的时候，最好顺便做到下面几点：

- 代码收拢性要强，不要存在那种类似的代码满天飞，能封装起来的就封装；
- 业务代码一定要有必要的准确的注释，千万别信那套方法名命名好了就能解释清楚的鬼话；
- 变量名要起好；
- 代码抽象层次要一致，不要跳跃，例如，你的业务方法，操作其他模块业务表的时候，都是调用Service类的，就不要突然冒出个直接使用xxxxxMapper去操作数据库表了；
- 流程性比较强，最好用 //1、 //2、 //3、 标注一下，这样会更加清晰。

必须做开发联调

当你的业务接口开发完成后，一定要做开发者之间的联调。联调是端对端的，就算其中一方做的再好，没有联调，还是会出问题。

开发联调通过后，建议叫产品过来提前验收

一般来说，功能测试通过后，上线前，会让产品先验收一下。但是我则喜欢开发联调完后，就先拉上产品经理，先大概验收一下。不要小看这一步，经常能提早发现一些问题的。

开发人员列出改动了哪些已有接口

列出改动细节有个好处：

让测试人员更加有针对性的做回归测试。虽说每次项目上线前，测试人员会做回归测试，但是很难做到全面回归，而没有回归到的场景，到线上可能就会造成bug。如果开发人员能列出改动点，则测试人员可以重点且认真的回归。

已有接口已经是在线上验证过的接口，如果改动了，是一定要做兼容性测试的，既要保证新功能可用，也要保证不影响旧功能。

尽最大努力,保证开发提测不delay

对于那种上线日期已经定了，一般会采用倒排的方式，推导出，开发哪个时间点提测，测试人员什么时候介入测试，测试多少天等，都会安排好。如果开发提测delay了，留给测试人员的测试时间就缩短了，会给测试人员造成很大的压力，压力一大，则更容易出错，直接影响测试质量，也就影响了上线质量。

当出现了提测delay的苗头，开发人员一定要及时反馈出来，由组长或者经理协调资源，进行补救。这里要重点强调的是，一个功能的提测，是涉及到前端、后端的，你想自己加班到深夜赶进度也是没用的，一定要以最快的速度，将问题暴露出来，由上级

去协调一下，留下相关的人一起奋斗一下，尽量保证按时提测。

测试人员测试阶段-看日志

不要以为提测后，就没自己啥事了，最好还是抽少许时间，去测试机器上看看日志，观察和分析一下入参和出参等，看看有没有什么异常或者不合理的数据。

【END】

如果看到这里，说明你喜欢这篇文章，请[转发、点赞](#)。微信搜索「web_resource」，关注后回复「进群」或者扫描下方二维码即可进入无广告交流群。

↓ 扫描二维码进群 ↓



推荐阅读

1. 一键下载 Pornhub 视频！
2. 一个女生不主动联系你还有机会吗？
3. 程序员开发了一款软件，完成了舔狗的绝地反杀
4. Spring Boot 多模块项目实践(附打包方法)
5. 团队开发中 Git 最佳实践



喜欢文章, 点个在看 

[阅读原文](#)

声明: pdf仅供学习使用, 一切版权归原创公众号所有; 建议持续关注原创公众号获取最新文章, 学习愉快!