



知乎回答：编程软件界面的文本编辑器是如何制作的

2018, May 17 by Tesla Ice Zhang

代码编辑器比起普通的文本编辑器，抛开文件管理、项目管理、各种外部工具集成不谈，只看编辑器本身的话，无外乎多了这么几个功能：

- 语法高亮
- 代码补船
- 格式化代码
- 重构（大部分『编辑器』不具备这个能力）

语法高亮很好实现，无非就是带 Attribute 的文本，或者，最简单的，带色彩的文本。各种 GUI 平台都提供了绘制这种东西的功能，比如落后的 Swing, 不知道落后不落后的 HTML/CSS, 先进的 JavaFX, 以及各种平台的自绘控件（应该是最好的，这种专门为代码编辑器定制的文本编辑器控件），都提供了。

代码补船很好实现，无非就是弹个里面装着一个列表控件的补全框框，处理下 focus（上下、Tab/Enter/Esc 作用于补全框框，其他按钮作用于编辑器），然后插入文本。补全框框的本质和右键

菜单那玩意是一样的，只是一个由写代码触发、一个由右键触发而已。处理 focus 和插入文本都是最基本的功能。

格式化代码很好实现，无非就是把各种格式化的命令行工具做的事情交给编辑器了而已。

重构，啊。

相信你看完这段文字对文本编辑器的实现应该有个初步印象了，是不是觉得『哇塞真JetBrains简单，我也能实现一个』，那么我来告诉你真相吧。

上面说的『很好实现』其实每个背后都隐藏着极其复杂的细节，比如：

首先重中之重，上面直接无视了一个非常重要的话题——代码分析。代码编辑器需要一个能对语法树和AST元信息进行高性能增量更新的分析器，才能确定『有哪些符号可以被补船出来』、『哪些 token 需要被高亮为哪些颜色』、『格式化的时候需要在哪些地方插空格/tab/换行』等信息。

补全和高亮的细节都是很繁琐的，先略过不提，以后在写。

格式化代码，写命令行工具也不容易，比如 gofmt/dartfmt 就做的不尽人意（关于表达式中的空格问题处理的不够细致），

ktlint的格式化也是垃圾（不想说了），Julia 的

DocumentFormat.jl（测试都没过，运行速度爆慢，可能等以后编译成可执行文件很方便后才可以改善）就是屎，其他没用过。

这个东西的错误处理也非常非常不好写，比如我们看看

dartfmt，一旦遇到有语法错误的代码就歇菜了，格式化出来就

是一坨稀粑粑，这种东西要是集成到 IDE 里面，写到一半，强迫症发作了手贱格式化一下，代码就变成稀粑粑了（IntelliJ 的 Dart 插件就真的是这样的，摊手。Java 那边就好得多）。这些就是编辑器开发中的难点了（和 txt, word 不一样的地方）。

[原文链接](#)

感觉写的还可以，也比较长，就转到博客了。

Tweet this 

Top

[创建一个 issue](#) 以申请评论

[Create an issue](#) to apply for commentary

协议/License

本作品 知乎回答：编程软件界面的文本编辑器是如何制作的 采用 [知识共享署名-非商业性使用-禁止演绎 4.0 国际许可协议](#) 进行许可，基于 <http://ice1000.org/2018/05/17/ZhihuAnswer/> 上的作品创作。

This work is licensed under a [Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License](#).



© 2017 Tesla Ice Zhang

