



南開大學
Nankai University

网络空间安全学院
软件工程实验报告

智慧编程工具软件调研

姓名：孙露

学号：2112060

专业：信息安全

2024 年 3 月 17 日

目录

| | |
|-------------------------------------|-----------|
| 1 作业内容 | 3 |
| 1.1 题目 | 3 |
| 1.2 作业要求 | 3 |
| 2 CodeGeeX | 3 |
| 2.1 软件概述 | 3 |
| 2.2 不同 IDE 的使用 | 5 |
| 2.2.1 VSCode | 6 |
| 2.2.2 JetBrains IDEs (以 PyCharm 为例) | 7 |
| 2.3 软件功能 | 8 |
| 2.3.1 代码生成 | 8 |
| 2.3.2 代码补全 | 9 |
| 2.3.3 代码翻译 | 10 |
| 2.3.4 注释生成 | 11 |
| 2.3.5 智能问答 | 13 |
| 2.3.6 单元测试 | 14 |
| 2.4 目标用户与潜在用户 | 14 |
| 2.5 结合使用体验的一些优缺点 | 15 |
| 2.5.1 缺点及优化 | 15 |
| 2.5.2 优点及特色 | 16 |
| 2.6 更新 CodeGeeX2 | 17 |
| 3 通义灵码 | 17 |
| 3.1 软件概述 | 18 |
| 3.2 不同 IDE 的使用 | 18 |
| 3.2.1 VSCode | 18 |
| 3.2.2 JetBrains IDEs (以 Pycharm 为例) | 19 |
| 3.3 软件功能 | 20 |
| 3.3.1 代码生成与补全 | 20 |
| 3.3.2 注释生成 | 21 |
| 3.3.3 代码问答 | 23 |
| 3.3.4 代码优化 | 24 |
| 3.3.5 单元测试 | 25 |
| 3.4 目标用户与潜在用户 | 26 |
| 3.5 结合使用体验的一些优缺点 | 27 |
| 3.5.1 优点及特色 | 27 |
| 3.5.2 缺点及优化 | 28 |
| 4 Codeium | 30 |
| 4.1 软件概述 | 30 |
| 4.2 不同 IDE 的使用 | 30 |

| | | |
|----------|-------------------------------|-----------|
| 4.2.1 | VSCode | 32 |
| 4.2.2 | JetBrains IDEs (以 Pycharm 为例) | 33 |
| 4.2.3 | Chrome 插件 | 34 |
| 4.3 | 软件功能 | 35 |
| 4.3.1 | 代码生成与补全 | 35 |
| 4.3.2 | 注释生成 | 36 |
| 4.3.3 | 智能问答 | 37 |
| 4.3.4 | 代码搜索 | 38 |
| 4.3.5 | 代码优化 | 39 |
| 4.3.6 | 单元测试 | 40 |
| 4.4 | 目标用户与潜在用户 | 41 |
| 4.5 | 结合使用体验的一些优缺点 | 41 |
| 4.5.1 | 优点及特色 | 41 |
| 4.5.2 | 缺点及优化 | 42 |
| 5 | 未来 | 42 |
| 5.1 | 如何提升竞争力 | 42 |
| 5.2 | 设计的产品功能 | 43 |
| 5.3 | 我的开发 | 44 |

1 作业内容

1.1 题目

2023 年时软件工程智能化进程的重大拐点，大模型与软件工程各环节深度融合，新一代智能化软件工程 (AI4SE) 进入快速发展期，持续涌现出以代码大模型为首的三类大模型：代码大模型具有强大的代码理解、代码生成、代码检查、代码优化、研发问答等能力；测试大模型重点关注智能单测、测试脚本生成、测试数据生成等方面；运维大模型关注运维知识问答、工单处理等方面，未来将从故障识别、故障预测、运维安全等维度推动 AIOps 全过程智能水平提升。你是否了解现有的智慧编程助手？未来在你编写代码时，是否会考虑使用代码模型辅助完成开发？请比较至少 3 个智慧编程工具，列出它们的优缺点和特色，以及改进思考。

1.2 作业要求

软件调研要求：

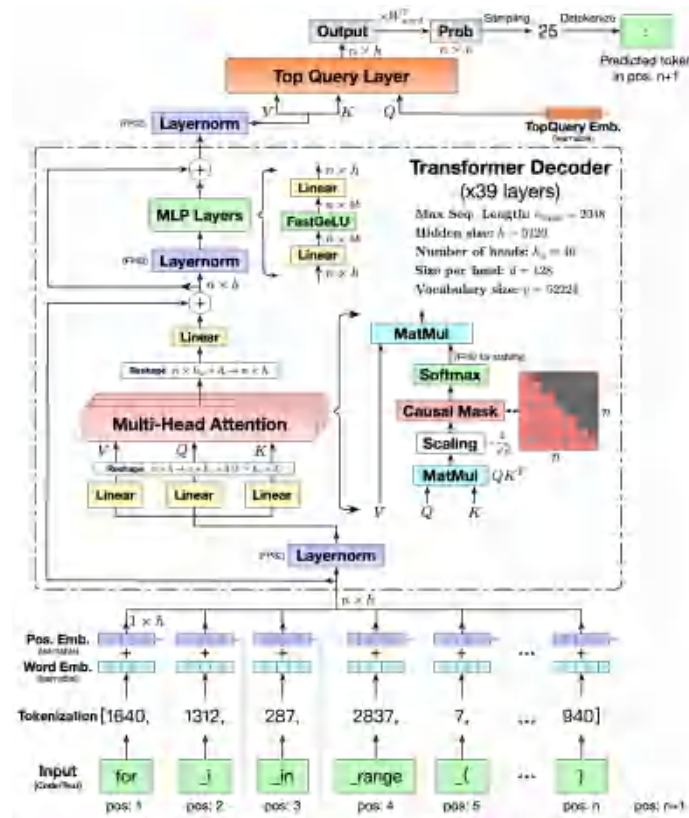
1. 注册/下载并使用 < 被测智能编程助手 > 的主要功能，从一些评估角度对待测软件进行评分，如：不同 IDE 下使用体验，代码生成质量，代码可读性，安全性等。
2. 根据你对 < 被测智能编程助手 > 的了解，分析这个软件竞争力和优劣势 (和类似软件相比)，指出你认为这个软件的核心目标用户和潜在目标用户？
3. 如果你是新上任的项目经理，对于 < 被测智能编程助手 > 你有什么见解，如何提高从而在竞争中胜出？以产品功能为例：你还需要设计什么样的功能？为何要做这个功能，而不是其它功能？
4. 如果你来开发一款新的智能辅助编程工具，你会以什么形式开发这款产品？你会怎么设计这款产品？

2 CodeGeeX

2.1 软件概述

CodeGeeX 是清华大学知识工程实验室研发的一款基于大模型的全能的智能编程助手，它可以实现代码的生成与补全、自动添加注释、代码翻译以及智能问答等功能。

CodeGeeX 是一个具有 130 亿参数的多编程语言代码生成预训练模型。CodeGeeX 采用华为 MindSpore 框架实现，在鹏城实验室“鹏城云脑 II”中的 192 个节点（共 1536 个国产昇腾 910 AI 处理器）上训练而成。截至 2022 年 6 月 22 日，CodeGeeX 历时两个月在 20 多种编程语言的代码语料库 (>8500 亿 Token) 上预训练得到。



CodeGeeX 使用纯解码器的 GPT 架构，并使用自回归语言建模。CodeGeeX 的核心架构是 39 层的 transformer 解码器。在每个 transformer 层，包含多头自注意力机制、MLP 层、layer normalization 和残差连接。使用类 GELU 的 FaastGELU 激活，其在 Ascend 910 AI 处理器上更加高效。

采用 GPT 的范式，在大规模无监督代码数据上训练模型。原始的 GPT 使用 pooler 函数来获得最终的输出，CodeGeeX 在所有 transformer 层上添加一个额外的查询层获得最终的 embedding。top query 层的输入被替换为位置 n+1 的 query embedding。最终的输出乘以词嵌入矩阵的转置来获得输出概率分布。对于解决策略，CodeGeeX 支持贪心、温度采样、top-k 采样、top-p 采样和 beam search。

训练语料包含两个部分。第一部分是开源代码数据集：Pile 和 CodeParrot。第二部分是直接从 GitHub 爬取的 Python、Java 和 C++ 代码，用于补充第一部分。为了帮助模型区分多种语言，在每个片段前添加了语言相关的标签。考虑到代码数据中有大量的自然语言注释且变量、函数、类别的命名通常也是有意义的单词，因此将代码数据也做文本数据并使用 GPT-2 tokenizer。

下面的图展示了 CodeGeeX 在代码生成和代码翻译任务上的测试结果

Table 3: Results of code generation task in HumanEval-XL

| Language | Metric | GPT-3.5 -0.1 | GPT-NeoX -200 | InCoder -0.7B | CodeGen -Multi-0.1 | CodeGen -Multi-1.6B | CodeGeeX -13B (ours) |
|------------|----------|-----------------|------------------|------------------|-----------------------|------------------------|-------------------------|
| Python | pass@1 | 11.10% | 13.33% | 16.41% | 19.41% | 19.22% | 22.89% |
| | pass@10 | 14.07% | 22.02% | 26.55% | 50.24% | 34.64% | 39.57% |
| | pass@100 | 20.98% | 38.56% | 43.95% | 40.05% | 55.17% | 60.92% |
| C++ | pass@1 | 2.54% | 9.40% | 6.50% | 11.44% | 18.65% | 17.06% |
| | pass@10 | 13.67% | 18.69% | 19.50% | 29.75% | 50.84% | 32.21% |
| | pass@100 | 20.14% | 36.39% | 26.02% | 42.62% | 50.90% | 51.00% |
| Java | pass@1 | 7.88% | 8.87% | 9.03% | 15.17% | 14.95% | 20.04% |
| | pass@10 | 18.37% | 19.55% | 18.04% | 31.74% | 36.73% | 36.70% |
| | pass@100 | 32.96% | 42.25% | 40.70% | 53.91% | 60.62% | 50.22% |
| JavaScript | pass@1 | 8.99% | 11.26% | 12.98% | 15.41% | 18.40% | 17.59% |
| | pass@10 | 10.22% | 20.78% | 22.48% | 27.32% | 32.80% | 42.20% |
| | pass@100 | 35.77% | 42.67% | 43.34% | 48.11% | 56.40% | 56.33% |
| Go | pass@1 | 4.01% | 5.00% | 6.89% | 9.06% | 13.05% | 14.43% |
| | pass@10 | 10.81% | 15.70% | 13.80% | 25.26% | 25.46% | 25.68% |
| | pass@100 | 23.70% | 32.08% | 28.51% | 41.01% | 48.77% | 47.14% |
| Average | pass@1 | 7.90% | 9.76% | 11.57% | 14.78% | 16.77% | 18.40% |
| | pass@10 | 14.79% | 19.55% | 20.21% | 27.69% | 32.09% | 33.29% |
| | pass@100 | 20.32% | 30.05% | 28.48% | 42.24% | 50.59% | 54.76% |

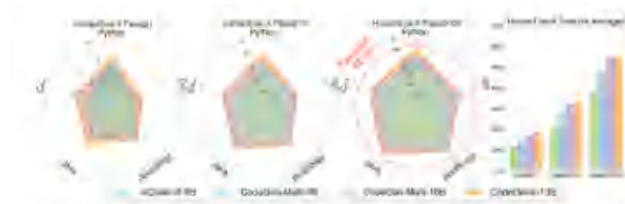


Figure 6: Results of code generation task in HumanEval-X. Left: Detailed pass@k performance in five languages. Right: CodeGeeX achieves the highest average performance compared with other open-sourced multilingual baselines. We also find that it gains performance when the sampling budgets are properly distributed to multiple languages.

Table 6: Results of code translation task in HumanEval-X.

| Model | Language | Target Language | | | | | | | | | |
|----------|----------|-----------------|------|------|------|------------|------|------|------|------|-------|
| | | Python | Java | C++ | C | JavaScript | Go | PHP | Perl | Ruby | Other |
| GPT-4 | Python | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | Java | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | C++ | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | C | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| GPT-3.5 | Python | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | Java | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | C++ | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | C | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| CodeGeeX | Python | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | Java | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | C++ | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | C | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |

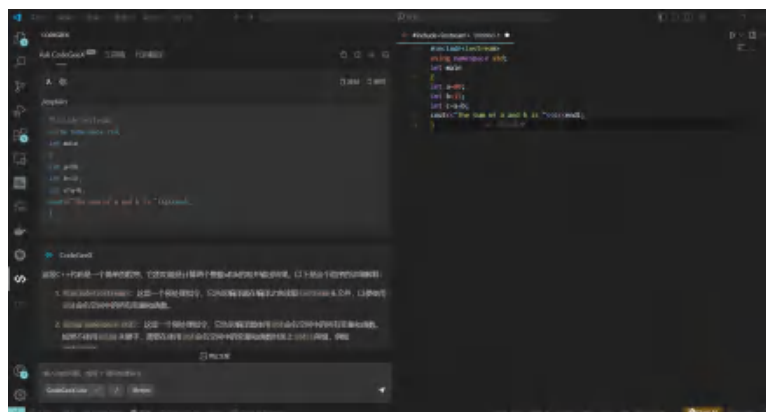
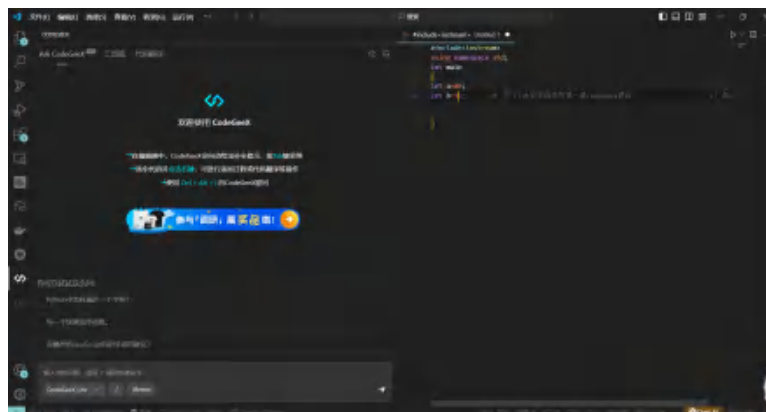
2.2 不同 IDE 的使用

根据 CodeGeeX 网站的介绍,CodeGeeX 支持多种主流 IDE,如 VS Code、IntelliJ IDEA、PyCharm、Vim 等,同时,支持 Python、Java、C++/C、JavaScript、Go 等多种语言。

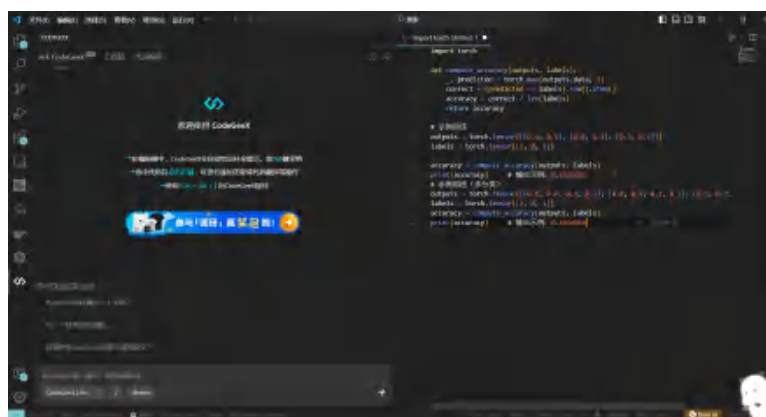


2.2.1 VSCode

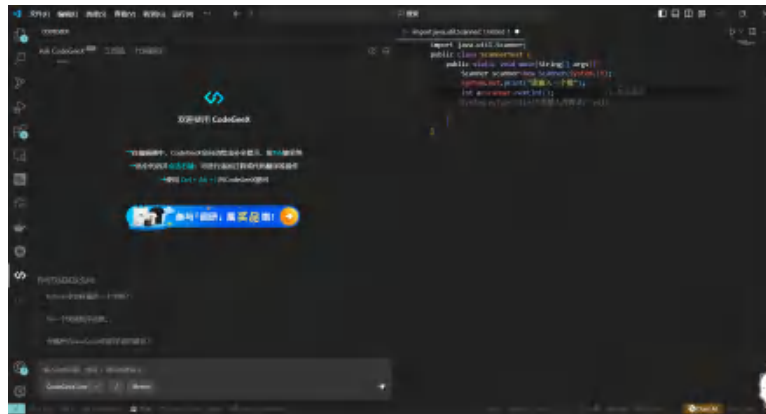
a. C++



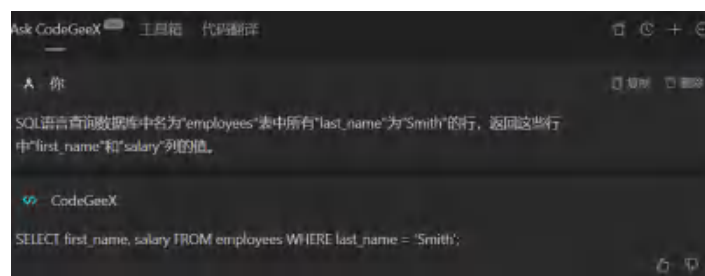
b. Python



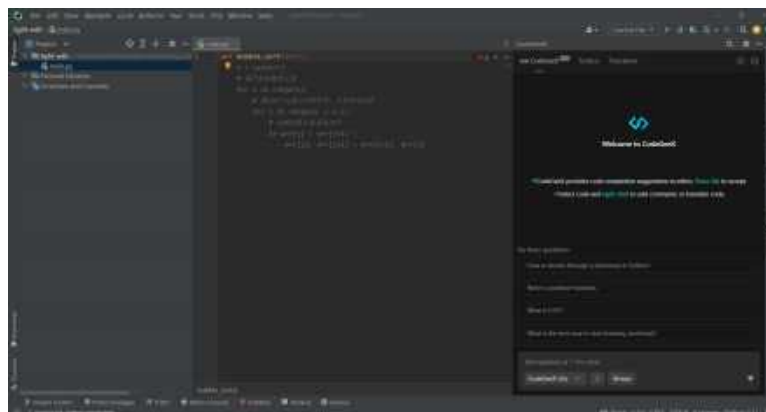
c. Java



d. SQL



2.2.2 JetBrains IDEs (以 PyCharm 为例)



CodeGeeX 插件支持四种使用模式：自动模式、交互模式、翻译模式和提示模式。

1、自动模式在该模式中，如您不想继续重复代码的编写，可以借用插件敲下 tab 键，它会自动生成代码。

具体操作步骤：

CodeGeeX 将在停止输入时，从光标处开始生成（右下角 CodeGeeX 图标转圈表示正在生成）。生成完毕之后会以灰色显示，如果对结果满意，按“Tab”即可插入生成结果。

在生成多个候选的情况下，可以使用“Alt/Option+[“或”]”在几个候选间进行切换。用户也可以在设置中改变“Candidate Num”，还可使用“Alt/Option+N”获得新的候选。

2、交互模式在该模式中，可以通过敲入一行注释，使插件自动生成一段完整的代码解决方案。

具体操作步骤：

通过“Ctrl+Enter”激活交互模式，CodeGeeX 将生成“X”个候选，并显示在右侧窗口中（“X”数量可以在设置的“Candidate Num”中修改）。点击候选代码上方的“use code”即可插入结果为当前光标位置。

3、翻译模式该模式可帮助将一种语言翻译成另一种语言，例如将用 C++ 写的某个算法翻译成 Python，而不用自己再敲一遍。

具体操作步骤：

在当前语言的文本编辑器中输入或者粘贴一段代码，选择代码，然后点击“Ctrl+Alt+T”激活翻译模式，根据提示选择要翻译的目标语言。CodeGeeX 将自动识别当前编辑器中代码的语言，并进行翻译，点击翻译结果上方的“use code”即可将翻译结果插入文件。您还可以在设置中选择插入时如何处理被翻译的代码——注释或者覆盖它们。

4、提示模式该模式可模仿已输入代码实现相应功能，如快速对大段代码进行解释。

该模式的原理是利用 CodeGeeX 强大的少样本生成能力。可以在输入中添加额外的提示来实现一些有趣的功能，包括且并不限于代码解释、概括、以特定风格生成等。如可以在自定义模板中提供一段逐行解释代码的例子，选择想要解释的代码，“Alt/Option+T”触发提示模式，选择写好的模板，CodeGeeX 将会解释输入的代码。

2.3 软件功能

根据 CodeGeeX 网站的介绍，CodeGeeX 主要功能包括代码生成、代码补全、代码翻译、注释生成、智能问答 5 个部分。



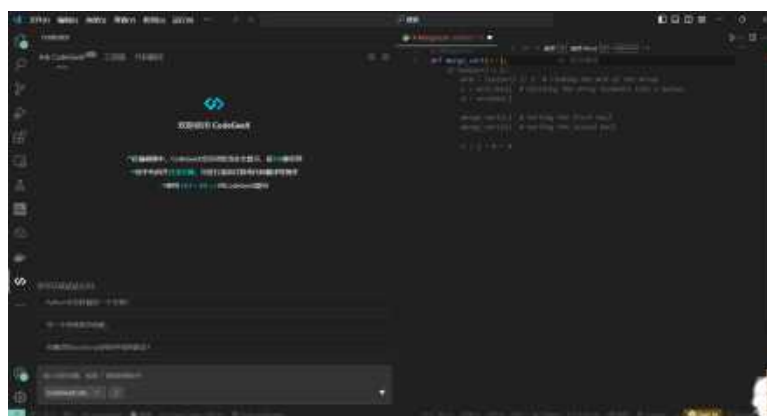
2.3.1 代码生成

CodeGeex 处于激活状态时，停止输入内容后，就会从光标处开始生成内容，生成待选内容后会以灰色显示，按 Tab 键即可插入生成结果。

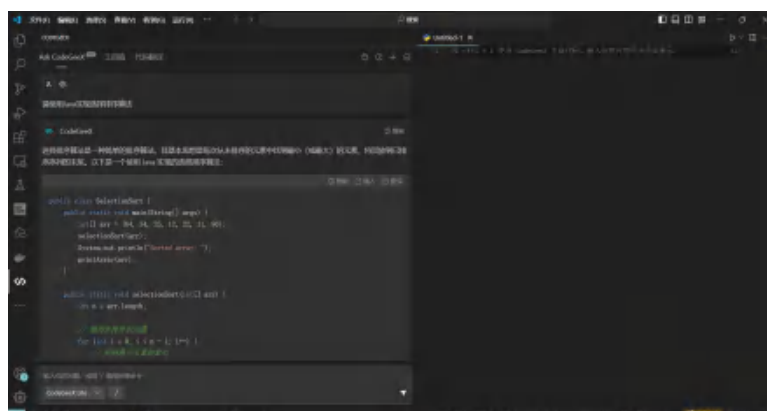
按 Ctrl+ 回车键还可以激活交互模式，可以在右侧窗口显示生成的 x 个候选片段，其中 x 数量可以在设置的 Candidate Num 中修改，最多可以生成 3 个候选项，数目越多生成越慢，点击候选片段上的 use code 即可插入使用。



(1) 生成归并排序的 python 代码

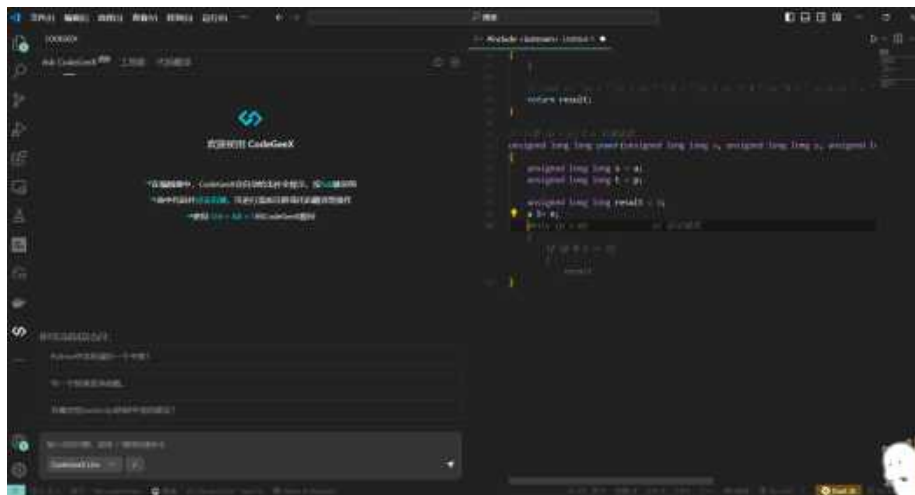


(2) 生成选择排序的 Java 代码



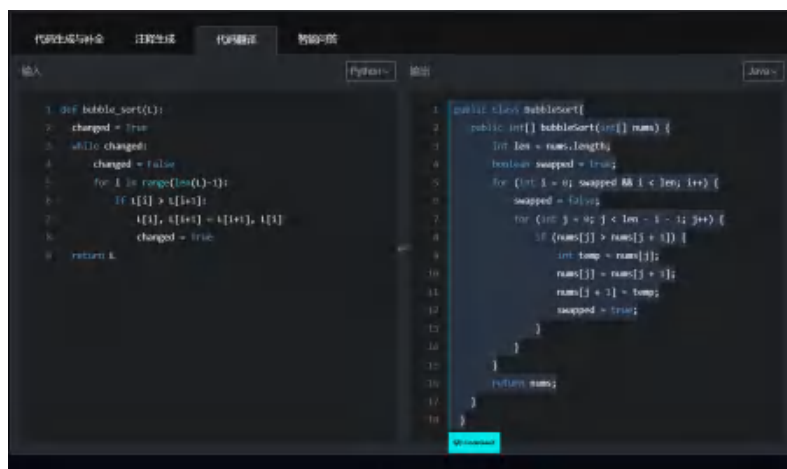
2.3.2 代码补全

(1) 补全计算 $(a^p) \% n$ 的幂函数的 C++ 代码

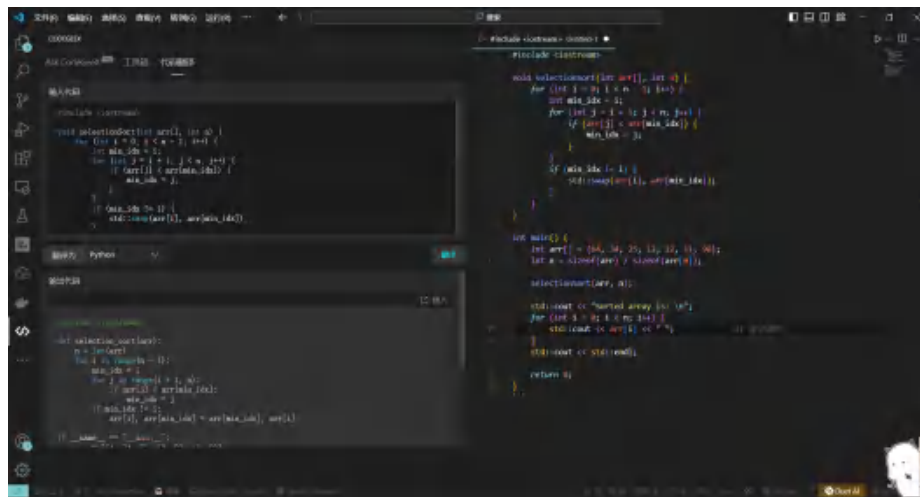


2.3.3 代码翻译

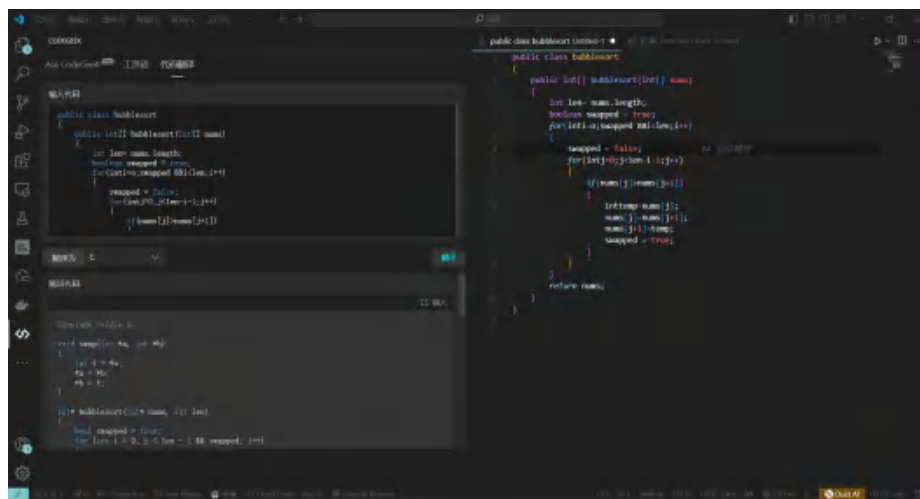
Windows 系统是 Ctrl+Alt+T, Mac 系统是 Ctrl+Option+T 激活翻译模式, CodeGeeX 会把该代码翻译成匹配当前编辑器语言的代码。点击翻译结果上方的 use code 插入。还可以在设置中选择希望插入的时候如何处理被翻译的代码, 可以选择注释它们或者覆盖它们。



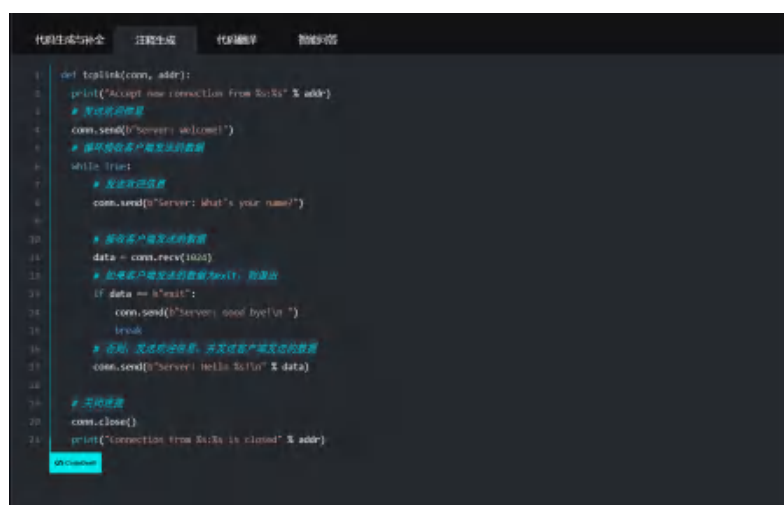
(1) 将 C++ 的选择排序翻译为 Python 代码



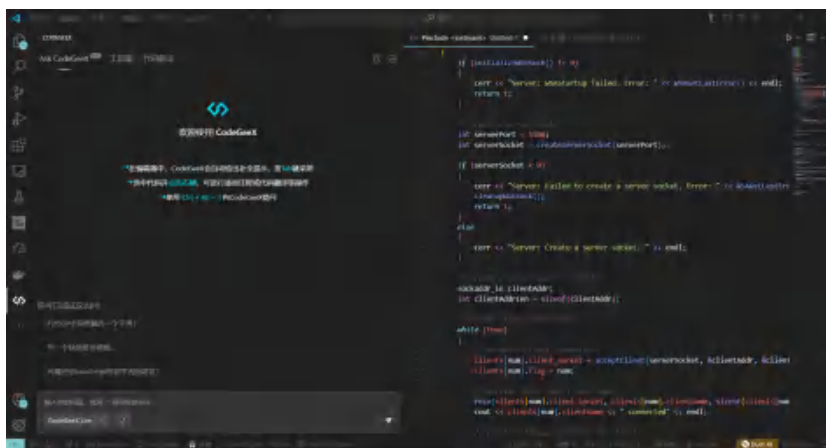
(2) 将 java 的冒泡排序翻译为 C 代码



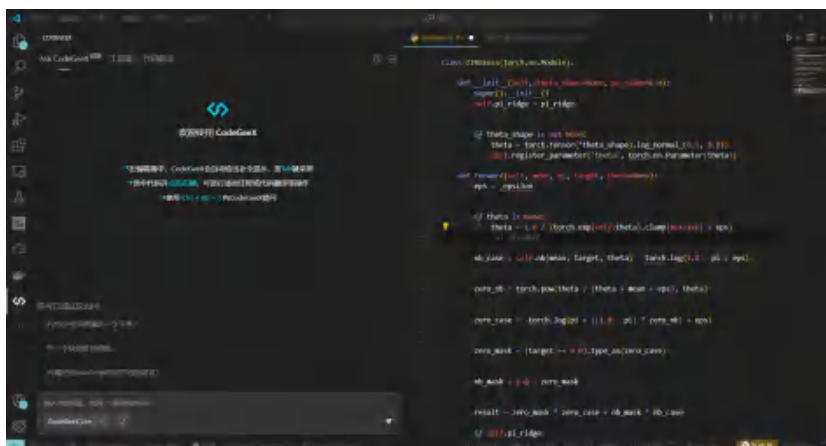
2.3.4 注释生成

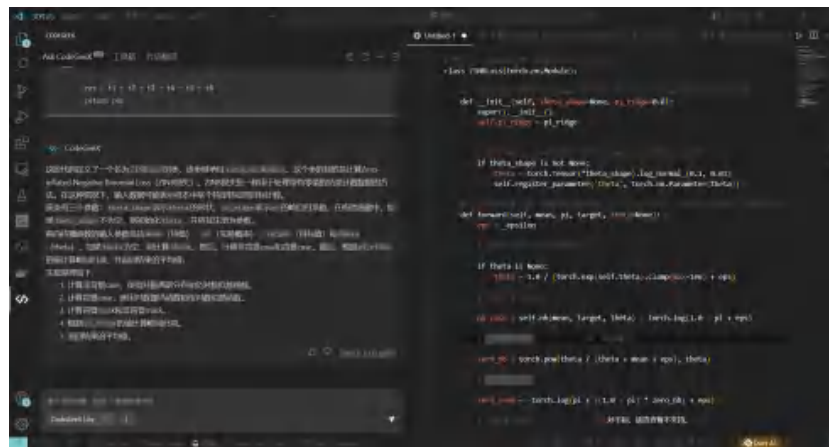


(1) 生成下面用于多人聊天室客户端 C++ 代码的英文注释

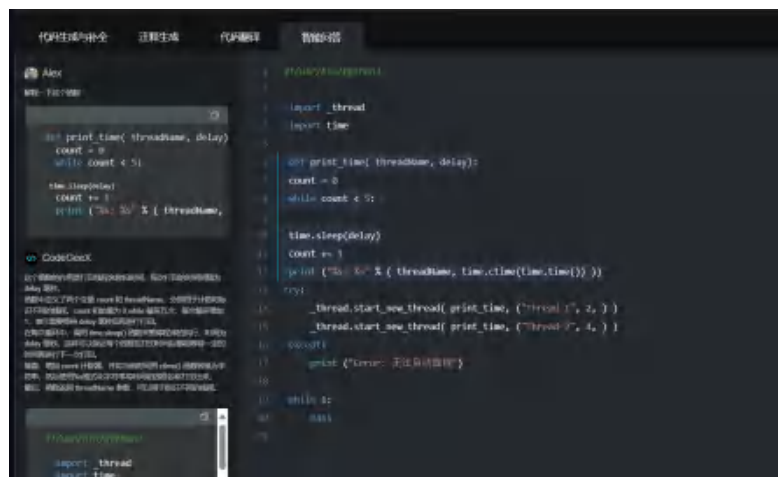


(2) 为下面实现零膨胀负二项分布的损失函数的 Python 代码生成中文注释

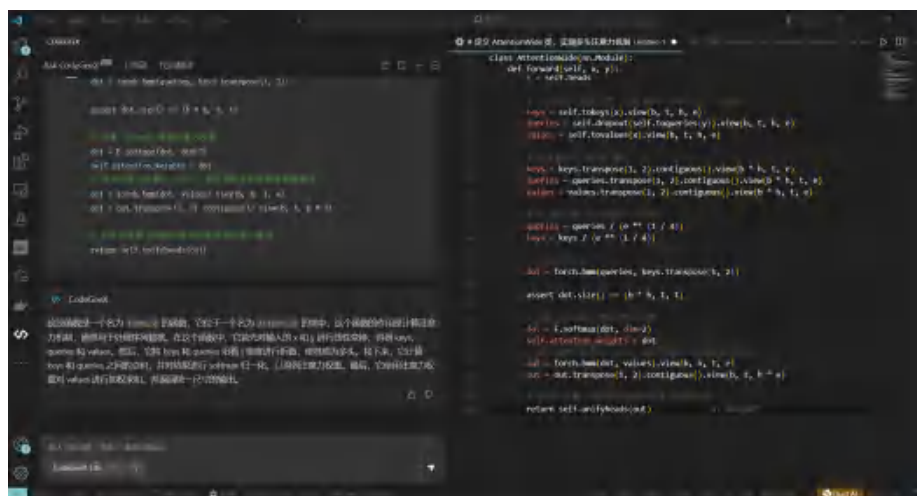




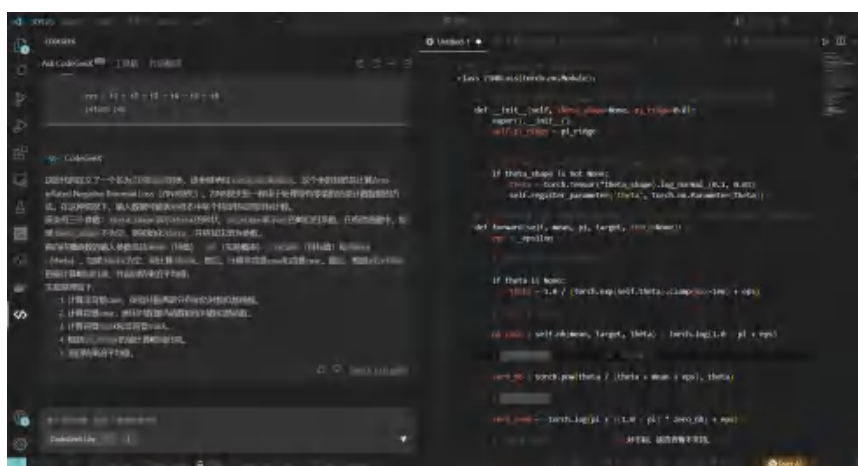
2.3.5 智能问答



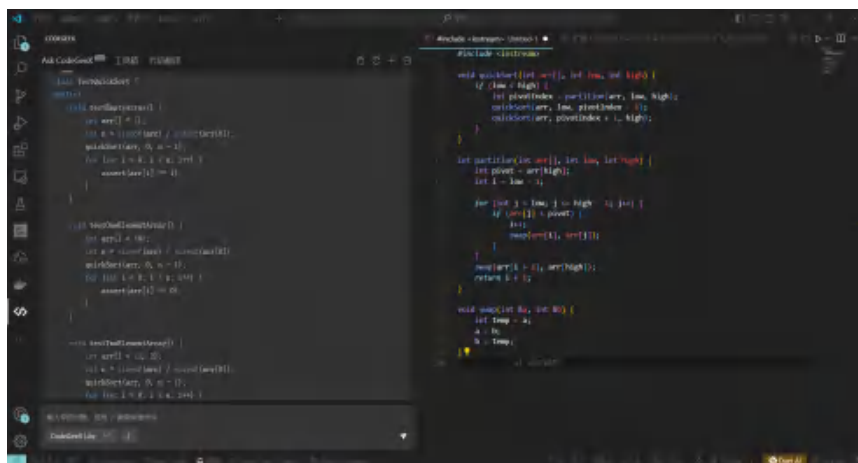
(1)



(2)



2.3.6 单元测试



2.4 目标用户与潜在用户

CodeGeeX 的目标是提高开发者的编码效率和代码质量，让开发者减少重复性繁琐的工作，更加专注于解决问题和创新，提高编程效率。因此，CodeGeeX 的目标用户应该是程序开发人员、算法工程师、研究人员等。

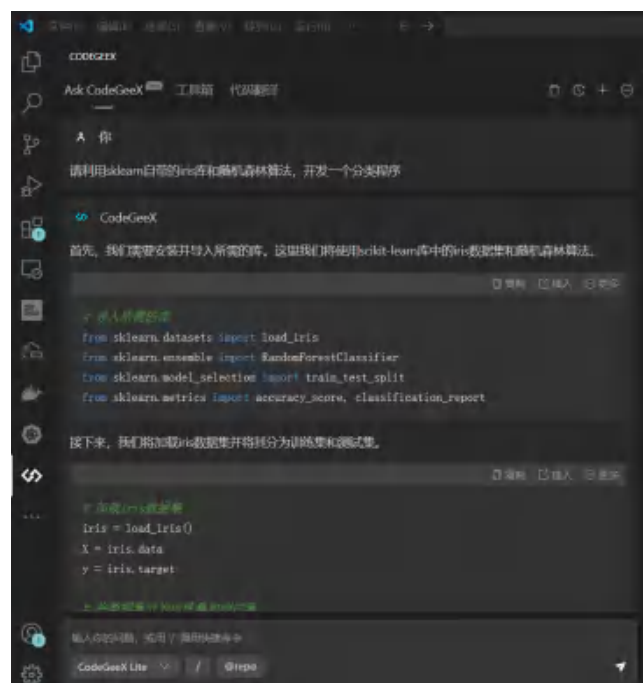
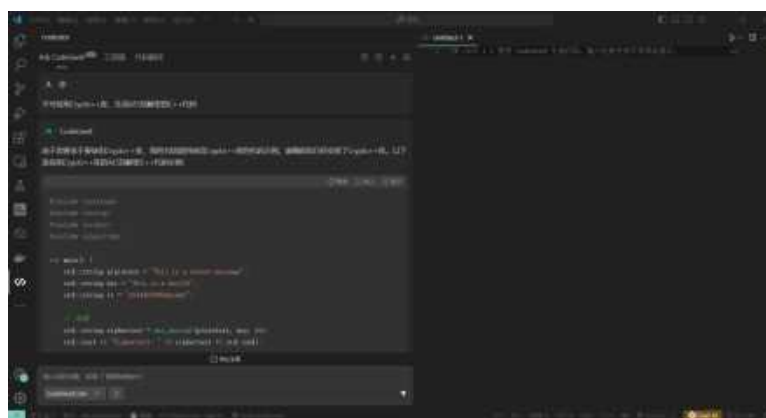
同时，CodeGeeX 也可用于编程初学者学生的学习中，便于老师的教学和学生对代码的理解，提升代码能力。

此外，CodeGeeX 也可用于编程能力不强但需要一个自动化程序以减少重复繁琐的工作，提升办公效率的工作人群中，可用 CodeGeeX 进行简单编程出程序框架，在此基础上进一步完善修改，以提升工作效率。

2.5 结合使用体验的一些优缺点

2.5.1 缺点及优化

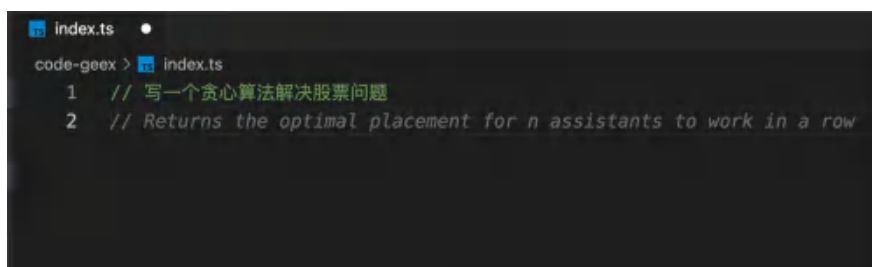
- 生成的代码有一部分过于简单，没有考虑到时间复杂度和空间复杂度等更深的问题，只是简单的实现了简单的需求。可能是因为 github 的开源代码，大多数都是很简单，写的很粗糙，训练集中精华代码不多，大模型也不容易学不出来好代码。代码的安全性也需要进一步考虑
- 代码生成质量不稳定，有时代码存在语法错误，也很难生成复杂需求的代码。如下面让他在不使用 Crypto++ 库的情况下生成 AES 加解密的代码，CodeGeeX 并不能很好的理解需求且生成的代码基本不可使用，需要人为重新修改。如让他利用 sklearn 自带的 iris 库和随机森林算法，开发一个分类程序，CodeGeeX 生成的代码报错 `AttributeError: 'numpy.ndarray' object has no attribute 'load_data'`，人为修正后才可成功运行。



- CodeGeeX 能使用不同语言解决编程问题，但各种语言擅长解答的问题分布却有较大差异，常用语言如 C++、C、Java、SQL、Python 等已可在准确率和速度较高的情况下解决大部分问题，但对需要使用 ts、golang 等不常用语言解决问题的场景需要进一步学习提升代码生成质量。CodeGeeX

已经能学习到各种语言正确的语法，却很难用不同语言解决同一个问题。可能是因为在训练集中出现了某种语言的类似实现，但缺少了其他语言的，当前模型的推理能力需要进一步提升。

- CodeGeeX 的上下文结合能力不强，编码能力较弱。如下面要它在 ts 文件中写一个贪心算法解决股票买卖问题时，它给的是 c++ 语言的头部引入。在重新让它生成代码时，它会补充注释，但是补充的是英文，没有结合上下文内容。



```
index.ts
code-geex > index.ts
1 // 写一个贪心算法解决股票问题
2 // Returns the optimal placement for n assistants to work in a row
```



```
index.ts
code-geex > index.ts
1 // 写一个贪心算法解决股票问题
2
3 #include <stdio.h>
```

- CodeGeeX 的少样本生成能力亟待进一步探索。对于大规模预训练模型，微调的成本是昂贵的，如何使用少量的样本就让模型生成想要的代码，对代码生成模型的实用化具有非常大的意义。chain-of-thought prompting 等技术或许也可以应用到 CodeGeeX 中。
- 模型大小和资源消耗较大，文档和社区支持也不够完善。
- 使用界面可以再进行优化（如添加必要的 copy、添加更多的快捷键等），便于用户交互及使用
- 不可避免的存在安全漏洞和潜在风险，无法理解安全性需求

2.5.2 优点及特色

- 支持多种主流 IDE，如 VS Code、IntelliJ IDEA、PyCharm、Vim 等，不同 IDE 环境下使用都比较流畅，且能够快速适配新的 IDE 版本。提供了调试功能，便于定位修复 bug。
- 支持 Python、C++、Java、JavaScript 和 Go 等多种主流编程语言的代码，能用不同的编程语言解决问题，且简单需求下常用的语言如 C++、Python、Java、SQL 等代码生成准确率也较高。
- 开源免费，所有代码和模型权重开源开放。
- 可以很好使识别光标所在位置是变量还是参数，便于编程人员理解代码。CodeGeeX 还能够实时分析代码，提供有关代码质量的反馈，帮助编写更加规范和可维护的代码。
- 可以快速查找所需的函数、类和方法，节省了大量的时间。

- CodeGeeX 允许根据个人喜好自定义设置，能够在享受便捷的同时保持自己的编码风格。使用简便，易于上手。
- 不需要梯子，没有任何网络限制，且响应较流畅。
- 包含代码补全、代码生成、代码翻译、代码注释、代码解释等常见工作场景，且仅需一个插件就可使用 CodeGeeX 的全部功能（如 Copilot 需要 5 6 个插件才可使用其全部功能），使用便捷
- 快捷键、图标设计、页面设计等较为简便，交互较为容易，提升了用户使用体验。

2.6 更新 CodeGeeX2

CodeGeeX2 是多语言代码生成模型 CodeGeeX (KDD' 23) 的第二代模型。不同于一代 CodeGeeX (完全在国产华为昇腾芯片平台训练)，CodeGeeX2 是基于 ChatGLM2 架构加入代码预训练实现，得益于 ChatGLM2 的更优性能，CodeGeeX2 在多项指标上取得性能提升 (+107 % > CodeGeeX; 仅 60 亿参数即超过 150 亿参数的 StarCoder-15B 近 10 %)，更多特性包括：

更强大的代码能力：基于 ChatGLM2-6B 基座语言模型，CodeGeeX2-6B 进一步经过了 600B 代码数据预训练，相比一代模型，在代码能力上全面提升，HumanEval-X 评测集的六种编程语言均大幅提升 (Python +57 %, C++ +71 %, Java +54 %, JavaScript +83 %, Go +56 %, Rust +321 %)，在 Python 上达到 35.9 % 的 Pass@1 一次通过率，超越规模更大的 StarCoder-15B。

更优秀的模型特性：继承 ChatGLM2-6B 模型特性，CodeGeeX2-6B 更好支持中英文输入，支持最大 8192 序列长度，推理速度较一代 CodeGeeX-13B 大幅提升，量化后仅需 6GB 显存即可运行，支持轻量级本地化部署。

更全面的 AI 编程助手：CodeGeeX 插件 (VS Code, JetBrains) 后端升级，支持超过 100 种编程语言，新增上下文补全、跨文件补全等实用功能。结合 Ask CodeGeeX 交互式 AI 编程助手，支持中英文对话解决各种编程问题，包括且不限于代码解释、代码翻译、代码纠错、文档生成等，帮助程序员更高效开发。

更开放的协议：CodeGeeX2-6B 权重对学术研究完全开放，填写登记表申请商业使用。

3 通义灵码



3.1 软件概述

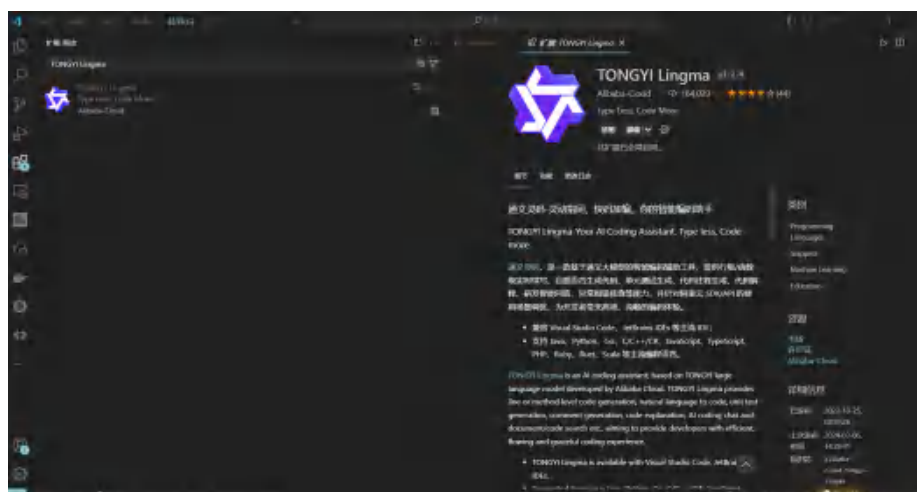
通义灵码，是阿里云出品的一款基于通义大模型的智能编码辅助工具，提供行级/函数级实时续写、自然语言生成代码、单元测试生成、代码优化、注释生成、代码解释、研发智能问答、异常报错排查等能力，并针对阿里云的云服务使用场景调优，助力开发者高效、流畅的编码。

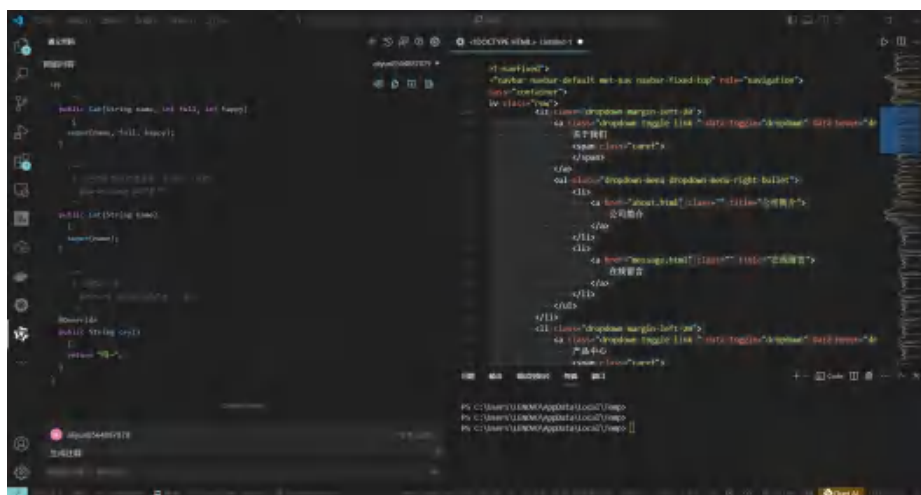
3.2 不同 IDE 的使用

通义灵码兼容 Visual Studio Code、JetBrains IDEs 等主流编程工具，支持 Java、Python、Go、JavaScript、TypeScript、C/C++、C# 等主流语言

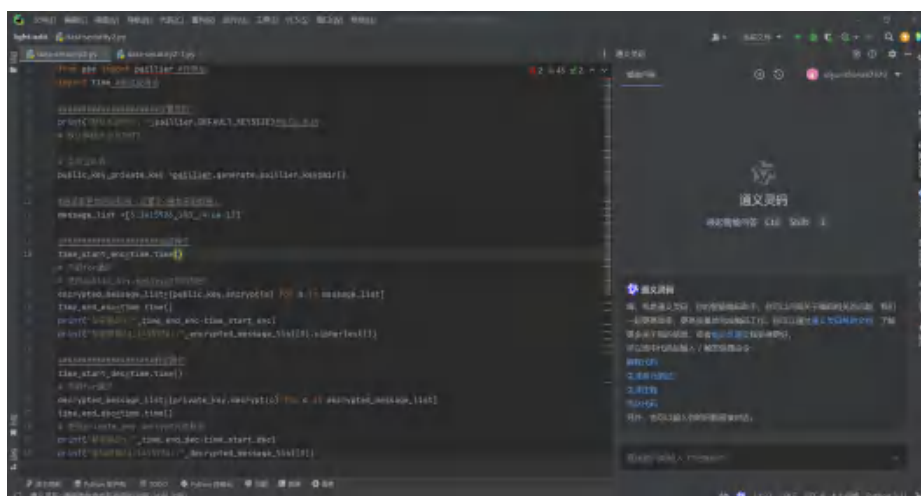
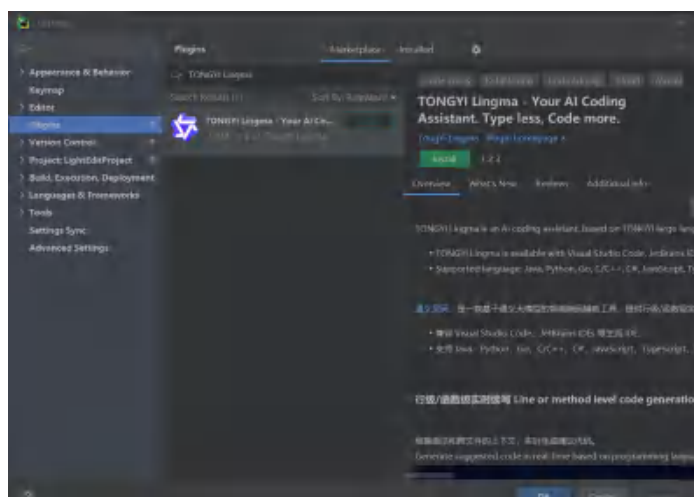


3.2.1 VSCode





3.2.2 JetBrains IDEs (以 Pycharm 为例)



3.3 软件功能

根据通义灵码网站首页介绍，通义灵码主要具有以下功能

3.3.1 代码生成与补全

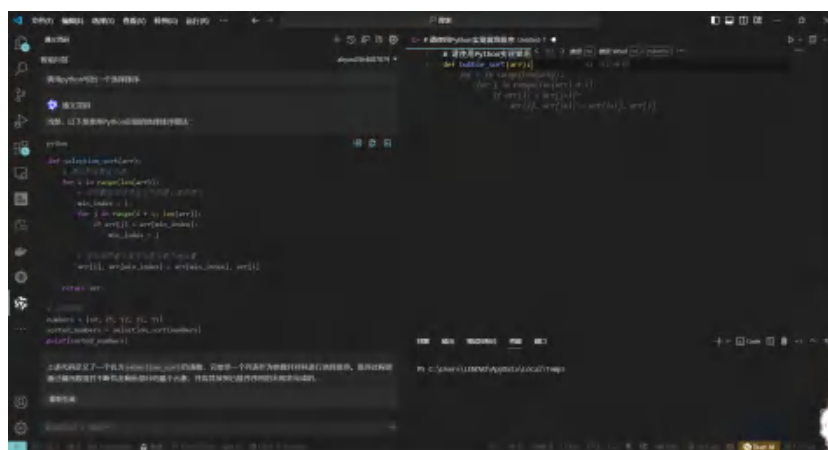
可根据上下文和当前语法自动预测和生成建议代码；还可根据自然语言描述生成想要的功能。



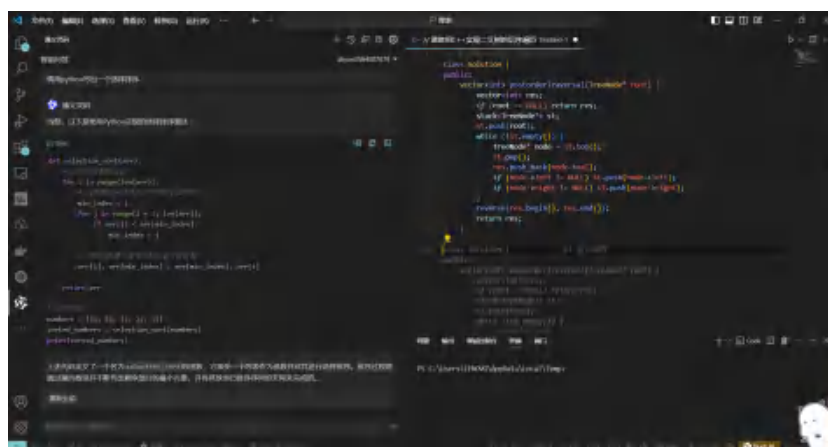
(1) 让其写出一个 python 的选择排序



(2) 让其实现 python 的冒泡排序

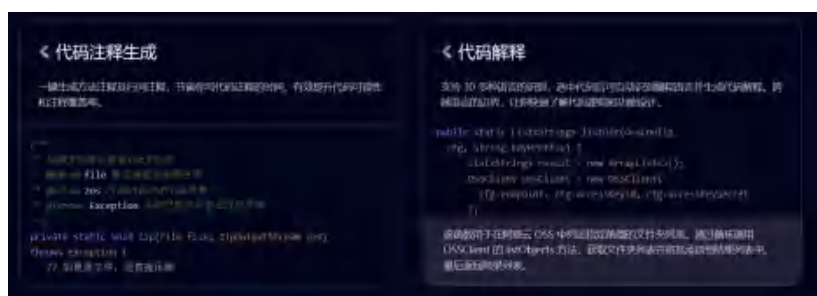


(3) 让其使用 C++ 实现二叉树的后序遍历

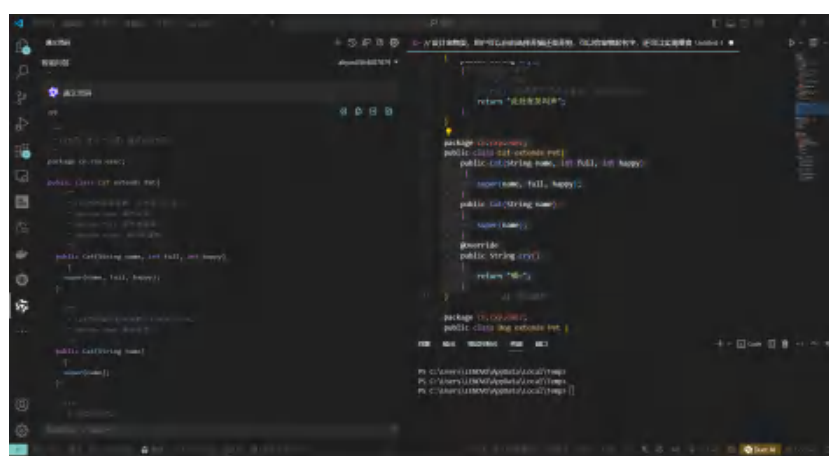


3.3.2 注释生成

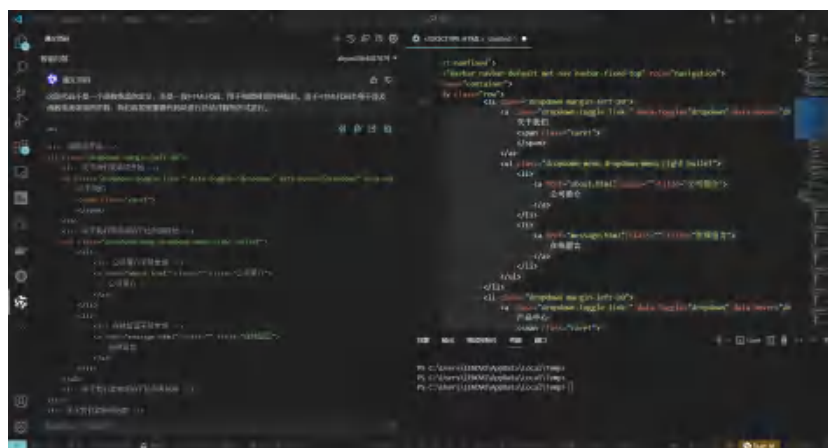
可一键生成方法及行间注释。覆盖各种编程语言，选中代码后可自动识别编程语言并生成代码解释。注释生成支持中英文，代码解释支持三种形式：中文精简、中文详细、英文。



(1) 让其对选中的 Java 代码生成注释

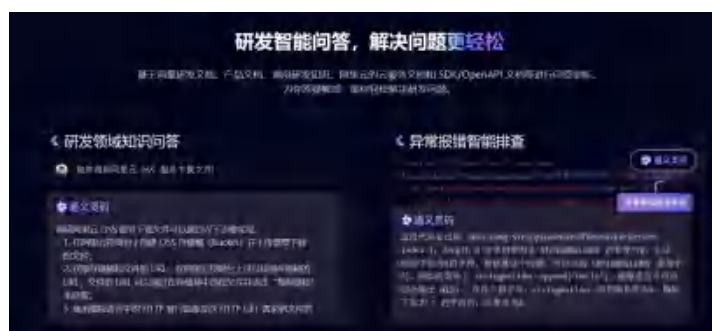


(2) 让其对选中的 html 生成注释

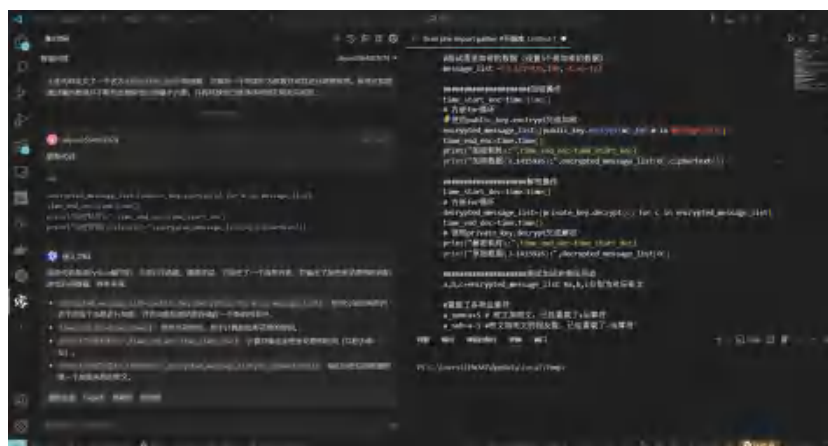


3.3.3 代码问答

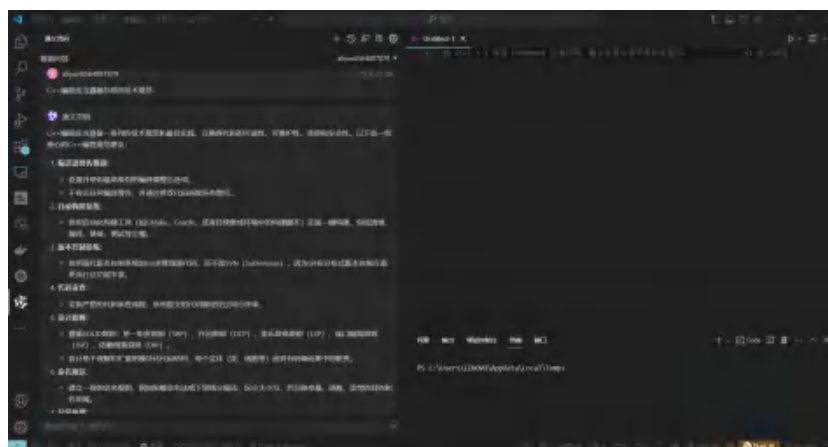
遇到编码疑问、技术难题时，一键唤起通义灵码，无需离开 IDE 客户端，即可快速获得答案和解决思路。聚焦研发领域知识的问答，帮助开发者快速解决问题。



(1) 让其对选中的 Python 代码进行代码解释



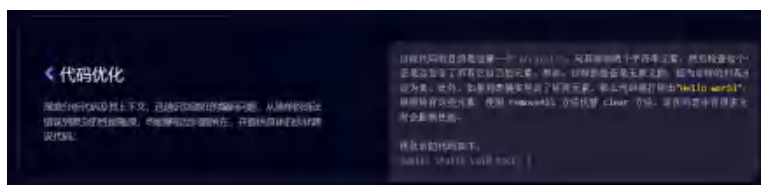
(2) 让其回答 C++ 编程应当遵循怎样的技术规范



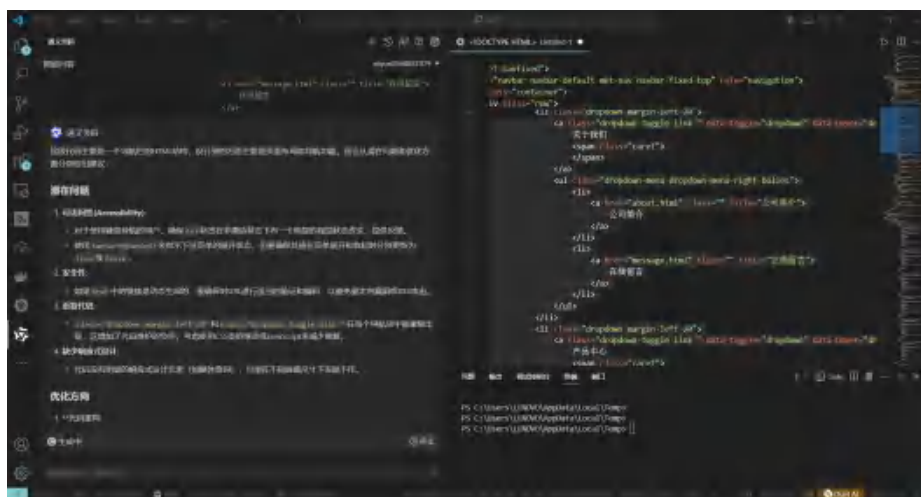
3.3.4 代码优化

还可结合上下文进行代码优化。深度分析代码及其上下文，迅速识别潜在的编码问题，从简单的语法错误到复杂的性能瓶颈，均能够指出问题所在，并提供具体的优化建议代码。

当运行出现异常报错时，一键启动报错排查的智能答疑，可结合运行代码、异常堆栈等报错上下文，快速给出排查思路或修复建议代码。



(1) 让其生成对选中代码片段的优化建议



(2) 生成对下面选中的 Java 片段的优化建议

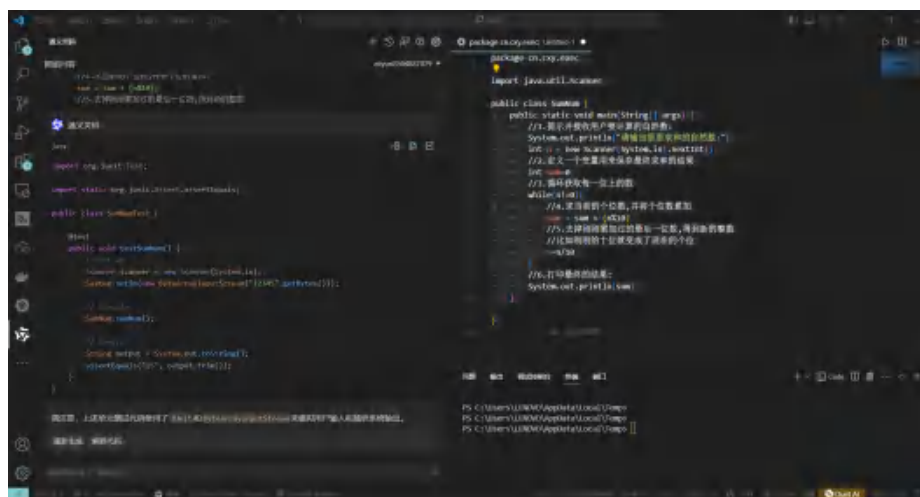


3.3.5 单元测试

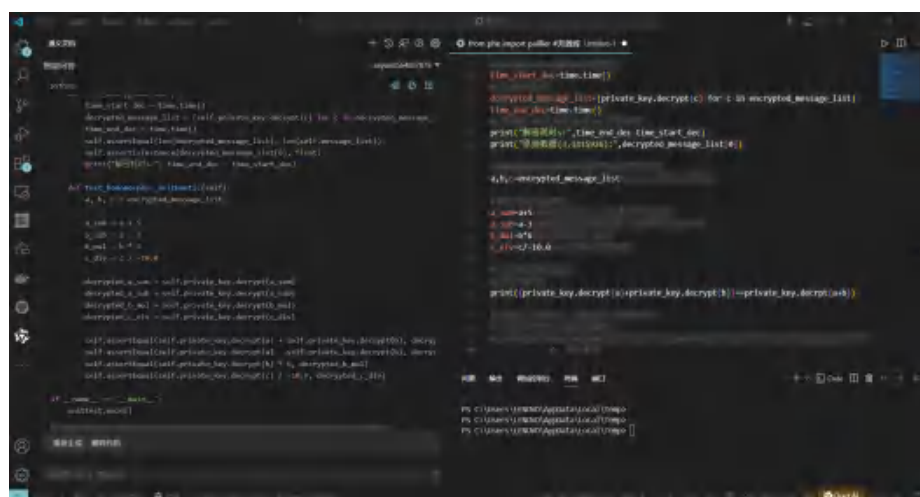
可自动生成多种框架的单元测试，支持根据 JUnit、Mockito、Spring Test、unit test、pytest 等框架生成单元测试以及单元测试相关代码解释。



(1) 让其生成对选中 Java 代码片段的单元测试



(2) 让其生成对选中 Python 代码片段的单元测试



3.4 目标用户与潜在用户

通义灵码的目标是提供行级/函数级实时续写、自然语言生成代码、单元测试生成、代码优化、注释生成、代码解释、研发智能问答、异常报错排查等能力，并针对阿里云的云服务使用场景调优，助力开发者高效、流畅的编码。因此，通义灵码的目标用户应该是程序开发人员（尤其阿里的开发人员）、算法工程师（尤其阿里）、研究人员（尤其阿里）等。

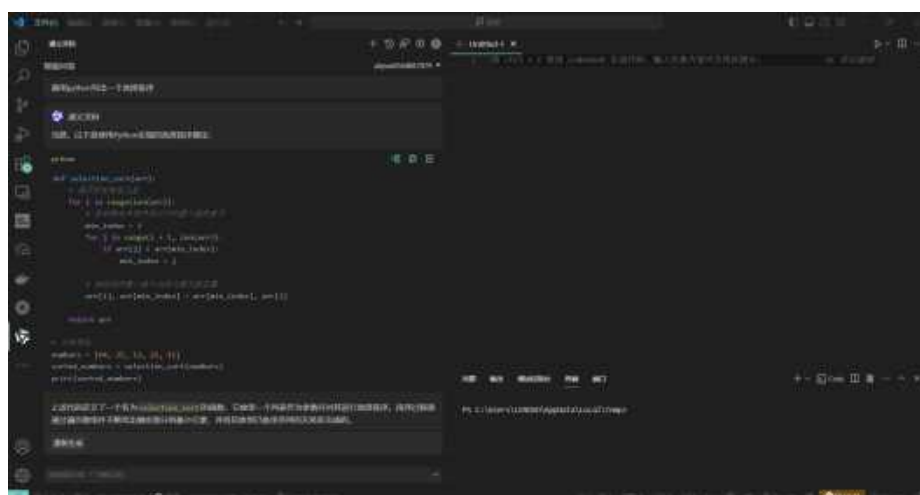
同时，通义灵码也可用于编程初学者学生的学习中，便于老师的教学和学生代码的理解，代码优化、代码解释、异常报错排场等功能对于初学者的学习有很大的帮助。

此外，通义灵码也可用于编程能力不强但需要一个自动化程序以减少重复繁琐的工作，比如可以编程快速完成 Excel 数据处理、文本读取处理等日常工作中可能会频繁需要的功能，以提升办公效率的非程序员的工作人群中，可用通义灵码进行简单编程出程序框架，在此基础上进一步完善修改，同时与百度、谷歌等搜索引擎配合使用大大提高日常效率

3.5 结合使用体验的一些优缺点

3.5.1 优点及特色

- 快捷键、图标设计（如问答界面的复制、插入键等）、页面设计等较为简便，用户交互较为容易，提升了用户使用体验。



- 国产化且免费，降低了用户使用门槛
- 注释生成支持中英文，代码解释支持三种形式：中文精简、中文详细、英文，便于中文用户的使用。
- 智能问答中对于代码编写、技术咨询类的问题，都能够比较快速的响应，减少了开发人员对 Google 和 Bing 的查询次数。可以回答开发相关的问题，提供调试和排查建议，支持阿里云的资源 SDK/OpenAPI 和帮助文档，经常包含相关来源的链接，准确性较高，帮助性较大。大多数情况下，产品生成的内容智能程度较高，能够根据用户的输入生成合适的代码、问题回答、测试等。



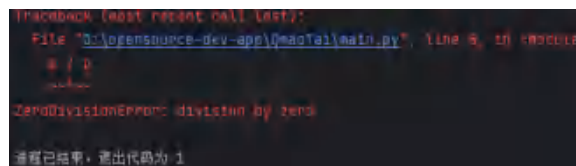
- 不管是注释生成、代码解释还是智能问答可以生成适当长度的内容，不过短也不过长，便于用户判断。
- 相对代码生成而言，代码注释和单元测试算比较好用的功能。使用通义灵码的代码解释可以更快的了解项目（虽然有时注释和解释略显粗糙）

3.5.2 缺点及优化

- 使用界面可以再进行优化,如避免通义灵码的快捷键与常用快捷键(eg: 通义灵码中 Ctrl+shift+L 用于唤起智能问答与 VSCode 中快捷方式冲突,有时通义灵码的快捷键与其他插件的快捷方式也存在冲突)冲突、缺少删除问答记录等,以进一步便于用户交互及使用
- 通义灵码的续写,采用了 Api 注解源码的默认使用写法,现在版本的代码生成对用户需求的理解不深(比如不太理解用户的提示性注释),不能很好的匹配用户的实际需求,可以进一步优化,有很大的优化空间
- 如果可以实现现在单元测试的时候直接在对对应目录创建文件,以及实现相关代码,就进一步方便用户的使用了
- 在 Java、Python 代码等的编写中,缺少对一些库和包的导入,可进一步优化
- 使用不同语言解决编程问题,但各种语言擅长解答的问题分布却有较大差异,常用语言如 C++、C、Java、Python 等(其中 Java 代码生成在常用的语言中质量相对较好)已可在准确率和速度较高的情况下解决问题,但对需要使用 ts、golang 等不常用语言解决问题的场景需要进一步学习提升代码生成质量。
- 代码生成质量不高,阿里相关的 API 还能略微好一点。有时代码存在语法错误,或不符合技术规范等,也很难生成复杂需求的代码。



```
Exception in thread "main" java.lang.ArithmeticException: / by zero
at com.haier.market.sp.console.TestMainTest.java:12
进程已结束，退出代码为 1
```

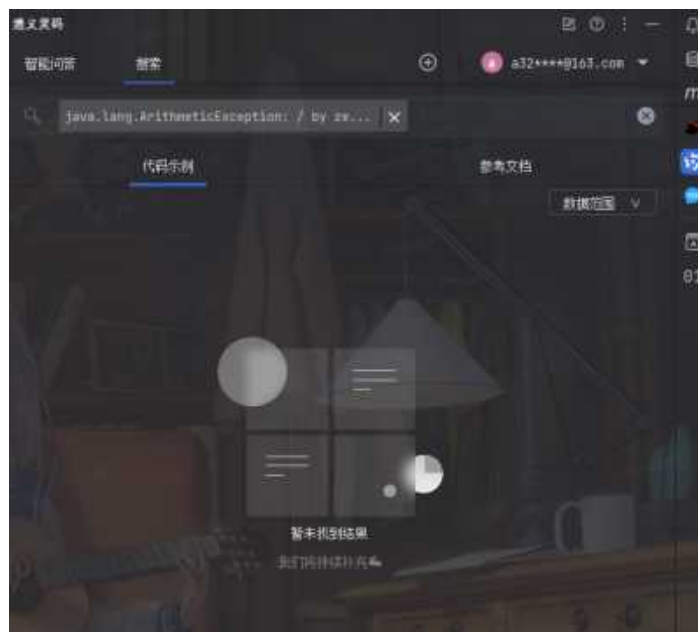


```
Traceback (most recent call last):
  File "D:\logensource-dev-app\qiaoTeli\main.py", line 9, in <module>
    a / 0
ZeroDivisionError: division by zero
进程已结束，退出代码为 1
```

- 有时智能问答生成回答的智能程度较低,只是提供一些常见或模板化的内容,有时甚至答非所问(如下),对用户的用处不大。

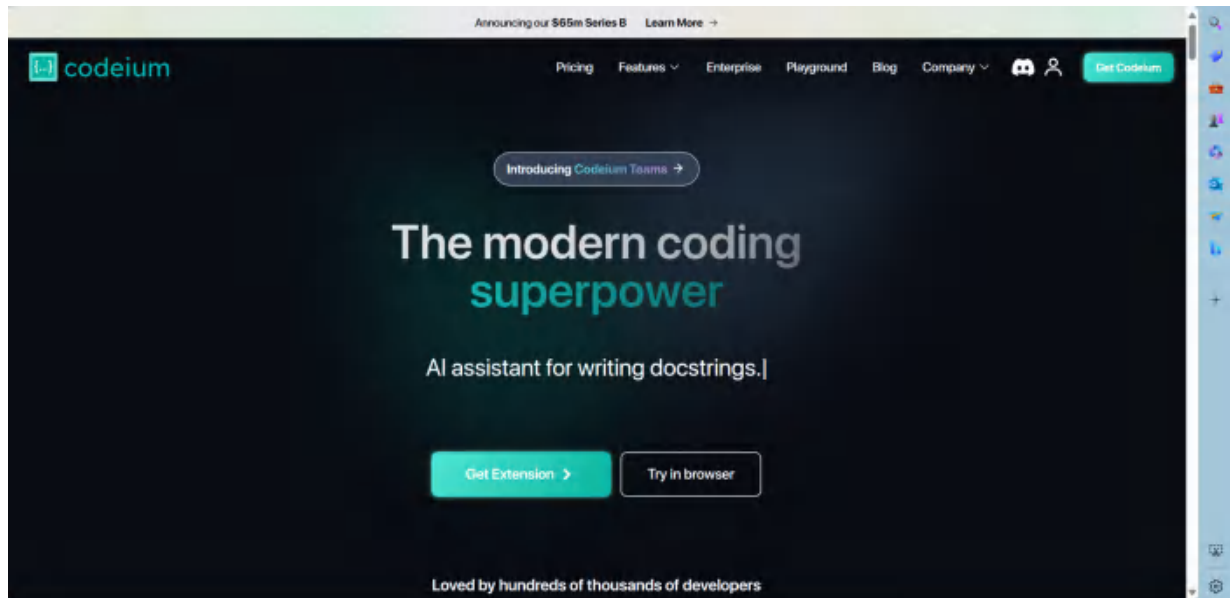


- 单元测试不同语言的质量不同，与 Java 相比，python 单元测试的质量需要进一步提升。Java 单元测试只引入的相关服务类，并没有引入相关的包，目前更多的是 SpringBoot 框架，希望未来可以添加 SpringBootTest 的相关实现
- 生成的代码注释有时会出现与用户的输入不完全匹配略微答非所问以及细节错误注释的情况。代码解释有时又过于简单过于粗糙，可读性不好，不方便编程新手对代码的理解和使用。
- 对于异常检查 (Java)，经常显示未找到异常结果。目前更多的是针对阿里相关的 SDK、文档等，对于这些范围内的异常可以检查出来，但对于范围外以及一些比较复杂的异常很难检查出来，是一个比较尴尬的功能。



- 上下文感知能力和上下文信息的补充不是很好，忽略了一些限制和约束，存在不足，可进一步优化。
- 不可避免的存在安全漏洞和潜在风险，不能很好的理解安全性需求

4 Codeium



4.1 软件概述

Codeium 是一款免费的智能编程助手，类似 Github Copilot，目前 Codeium 提供 70+ 种语言的代码完成工具，具有闪电般的速度和最先进的建议质量。支持 17+ 几乎所有主流编程语言和 IDE，个人用户免费使用。

轻松集成至所有 JetBrains IDE 产品中，包括我们最常用的 Pycharm、IDEA、Android Studio 等，也可以集成至 Vim 编辑器中。

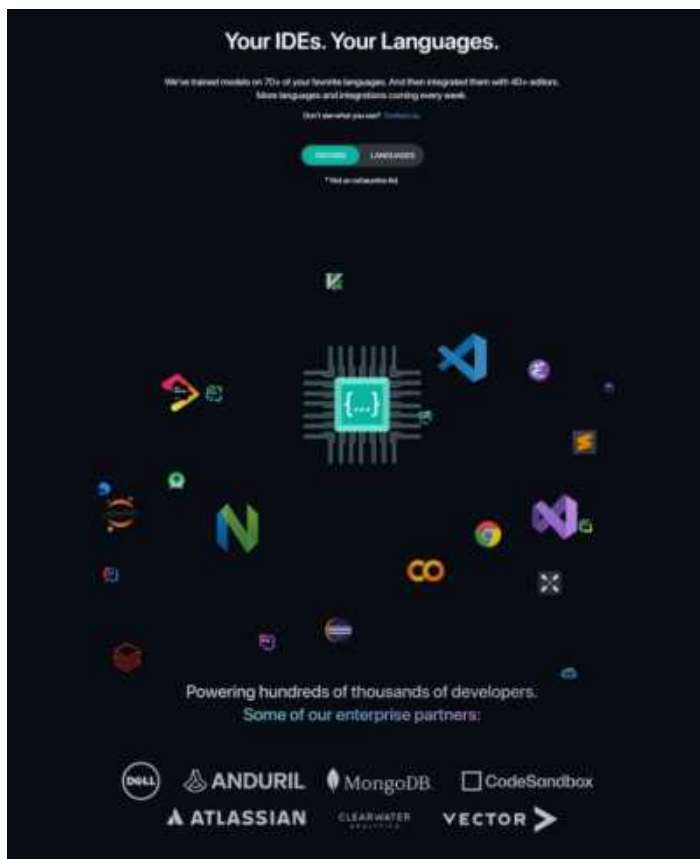
使用 Codeium, 可以永远无限制地完成单行和多行代码, 支持 40 多种编程语言: Javascript、Python、Typescript、PHP、Go、Java、C、C++、Rust、Ruby 等。

Codeium 由 Exafunction 的团队开发，该团队核心成员主要由华人组成，Exafunction 通过优化深度学习推理工作负载，在资源利用率和成本改进方面可以提供 10 倍的性能和效益，专注于构建深度学习应用程序，而不是管理集群和微调性能。

Codeium 背后由美国两家顶级风险投资机构 Greenoaks 和 Founders Fund 投资，主要聚焦在科技和新兴市场领域进行投资，其也投资了马斯克的 SpaceX。

4.2 不同 IDE 的使用

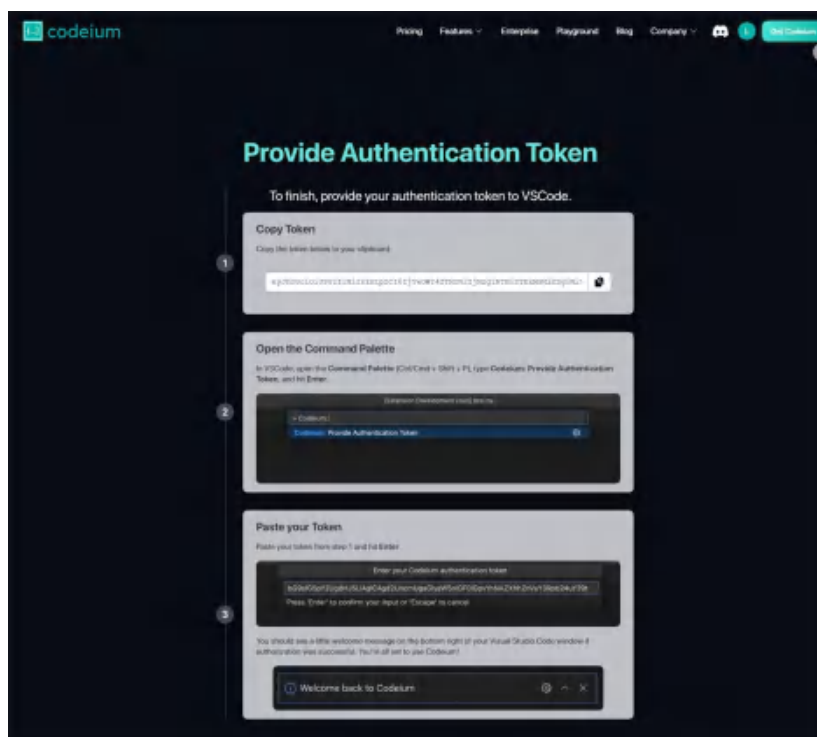
Codeium 官网介绍说已对 70 多种最喜爱的语言进行了模型训练。然后将它们与 40 多种编辑器集成。每周都会有更多语言和集成。

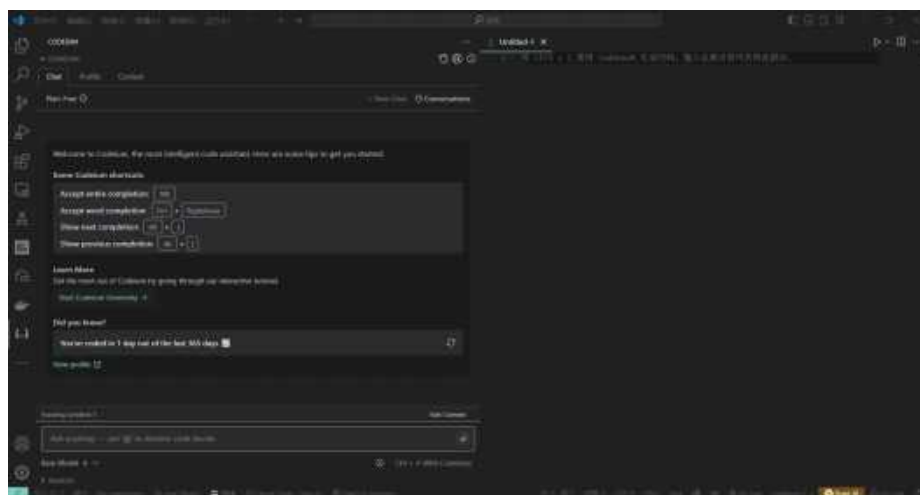


4.2.1 VSCode

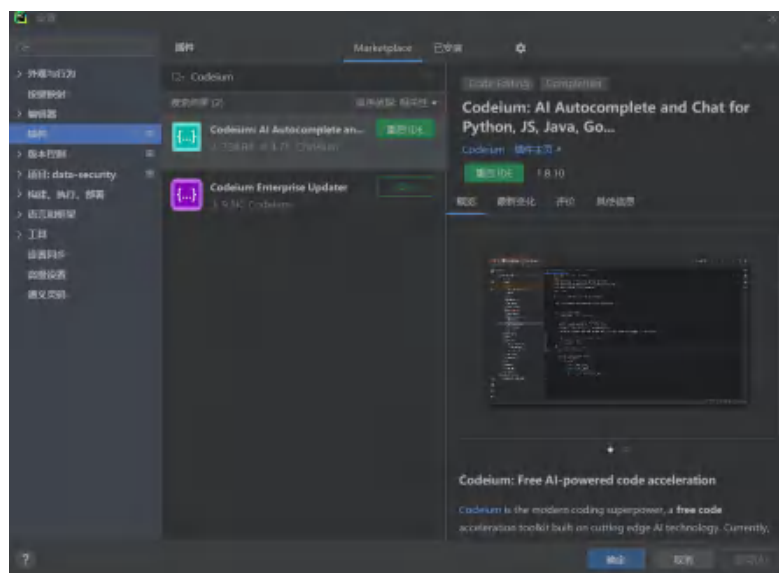


注册之后，会弹出一个 Token 页面，并提供一组密钥，然后回到 VS Code，按下 `Ctrl+Shift+P`，输入 `Codeium:Provi...`，并将复制的密钥填入。当 VS Code 下面欢迎字样时，就说明可以正常使用了。点开左侧的... 图标，也会看到 CHAT 和 SEARCH 中出现了内容，就可以跟他对话了。具体使用时，并不需要科学上网。

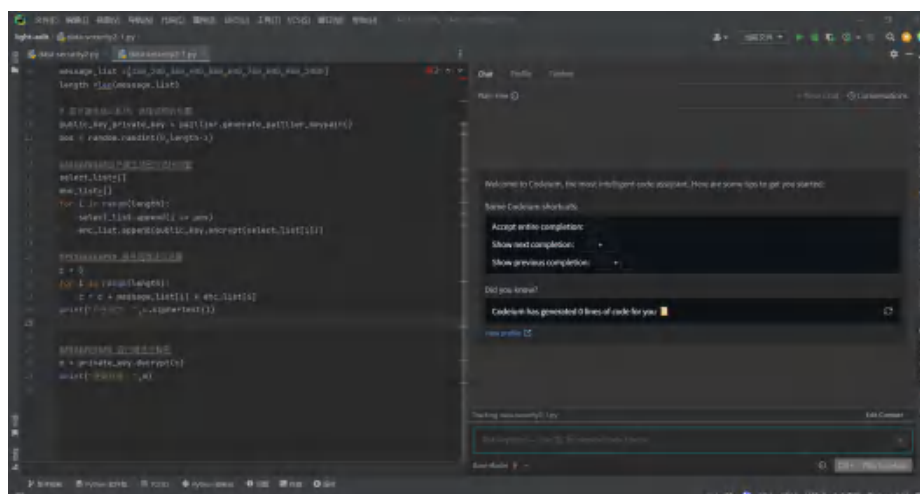




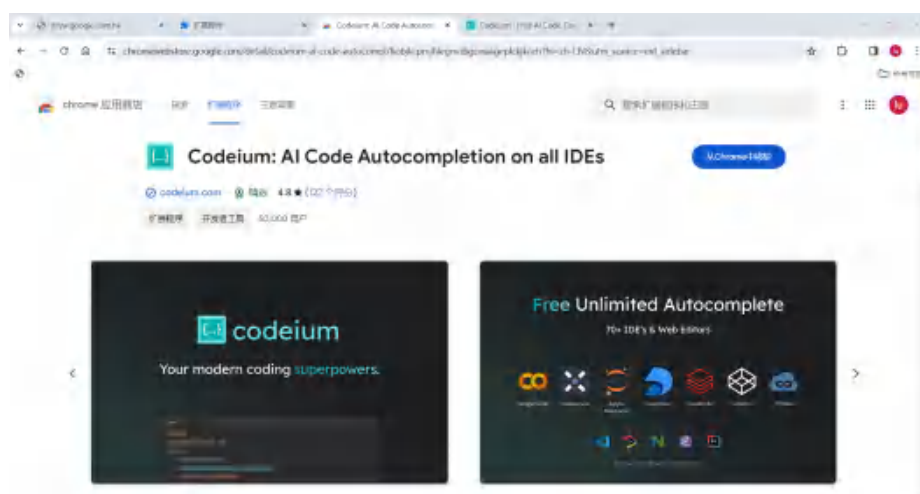
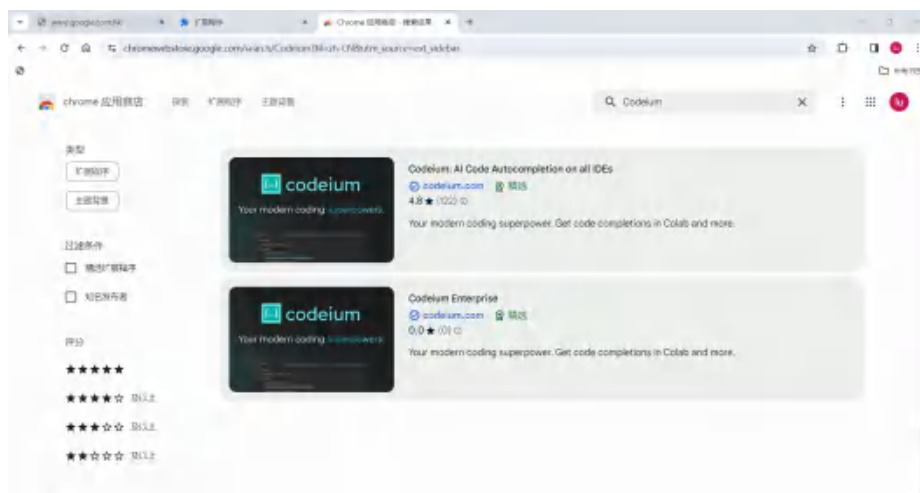
4.2.2 JetBrains IDEs (以 Pycharm 为例)



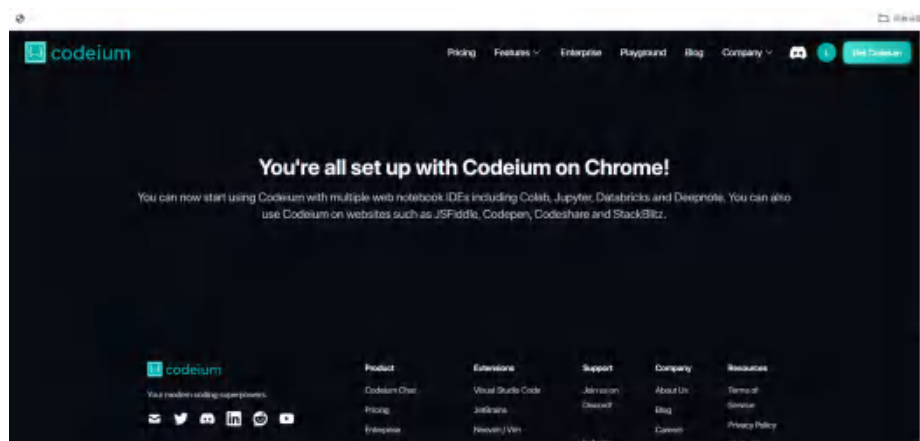
与 VSCode 中的使用一样，也是需要先在官网注册登录后才能使用

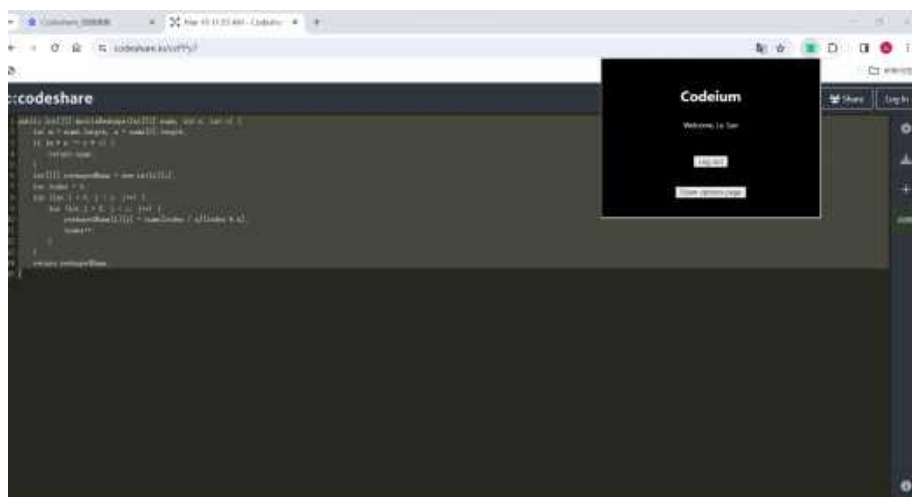


4.2.3 Chrome 插件



根据提示应该是只能在特定的几个网站使用，如 JSFiddle, Codepen, Codeshare, StackBlitz 等。

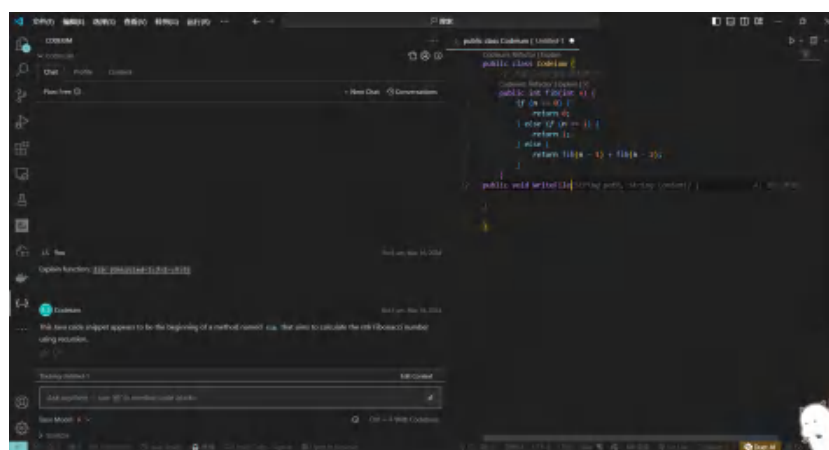




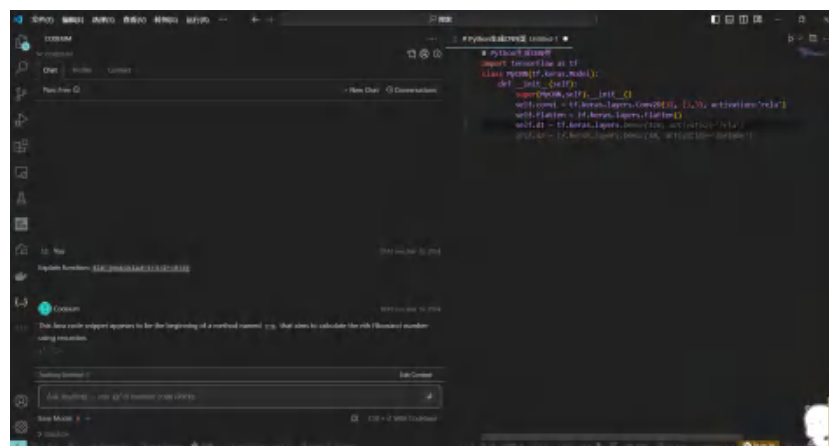
4.3 软件功能

4.3.1 代码生成与补全

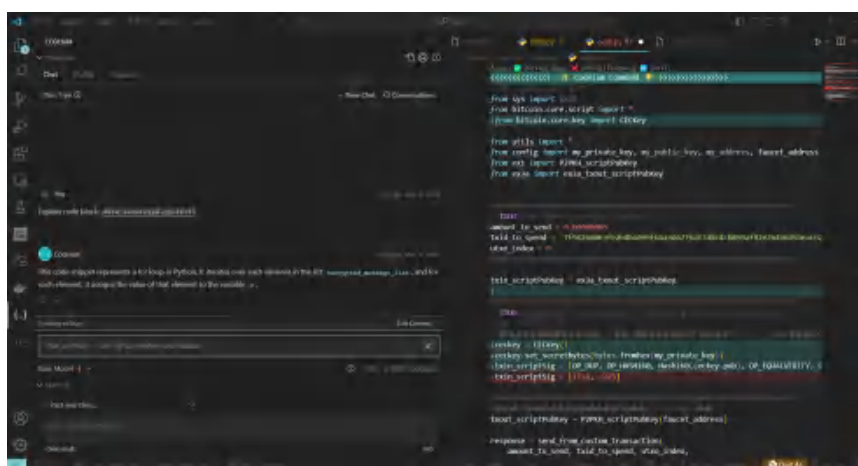
(1) 自动生成 Java 代码函数的参数



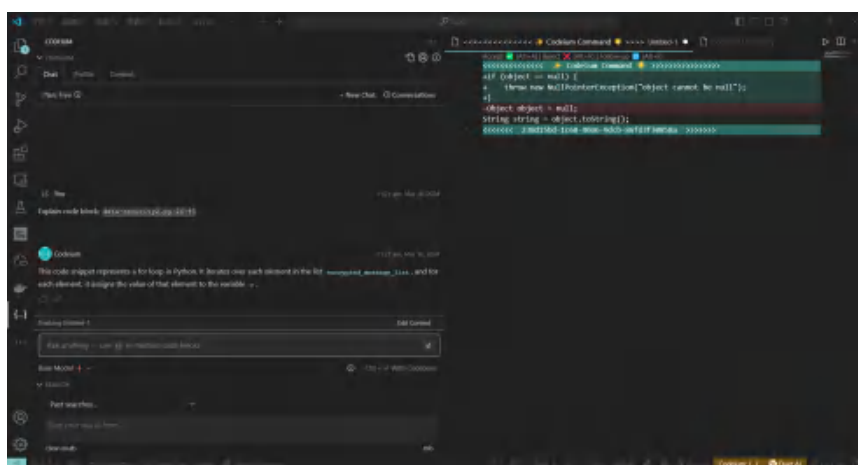
(2) 使用 Python 编写 CNN 类的代码



(3) 完成 TODO 的补全

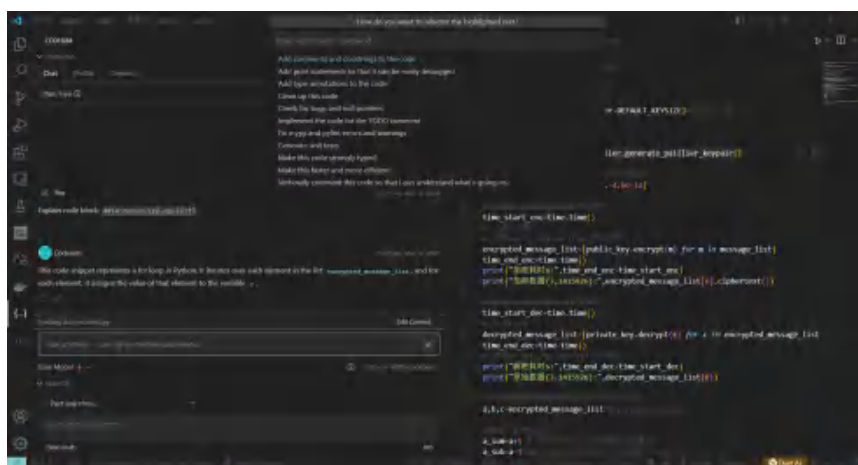


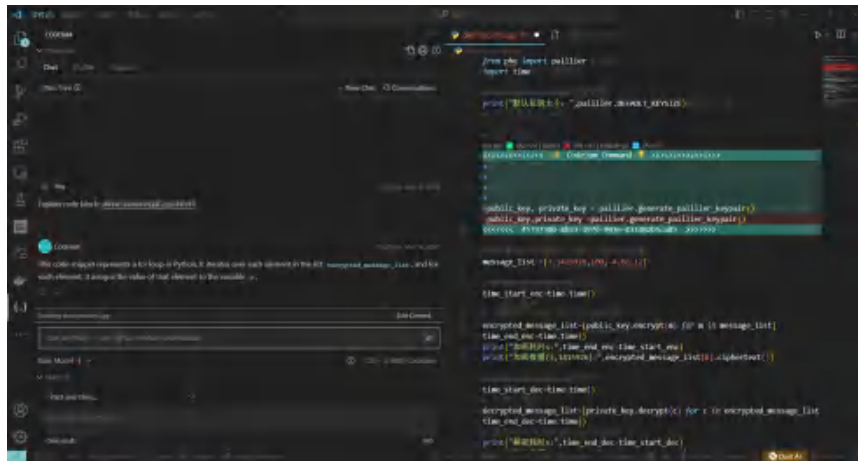
(4) 在调试代码时生成对空指针或 bug 的检查代码



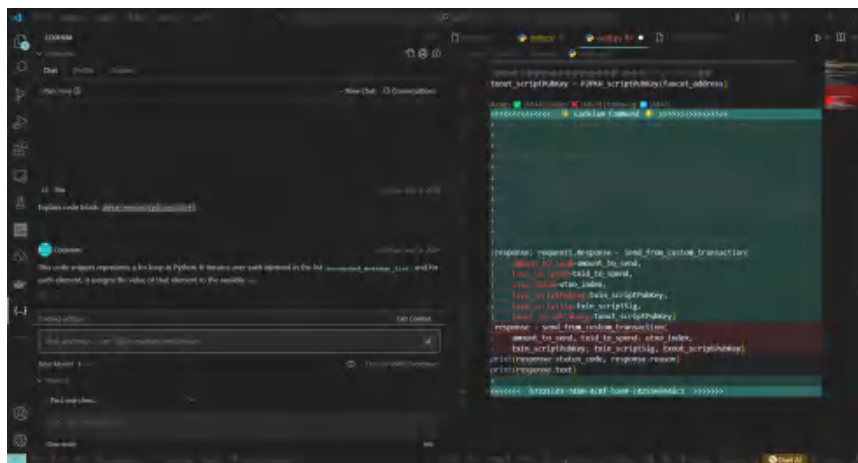
4.3.2 注释生成

(1) 对选中的 Python 代码生成注释



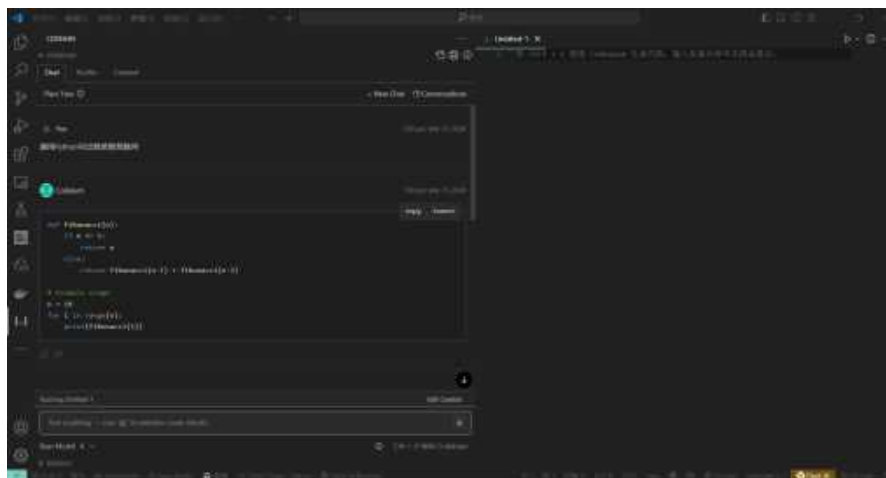


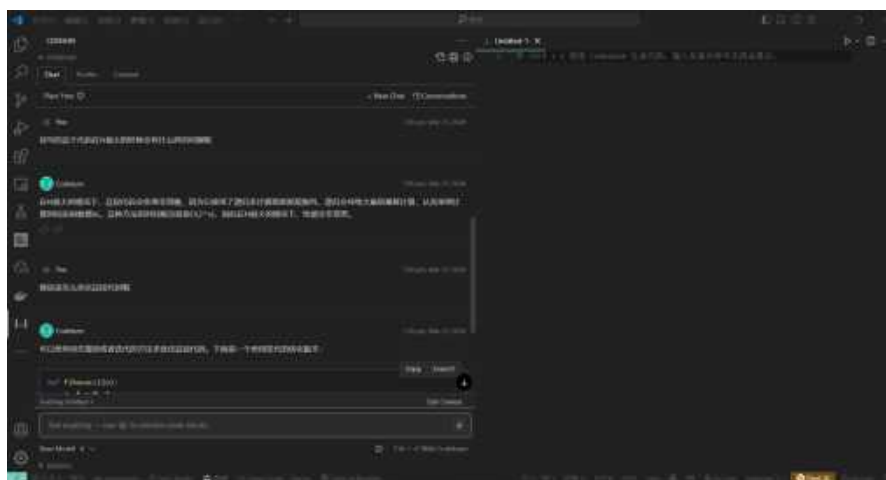
(2) 对选中的 Python 代码生成代码解释



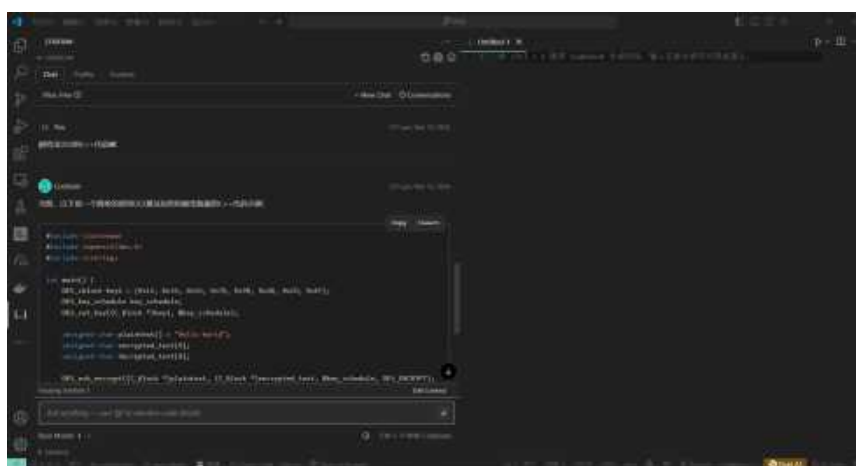
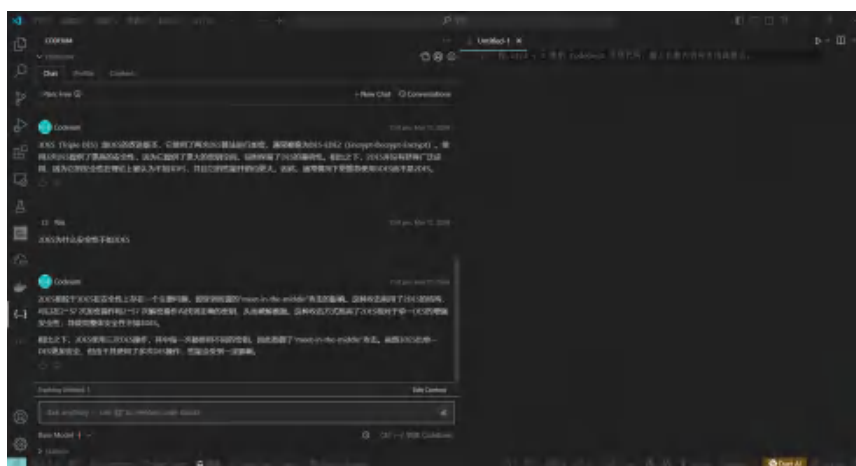
4.3.3 智能问答

(1) 用 Python 写出斐波那契数列的代码，并问答其优化相关的问题



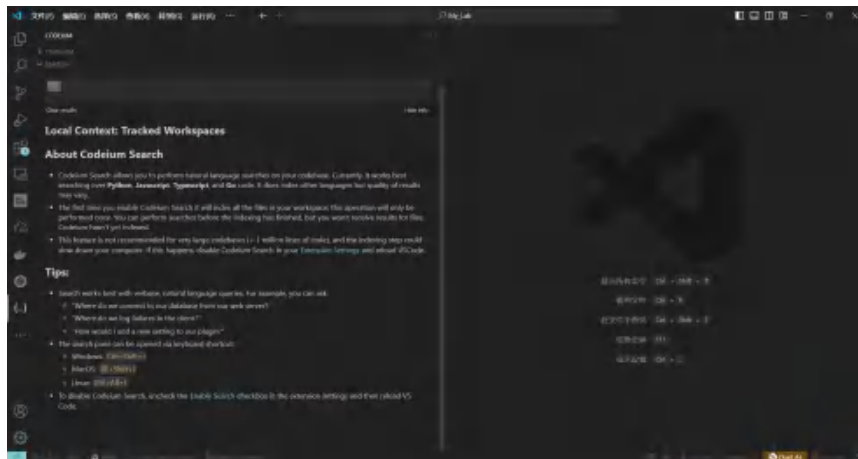


(2) 多次连续问答有关 DES 算法的问题，如为什么 2DES 的安全性不如 3DES，C++ 代码框架生成等



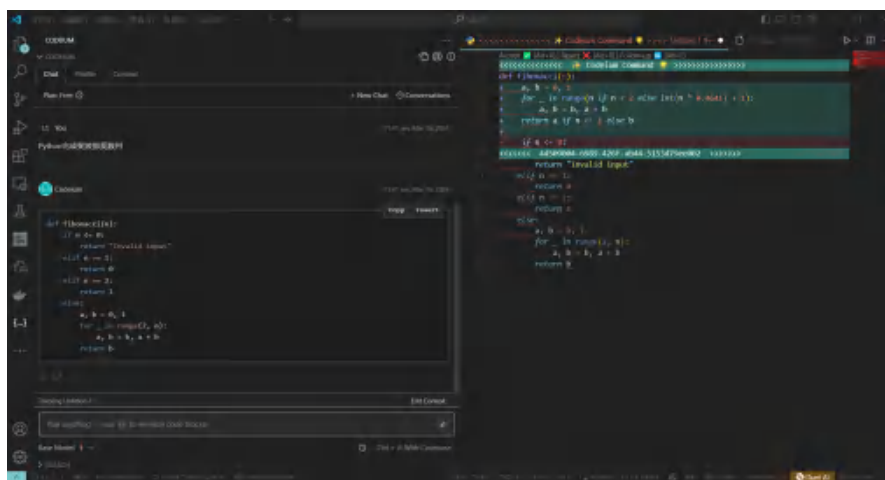
4.3.4 代码搜索

只有 VSCode 中的 Codeium 才有的功能, 其他 IDE 插件中暂时没有该功能。可以提供技术搜索。

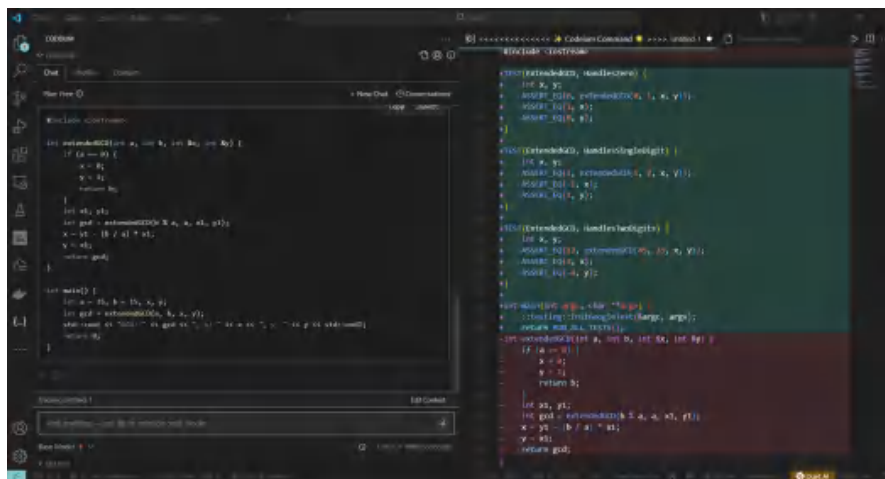


4.3.5 代码优化

(1) 对 Python 斐波那契代码进行优化

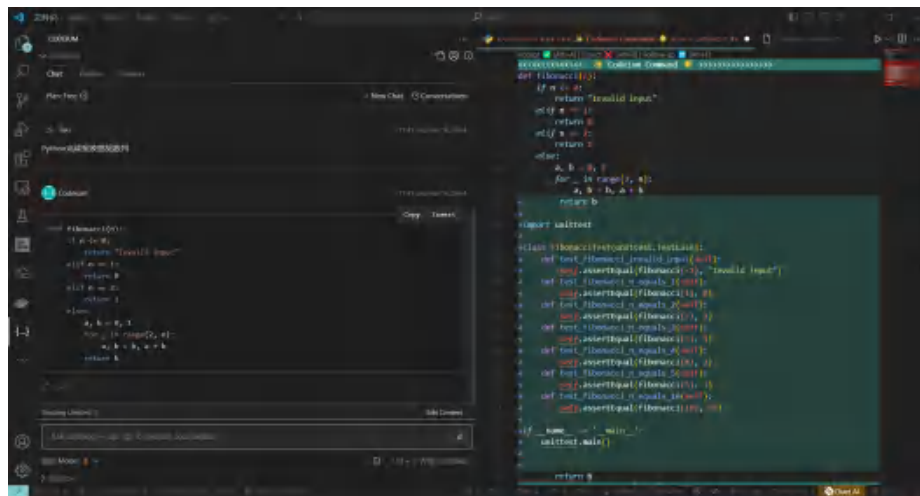


(2) 对下面求字符串分割问题的 Java 代码进行优化

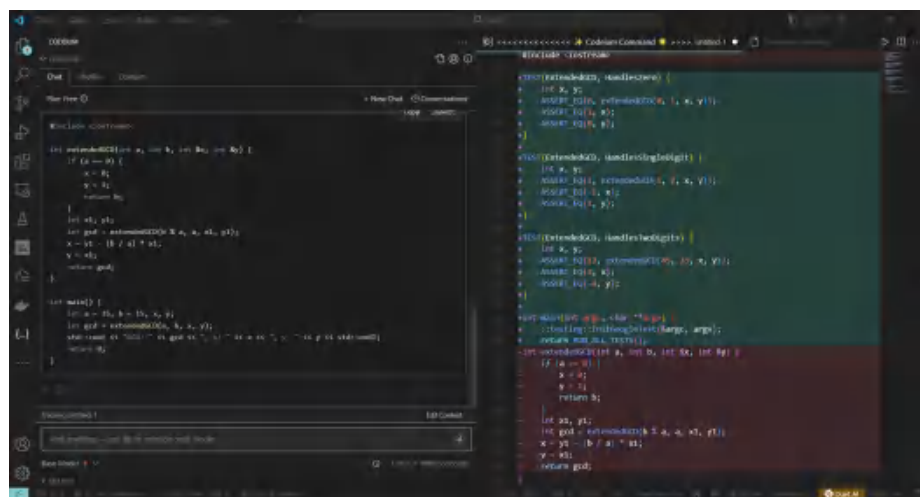


4.3.6 单元测试

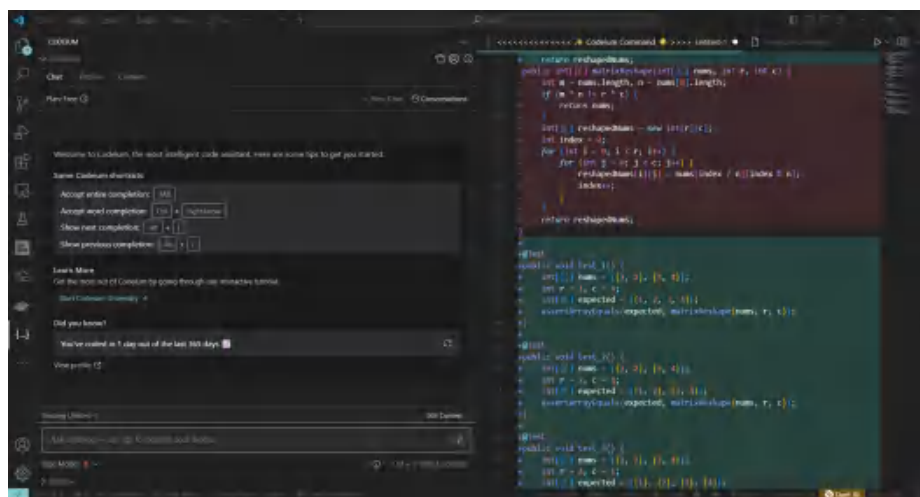
(1) 对下列 Python 代码生成单元测试



(2) 对选中的 C++ 代码生成单元测试



(3) 对选中的 Java 代码生成单元测试



4.4 目标用户与潜在用户

Codeium 是面向个人用户及团队的 AI 编程工具。因此, Codeium 的目标用户应该是程序开发人员、算法工程师、研究人员等。Codeium 商业版的一些功能如代码重构 (Codeium 可以自动重构代码, 帮助程序员更好地管理和维护代码)、代码调试 (Codeium 内置了调试器, 能够帮助程序员快速发现和解决代码中的问题)、版本控制 (Codeium 支持多种版本控制系统, 包括 Git、SVN 等, 方便程序员管理和维护代码)、团队协作 (Codeium 支持多人协作开发, 能够提高团队的工作效率)、代码托管、微调等功能可以协助工程师更好的完成团队的项目, 提升团队的工作效率

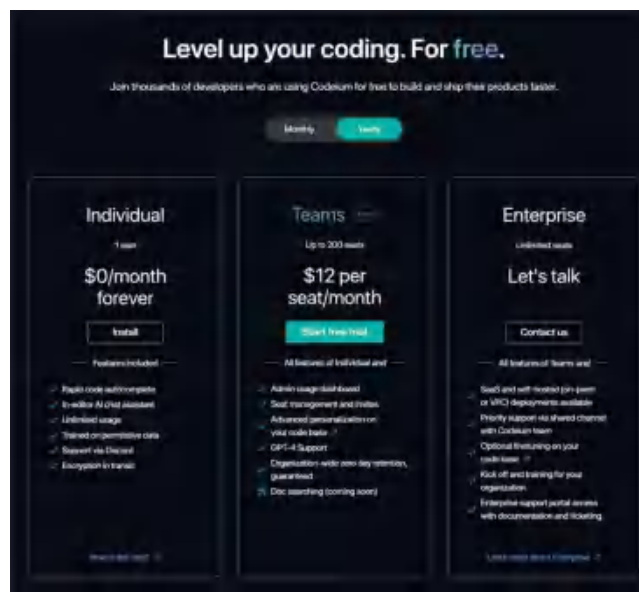
同时, Codeium 也可用于编程初学者学生的学习中, 便于老师的教学和学生对代码的理解, 代码优化、代码解释、异常报错排查等功能对于初学者的学习有很大的帮助。

此外, Codeium 也可用于编程能力不强但需要一个自动化程序以减少重复繁琐的工作, 比如可以编程快速完成 Excel 数据处理、文本读取处理等日常工作中可能会频繁需要的功能, 以提升办公效率的非程序员的工作人群中, 可用 Codeium 进行简单编程出程序框架, 在此基础上进一步完善修改, 同时与百度、谷歌等搜索引擎配合使用大大提高日常效率

4.5 结合使用体验的一些优缺点

4.5.1 优点及特色

- 对个人用户免费, 主要靠商业版 (企业版) 挣钱



- 安全性有一定的考量。Codeium 声称非常重视安全和隐私, 不出售/共享用户数据/代码或在用户代码上进行生成模型的训练 (甚至使完全退出任何代码片段遥测变得超级容易) 实际使用的是如如果用户在推送模型或扩展更改后接受更多建议之类的信号, 以此迭代 (即聚合的 A/B 测试)
- Codeium 并不是使用的目前非常流行的 Open AI 公司的 GPT 模型进行训练的, 应该是一个自己训练的结合了深度学习和自然语言处理技术的模型, 响应速度较快
- Codeium 可以在 Chrome 浏览器中使用, 目前其他智能编程工具还不能在浏览器中使用
- 支持中英文问答, 方便中文用户的使用

- 交互界面简单干净，copy,insert 等键可以很方便的进行代码复制与代码测试

4.5.2 缺点及优化

- Codeium 虽然对个人用户免费，但 Codeium 有很多功能只在团队或组织层面才有意义，例如代码自主托管或对代码库进行训练微调（但个人用户无法使用类似功能，本次报告也没有办法进行使用调研）。自主托管需要预先支付硬件成本，这对于个人开发者来说是无法承受的，而微调只有在存储库足够大，并具有足够的应用程序/产品特定逻辑或惯例时才有意义。
- Codeium Search、Codeium Chat 等部分功能目前只在 VSCode 中能使用，其他 IDE 中是暂时没有这部分功能的
- 注册的时候需要科学上网，不能在国内的网络环境下注册
- 登录时有时找不到自动登录的窗口，有时手误退出后很难再找到 Codeium 的弹窗，不方便用户的使用
- 部分快捷键与 VSCode 中快捷键以及其他插件的快捷键冲突
- Codeium 部分功能如代码注释、代码搜索只能英文提问，并生成英文回答，不兼容中文，不方便中文用户的使用
- Codeium 的性能和准确性受到训练数据的影响，如果训练数据质量不高，生成的代码可能不够准确或合理。
- Codeium 是基于 AI 技术生成代码，存在一定的安全风险，可能会生成不安全或有漏洞的代码。
- 生成的代码质量不高，会有各种语法错误需要人为修改
- 智能问答有时答非所问，无法理解用户需求。上下文结合能力不强，需要进一步改进

5 未来

如果你是新上任的项目经理，对于智能编程助手如何提升竞争力从而在竞争中胜出？以产品功能为例，还需要设计什么样的功能？为什么要做这个功能而不是其他功能？

如果你来开发一款新的编程工具，你会以什么形式开发这款产品？你会怎么设计这款产品？

5.1 如何提升竞争力

- 进一步提升上下文结合能力，语言模型的代码生成依赖上下文知识，研发人员维护知识并在交互中输入知识，增加知识工程的成本
- 向用户收取很少的维护费或者免费对个人用户使用，降低使用成本，并尽可能减少对用户使用的约束（如网络环境、用户习惯使用的语言、编程环境等），最好可以开源，方便用户本地部署生成式 AI 模型
- 进一步优化页面设计与用户的交互，数据的存储等，方便用户的使用，提升用户的使用体验（如现有的快捷键有的是与 IDE 是冲突的，用户使用十分不便）

- 优化所使用的模型，进一步提高代码生成质量，生成正确（现在大部分编程助手语法错误都很多，都需要手动多次修改，或者完全不可用）的代码。并且希望生成的代码希望可以在生成阶段就考虑到代码优化方面的需求。智能问答提升回答质量，支持多语言的输入和输出，精准捕捉用户的关键需求，准确回答用户的问题。
- 目前软件研发全流程中，简单、重复的任务的交付效率将已经获得了一定的提升，但复杂任务收益不多。未来可以进一步优化所使用的大模型，提升复杂任务的收益。
- 做好软件的运营及售后服务，进一步建设并完善编程助手的社区和相关的文档，管理基础化模型并提供标准化的服务，接受用户的反馈并持续优化，提升开发者的使用体验。
- 关注数据隐私保护和 AI 安全合规，关注新形势下的软件工程面对的安全与合规的挑战。未来将面临更严峻的安全和隐私保护挑战，未来的攻击手段会愈演愈烈。数据成为了最重要的资产之一，需要了解来源、整合和使用方式。我们还需要了解模型是如何做出决策的，并能够解释其结果和推理过程，还需要在算法偏见、隐私保护、人类参与等方面进行审查和监管。明确数据使用规则，避免使用用户代码等敏感数据训练模型，以保护用户隐私。同时需要进一步明确 ai 工具提供的代码知识产权归属，避免法律风险
- 研发数据资产和知识管理，关注数据使能的软件研发体系。通过全链路工具的打通，获取各个环节的数据并提炼出其中的知识，全方位地为个体成长学思践悟的过程提供知识的支撑，提高个体获取知识的效率并降低认知负载，同时为全面协同的数据驱动的决策提供支撑。
- 关注人与流程的变化，基于人与 AI 协调的流程实践
- 从赋能个体到赋能团队，提升集体的工作效率，并与跨角色全流程的一体化平台结合，规模化发展。

5.2 设计的产品功能

- 基于设计文档直接生成可执行代码，不再需要人力将需求理解明确后再喂给编程工具。现在用户需要理解文档需求后，将需求提炼出来再给智能编程工具输入，有一个人为转述的过程，而这个过程是可能出现差错的。如果智能编程工具能够自行阅读并理解长文本的设计文档（或图表），自行提炼用户需求，根据需求完成用户的需求，就能进一步自动化，提升编程效率。
- 在智能编程助手中增加有代码托管、数据存储管理的功能，学习用户的整理习惯，并根据用户的使用习惯进行信息管理，提升使用效率。现有的智能编程助手中并没有类似功能，用户的信息存储在不同位置，每次使用时十分不方便（有时会忘记文件放在哪里了），如果智能编程助手能够很好的管理用户的重要文件（如代码文件、数据文件等），能够很方便用户的使用。
- 增加自适应与自学习功能，增强效益效率。实现个性化 AI 功能（如学习用户的代码编写风格、数据处理习惯、注释习惯、预判用户需要的文档文件等），真正成为用户的助手
- 特定编程环境加入编码规范，方便进行团队的多人代码编写工作，实现同时多人编辑并互不打扰，互相能看理解对方的代码等，便于进行团队管理。

5.3 我的开发

选择以软件形式开发这款产品。软件是一种通用的、可执行的文件，可以安装在计算机上运行，并与其他应用程序交互。这种形式允许我开发一款产品，能够在不同的操作系统上运行，并且可以方便地更新和维护。

在设计这款产品时，可以遵循以下步骤：

确定产品目标：首先，明确这款产品的主要目标和使用场景。这有助于制定开发计划和选择合适的开发语言。

选择开发语言：根据产品目标，选择一个合适的开源编程语言。这种语言应该易于学习，有强大的社区支持，并且能够满足产品的性能和功能需求。

设计用户界面：根据产品目标，设计一个简洁且易于操作的用户界面。这个界面应该能够清晰地传达信息，吸引用户注意力，并帮助用户完成所需的功能。

实现功能：在设计完用户界面后，使用所选的开源语言实现产品功能。这个过程需要遵循良好的编程实践，包括编写注释、单元测试和代码审查，以确保代码的可读性和准确性。

测试和优化：在实现功能后，需要对产品进行充分的测试。这些测试应该包括单元测试、集成测试和系统测试，以确保产品在各种情况下都能正常工作。在测试过程中，可能需要对代码进行优化，以提高性能和减少资源消耗。

发布产品：在进行充分的测试后，可以将产品发布给用户。这种发布可以是以线上形式提供服务，也可以是以安装文件的形式发送给用户。在发布后，应该监控用户的使用反馈，并根据这些反馈对产品进行持续的改进和优化。

参考文献

- [1] Zheng, Q., Xia, X., Zou, X., Dong, Y., Wang, S., Xue, Y., Shen, L., Wang, Z., Wang, A., Li, Y., Su, T., Yang, Z., & Tang, J. (2023). CodeGeeX: A Pre-Trained Model for Code Generation with Multilingual Benchmarking on HumanEval-X. Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining.
- [2] <https://zhuanlan.zhihu.com/p/619223174>
- [3] <https://codegeex.cn/>
- [4] <https://github.com/THUDM/CodeGeeX2>
- [5] <https://developer.aliyun.com/article/1378958>
- [6] <https://tongyi.aliyun.com/lingma/>
- [7] <https://codeium.com/>
- [8] <https://www.thoughtworks.com/zh-cn/insights/blog/machine-learning-and-ai/software-development-in-AI2>