

Smart IDE 需求文档:

访谈信息:

1. 举出一些具体的场景，并说明在这个场景下如何能体现出“Smart”

(1) 大多数现有的补全功能都只能局限在关键词推断，比如说输入“fo”，补全出“for”。这在关键词很长的时候比较有用，比如类名或者变量名比较长的时候很方便，但更多的关键词是比较短的，比如 for, if, 如果只能推断这个关键词本身的话补全功能是比较没用的。所以如果输入 for，能自动补全出 `for(int *** = ***; *** < ***; ***) { }`，其中***是预留出来的空缺，用户可以通过 tab 键在空缺之间切换，这样就比较“Smart”了

(2) 用户在创建新项目时经常需要使用现成的框架，而不是每次都从空项目开始。因此在创建项目时如果能让用户选择一些常用的框架，IDE 自动为用户创建好相应的文件和代码，就可以减轻开发负担。(事实上现在的许多 IDE 都有这种功能)

(3) 用户在进行版本控制时往往是依赖 IDE 外部的工具，比如 git，写完代码后需要打开命令行执行一些 git 命令，其实是不必要的。IDE 内完全可以内置一些常用的版本控制脚本，让用户一键执行。至于应该集成哪些脚本，就是可以体现出“Smart”的地方了。

(4) 用户在使用 IDE 写代码时，如果对语言或框架有些问题，需要去找开发文档。如果 IDE 能内置开发文档，那就非常方便了。更具体地说，应该是开放一个接口，让框架的开发者能将开发文档嵌入到 IDE 中，这样用户在遇到不知道如何使用一些 API 时，通过一些简单的快捷键，比如 Ctrl+Alt+左键点击这个 API 的名字，就能找出该 API 对应的文档。

(5) 一些程序的开发者可能在设计程序框架时会先画出 UML 图，根据 UML 图确定设计问题后才进行开发。那么如果 IDE 能够帮助开发者绘制 UML 图，甚至根据该 UML 图自动生成相应的类名以及继承关系，那就更加 Smart 了

(6) 一些成熟的语言，比如 Java，有诸如 Javadoc 这样成熟的注释规范。那么 IDE 完全可以自动为代码生成相应的注释框架，比如 IDEA 安装一些插件后就可以自动生成 Javadoc 格式的注释。

记录员：李祐德