



南開大學
Nankai University

计算机学院
软件工程实验报告

个人作业 1—软件调研

姓名：杨馨仪

学号：2011440

专业：计算机科学与技术

2024 年 3 月 17 日

目录

1	Copilot	2
1.1	软件概述	2
1.2	软件功能	2
1.2.1	代码补全	2
1.2.2	代码生成	3
1.2.3	代码问答	3
1.2.4	注释生成	4
1.2.5	代码测试	4
1.2.6	代码修正	5
1.3	总结	5
2	IntelliCode	5
2.1	软件概述	5
2.2	软件功能	6
2.2.1	代码补全	6
2.2.2	排序优化	6
2.3	总结	7
3	TabNine	7
3.1	软件概述	7
3.2	软件功能	7
3.2.1	代码补全	7
3.2.2	代码问答	9
3.2.3	注释生成	9
3.2.4	代码测试	10
3.2.5	代码查错	10
3.3	总结	11

1 Copilot

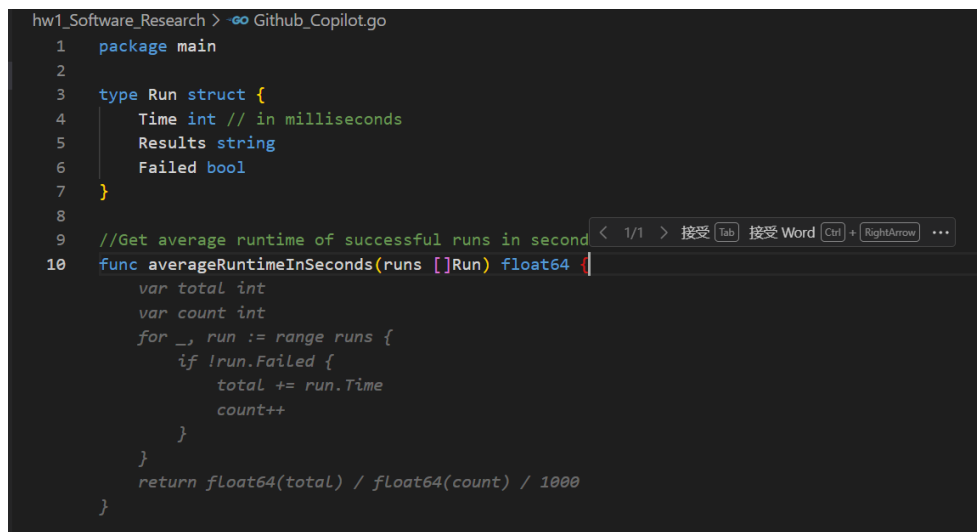
1.1 软件概述

GitHub Copilot 是由 GitHub 与 OpenAI 合作开发的人工智能工具，它支持多种集成开发环境，并主要面向 Python、JavaScript、TypeScript、Ruby 和 Go 等编程语言。Copilot 基于 OpenAI Codex，这是一个由 GPT-3 模型修改而来的人工智能模型，能够理解自然语言并生成对应的代码解决方案。该工具训练于大量的英语语言和公共 GitHub 存储库的数据集，并在 GitHub 网站中提供了诸如代码转换、自动补全、代码重复消除等功能。

1.2 软件功能

1.2.1 代码补全

如下图所示，我们仅需要输入代码的功能描述以及代码的方法名称，Copilot 会自动识别代码的上下文，等待 2 3 秒即可生成完整的方法片段。



```
hw1_Software_Research > Github_Copilot.go
1 package main
2
3 type Run struct {
4     Time int // in milliseconds
5     Results string
6     Failed bool
7 }
8
9 //Get average runtime of successful runs in second
10 func averageRuntimeInSeconds(runs []Run) float64 {
    var total int
    var count int
    for _, run := range runs {
        if !run.Failed {
            total += run.Time
            count++
        }
    }
    return float64(total) / float64(count) / 1000
}
```

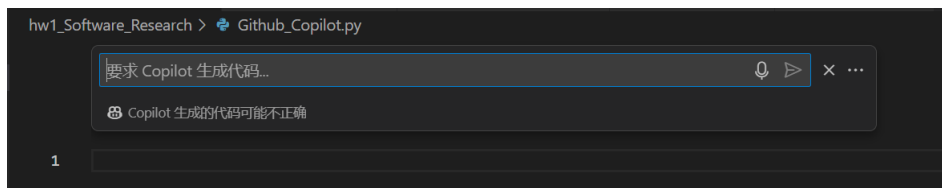
除此之外，Copilot 还能自动填充具备固定模式的重复代码



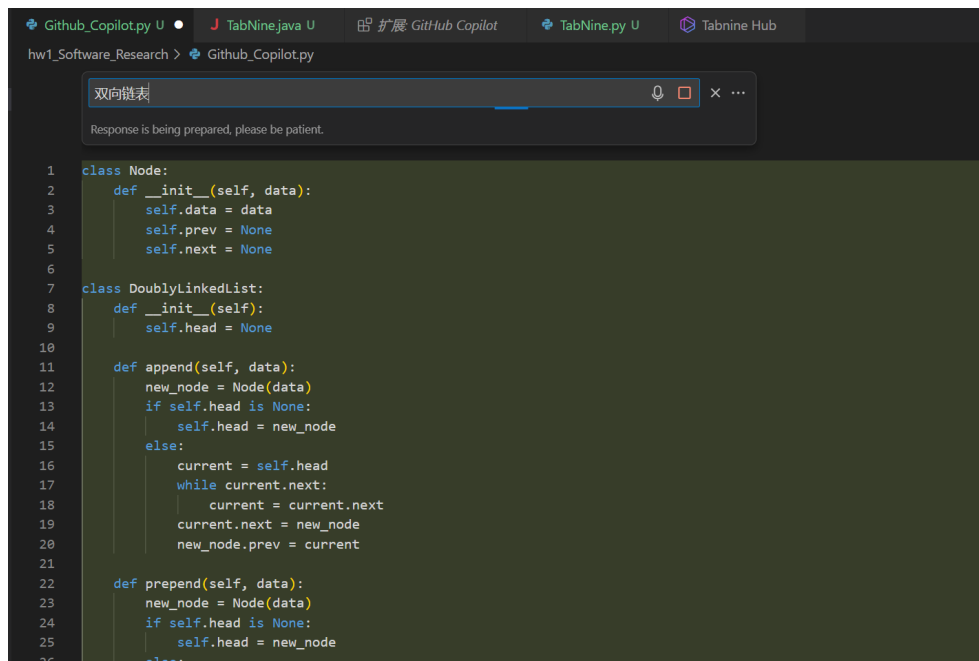
```
J Github_Copilot.java U J TabNine.java U 扩展 Github Copilot
hw1_Software_Research > J Github_Copilot.java
1 interface CommentMarker {
2     start: string;
3     end: string;
4 }
5
6 const markers: { [language:string]:CommentMarker } = {
7     javascript: { start: '/*', end: '*/' },
8     python: { start: '"""', end: '"""' },
9     css: { start: '/*', end: '*/' },
10    html: { start: '<!--', end: '-->' },
}
```

1.2.2 代码生成

点击旁边的黄色星星图标就会询问用户要求 copilot 生成什么样的代码

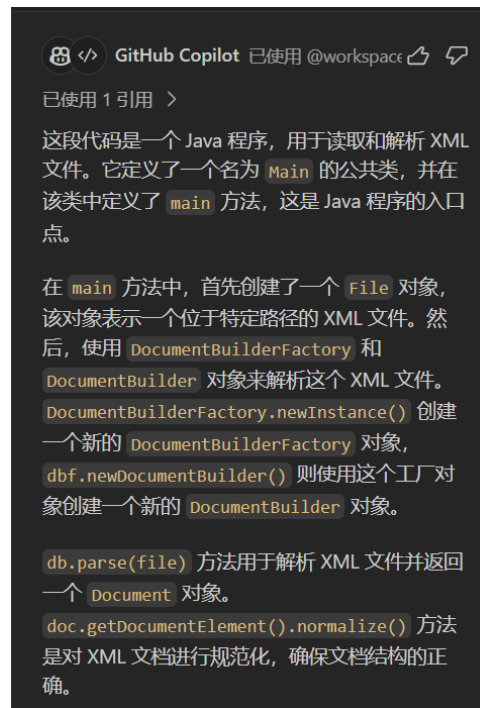


这里我尝试让他生成双向链表，copilot 可以直接把整个类生成出来。之后我在 TabNine 中尝试，他只能一行一行或一个函数一个函数的实现，无法像 copilot 这样有效率的完成。



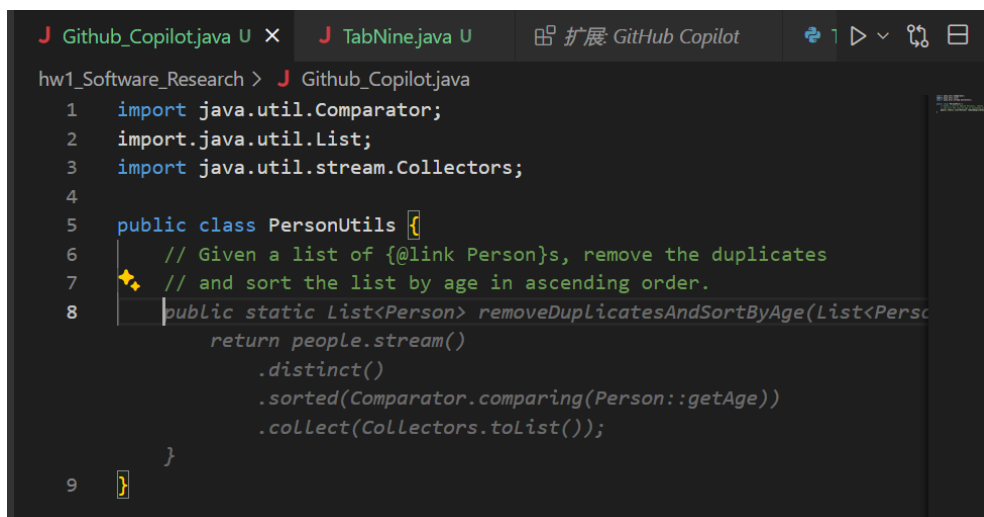
1.2.3 代码问答

可以选中某一段代码，然后左键询问这段代码的作用，copilot 会回答在左侧的对话框里，在这个对话框中也可以询问其他技术问题。



1.2.4 注释生成

copilot 可以实现根据代码注释生成代码片段，这个功能在对 Intellicode 和 TabNine 的测试中是无法实现的。



1.2.5 代码测试

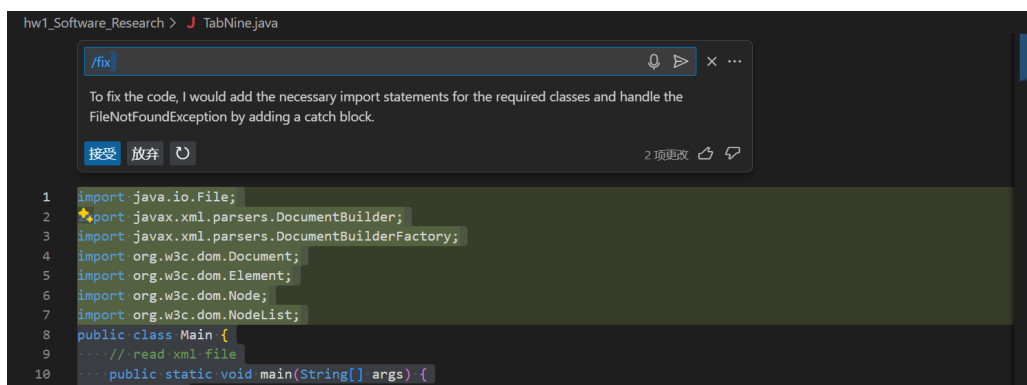
用 Copilot 写单元测试是非常方便的，它能帮助我生成很多测试数据，减少 Mock 数据的烦恼。



```
hw1_Software_Research > Github_Copilot.py > ...
tabnine: test | explain | document | ask
1 def strip_suffix(filename):
2     """
3     Removes the suffix from a filename.
4     """
5     return filename[:filename.rfind('.')]
6
7 def test_strip_suffix():
    assert strip_suffix('file.txt') == 'file'
    assert strip_suffix('file.tar.gz') == 'file.tar'
    assert strip_suffix('file') == 'file'
    assert strip_suffix('file.') == 'file'
    assert strip_suffix('.file') == ''
    assert strip_suffix('') == ''
    assert strip_suffix('file.txt.txt') == 'file.txt'
```

1.2.6 代码修正

跟代码问答类似，可以对已经写好的代码进行纠错。



```
hw1_Software_Research > TabNine.java

/fix
To fix the code, I would add the necessary import statements for the required classes and handle the
FileNotFoundException by adding a catch block.
接受 放弃 刷新 2 项更改

1 import java.io.File;
2 import javax.xml.parsers.DocumentBuilder;
3 import javax.xml.parsers.DocumentBuilderFactory;
4 import org.w3c.dom.Document;
5 import org.w3c.dom.Element;
6 import org.w3c.dom.Node;
7 import org.w3c.dom.NodeList;
8 public class Main {
9     // read xml file
10    public static void main(String[] args) {
```

1.3 总结

Copilot 的模型非常强大，其原理本质上跟 TabNine 是相似的，但是 Copilot 的代码生成效果要好很多，毕竟模型大小不是一个层次的。Copilot 代码生成结果的展现形式跟其他工具都不同，它采用了内联的方式，这种方式非常直观的展示的代码生成后的样子，适合多行代码片段的展示，但是这种方式不适合推荐结果比较多时，只能通过快捷键切换结果，而且如果开启了自动触发，在写代码时经常会在我不需要的时候出现，把正在编写的代码挤到下面去，这对我的日常编码是一种打扰。

2 InteliCode

2.1 软件概述

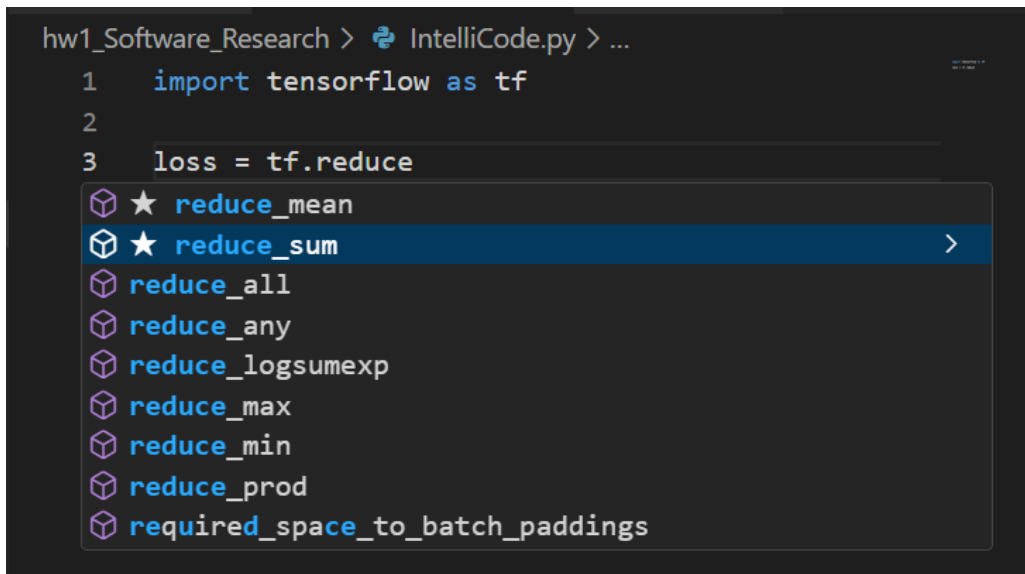
IntelliCode 可通过代码补全来帮助你提高准确性和一致性，它可用一次性填写整行代码。AI 会检测代码上下文，包括变量名称、函数和正在编写的代码类型，以提供最佳建议。更棒：IntelliCode 在计算机上运行，确保专用代码保持私有。IntelliCode 将你最可能使用的内容放在 IntelliSense 完成列表的

顶部。这些节省时间的建议基于分析 GitHub 上的数千个开放源代码贡献。对于某些语言，甚至可以添加自己的自定义方法和代码基属性。

2.2 软件功能

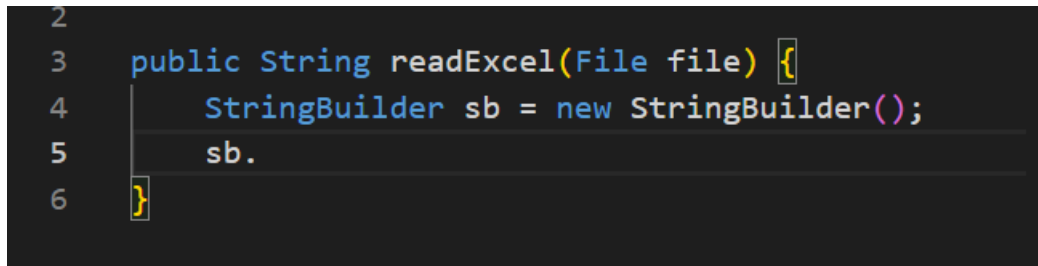
2.2.1 代码补全

下图是在 python 语言下的测试，可以看到 intellicode 之后提示 API，而无法像其他两种 AI 编程工具一样提示一行或整段代码



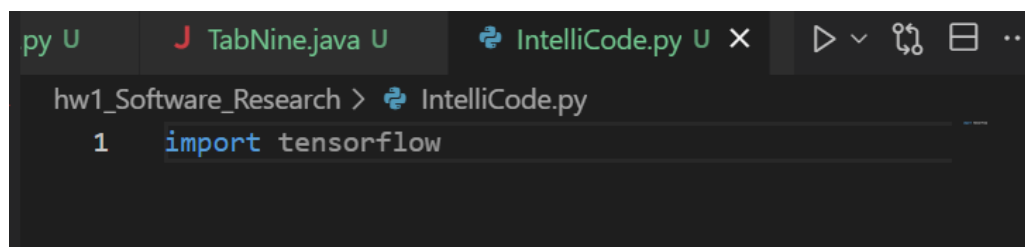
```
hw1_Software_Research > IntelliCode.py > ...
1  import tensorflow as tf
2
3  loss = tf.reduce
  ★ reduce_mean
  ★ reduce_sum >
  reduce_all
  reduce_any
  reduce_logsumexp
  reduce_max
  reduce_min
  reduce_prod
  required_space_to_batch_paddings
```

下图是在 java 语言下的测试，可以看到 intellicode 的能力变得更差了。



```
2
3  public String readExcel(File file) {
4      StringBuilder sb = new StringBuilder();
5      sb.
6  }
```

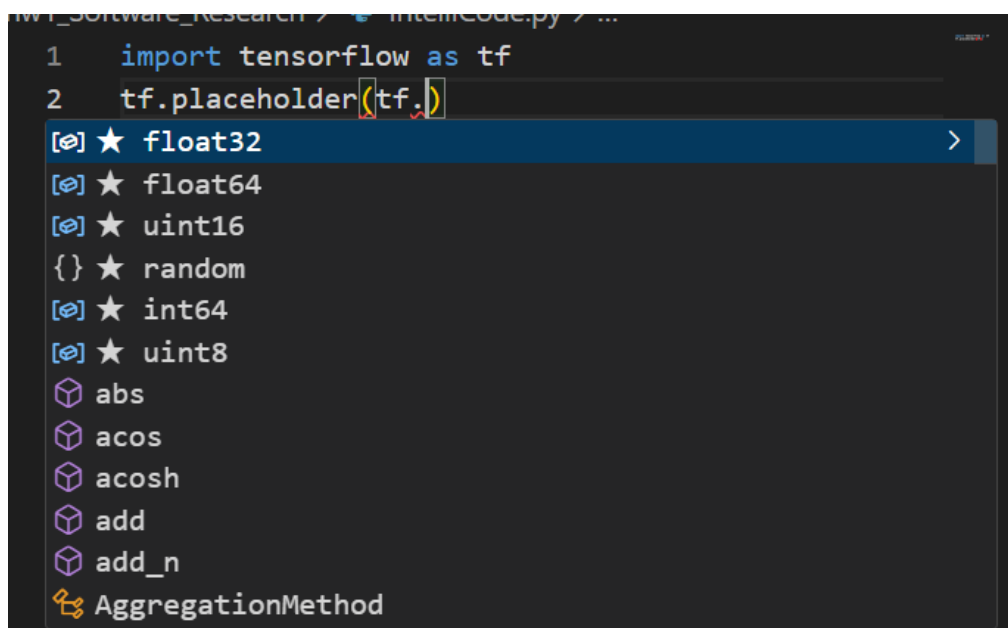
此外当我 Import tensorflow 时，他没有为我在推荐之后的代码，而在我使用 copilot 时，他会补充上 as tf。可见 copilot 由大量的代码训练而成，更了解人们的编程习惯。



```
py U  J TabNine.java U  IntelliCode.py U x
hw1_Software_Research > IntelliCode.py
1  import tensorflow
```

2.2.2 排序优化

intellicode 的一个特色在于排序优化，会把更常用到的内容排在前面，但有时这样的预测并不准确



2.3 总结

IntelliCode 能对 IDE 自带补全结果进行排序优化，但是支持的场景比较少，并不是所有补全都能提供排序优化，目前体验下来仅局限于方法调用的代码。整体体验下来，感觉 IntelliCode 在 Java 语言上实用性不强，基本上没有体感；在 Python 语言上有时候还是比较方便的，适合不喜欢长序列/片段级代码补全工具的开发者。

3 TabNine

3.1 软件概述

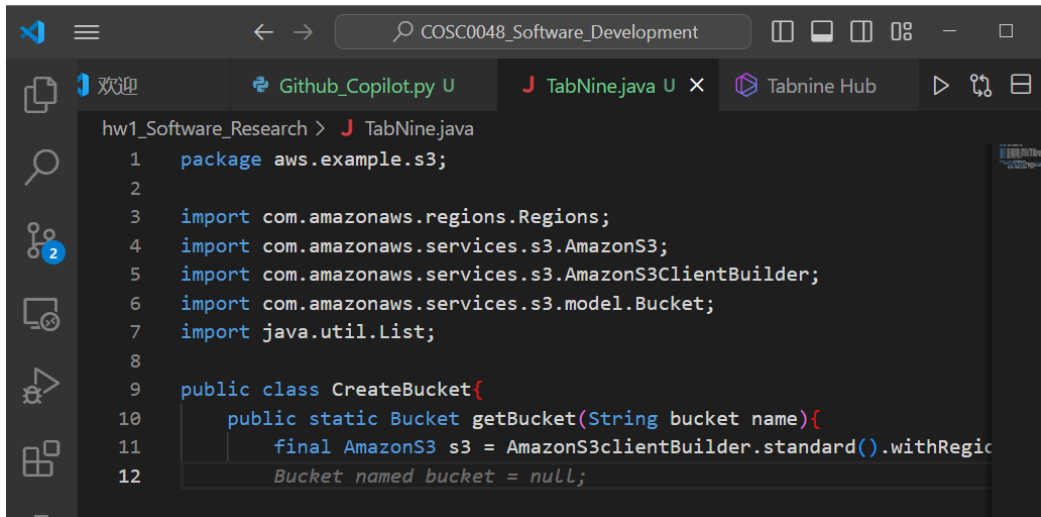
Tabnine 是一款人工智能驱动的代码补全工具，旨在提高开发者的编码效率。它可以在编程过程中自动推荐代码片段，并预测开发者可能要输入的代码，从而加快编写代码的速度。Tabnine 使用机器学习技术，通过分析代码库中的大量代码示例来生成代码建议，并根据上下文和开发者的输入进行调整。该工具支持多种集成开发环境（IDE）和编程语言，如 Visual Studio Code、IntelliJ IDEA、PyCharm、Sublime Text 等，涵盖了包括 Python、JavaScript、Java、C++ 等在内的多种主流编程语言。Tabnine 的目标是为开发者提供高效、准确的代码补全功能，从而节省时间和提高编码效率。

3.2 软件功能

3.2.1 代码补全

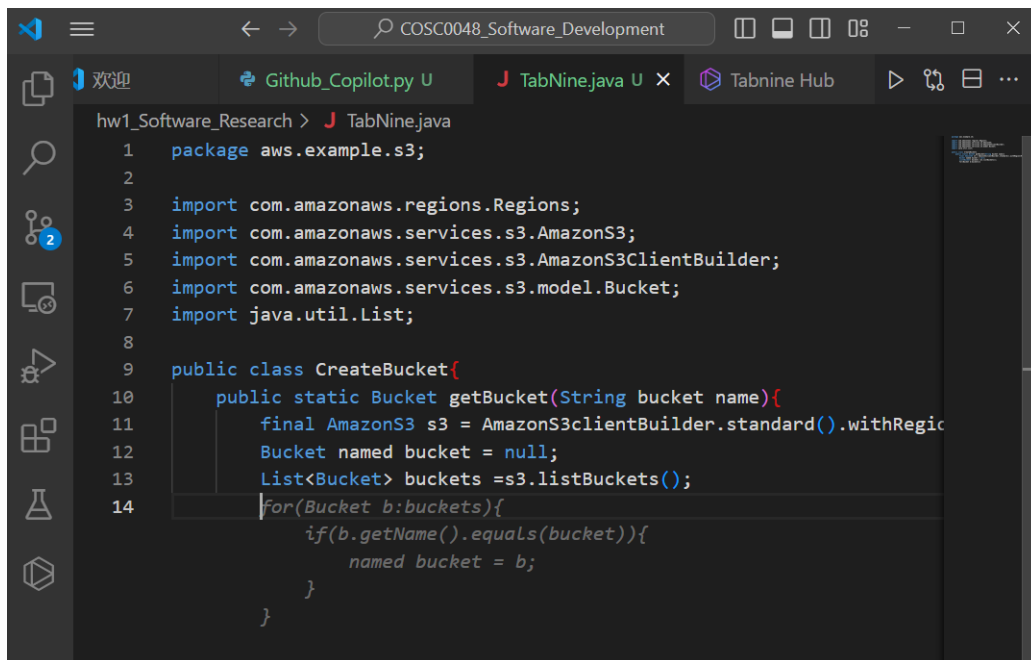
TabNine 可以实现整行补全和代码段补全，但性能比 copilot 要差，比如我让他写双鞋链表就无法写出。

1. 整行补全



```
hw1_Software_Research > J TabNine.java
1 package aws.example.s3;
2
3 import com.amazonaws.regions.Regions;
4 import com.amazonaws.services.s3.AmazonS3;
5 import com.amazonaws.services.s3.AmazonS3ClientBuilder;
6 import com.amazonaws.services.s3.model.Bucket;
7 import java.util.List;
8
9 public class CreateBucket{
10     public static Bucket getBucket(String bucket name){
11         final AmazonS3 s3 = AmazonS3clientBuilder.standard().withRegio
12         Bucket named bucket = null;
```

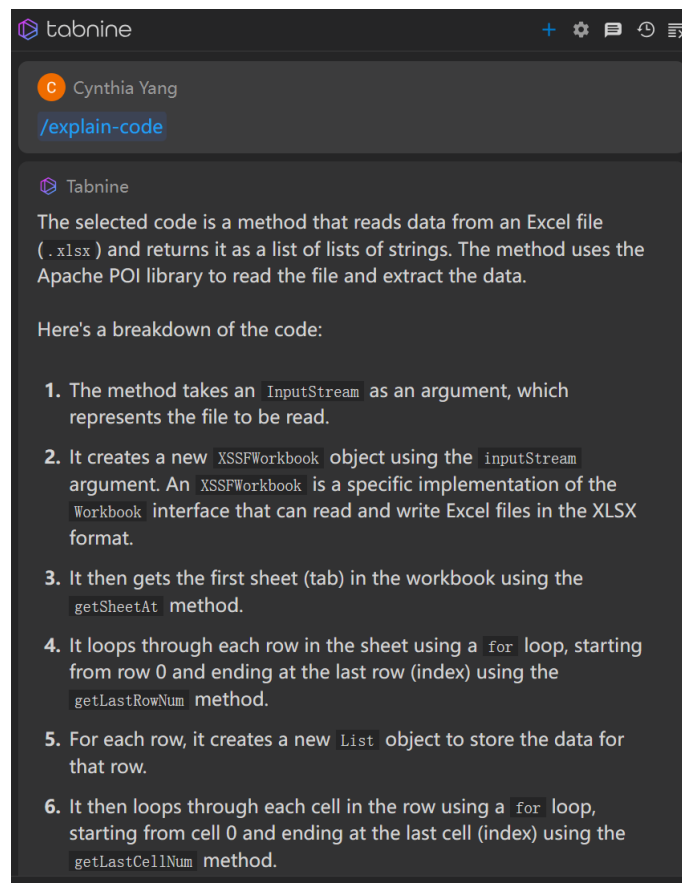
2. 代码段补全



```
hw1_Software_Research > J TabNine.java
1 package aws.example.s3;
2
3 import com.amazonaws.regions.Regions;
4 import com.amazonaws.services.s3.AmazonS3;
5 import com.amazonaws.services.s3.AmazonS3ClientBuilder;
6 import com.amazonaws.services.s3.model.Bucket;
7 import java.util.List;
8
9 public class CreateBucket{
10     public static Bucket getBucket(String bucket name){
11         final AmazonS3 s3 = AmazonS3clientBuilder.standard().withRegio
12         Bucket named bucket = null;
13         List<Bucket> buckets = s3.listBuckets();
14         for(Bucket b:buckets){
15             if(b.getName().equals(bucket)){
16                 named bucket = b;
17             }
18         }
19     }
20 }
```

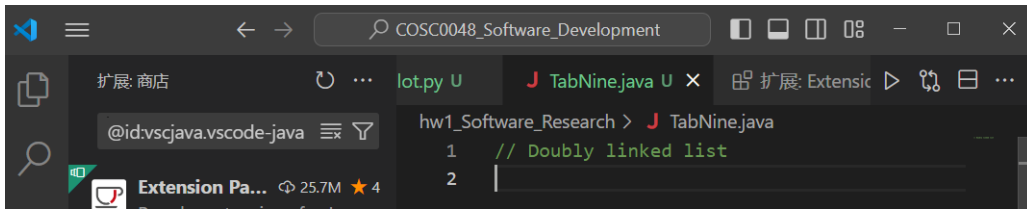
3.2.2 代码问答

也是通过右键可以帮助解释代码



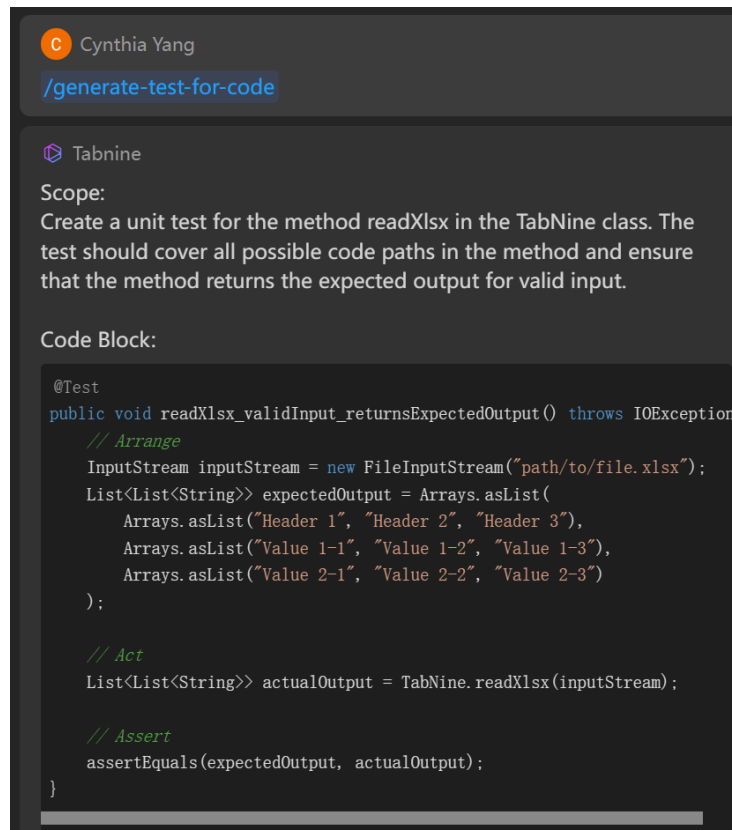
3.2.3 注释生成

尝试之后发现不存在该功能



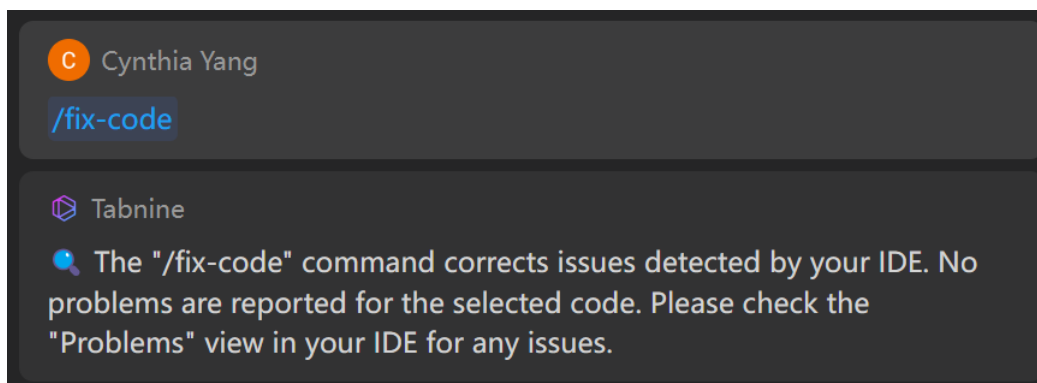
3.2.4 代码测试

右键可以帮助生成测试代码



3.2.5 代码查错

右键可以帮助代码差错



3.3 总结

TabNine 能提供长序列的代码补全，在我体验过的同类工具中，TabNine 是支持的开发语言种类以及 IDE 平台最多的工具。但是与其他工具相同，它会推荐出不完整的代码以及存在语法错误的代码。从部分开发者的反馈了解到，有部分开发者喜欢 TabNine 即时学习代码模式的能力，这点是相比其他工具比较不错的，虽然目前提取代码模式的能力还存在比较多的缺陷，但是相信未来会逐渐完善。