SmartCode AI 代码助手
使用指南

文档版本 01

发布日期 2025-04-17

英捷创软科技(北京)有限公司
leansoftx.com

注意

您购买的产品、服务或特性等应受英捷创软科技(北京)有限公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，英捷创软科技(北京)有限公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

英捷创软科技(北京)有限公司

地址：北京市海淀区中关村西区善缘街 1 号

邮箱：info@leansoftx.com

网址： https://leansoftx.com

目录

一、什么是 SmartCode AI 代码助手

SmartCode 是为使用 Visual Studio IDE 的开发者提供的简单易用的 AI 编程助手插件，基于华为盘古研发大模型的智能开发助手，重塑了智能化软件研发的新范式，让开发者更加聚焦业务创新，事半功倍。插件基于智能生成、智能问答两大核心能力，覆盖了代码生成、研发知识问答、单元测试用例生成、代码解释、代码注释、代码调试、代码翻译、代码检查等开发场景，释放软件研发生产力。

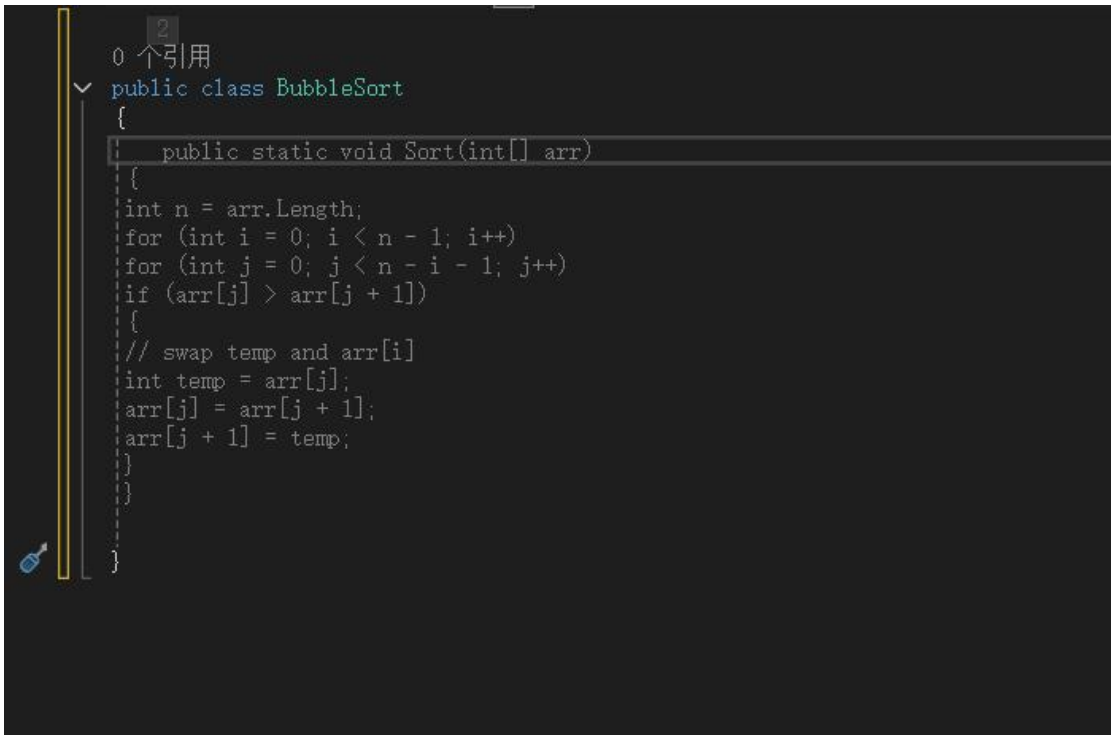
二、使用说明

2.1 生成代码

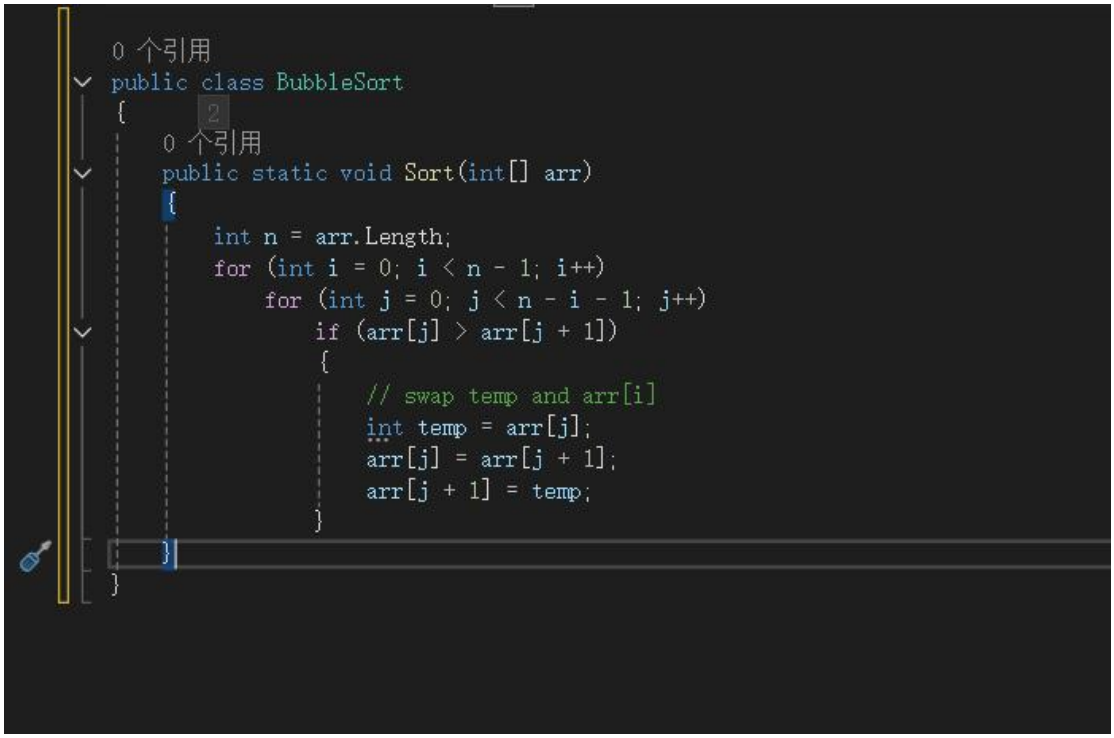
代码助手支持通过快捷键在 IDE 中触发根据代码上下文生成代码，也可以在研发对话窗口使用代码注释或自然语言描述生成代码。

使用快捷键通过上下文生成代码

1、打开一个 c#文件，将编辑光标移动至需要生成代码位置，按下快捷键“Alt+C”。



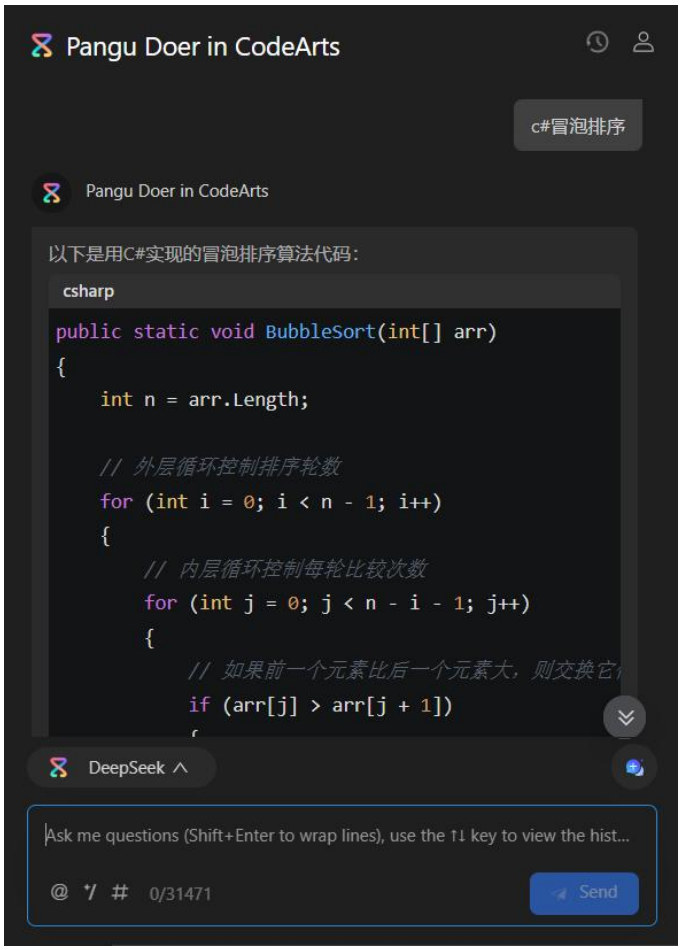
2、代码助手将根据上下文生成代码，按下 Tab 键可接受代码助手生成代码内容。



通过问答生成代码

1、在研发对话窗口输入框中输入生成代码需求，如：“C#冒泡排序”，单击研发对话窗口输入框右下角发送按钮发送。

2、代码助手将在研发对话窗口中生成 Java 冒泡排序代码。



3、单击输入框右上角  可以将对话内容归档并新建会话，单击研发对话窗口右上角  可以查看历史提问。

4、对代码助手生成的代码块，可以进行如下操作：

单击  复制代码。

单击  在当前光标位置插入代码。

单击  将代码另存为文件。

5、对代码助手回答的内容，可以进行如下操作：

- 单击  可以针对提问重新生成结果。
- 单击  可以复制回答内容。

- 单击  对回答满意。
- 单击  对回答不满意。

2.2 解释代码

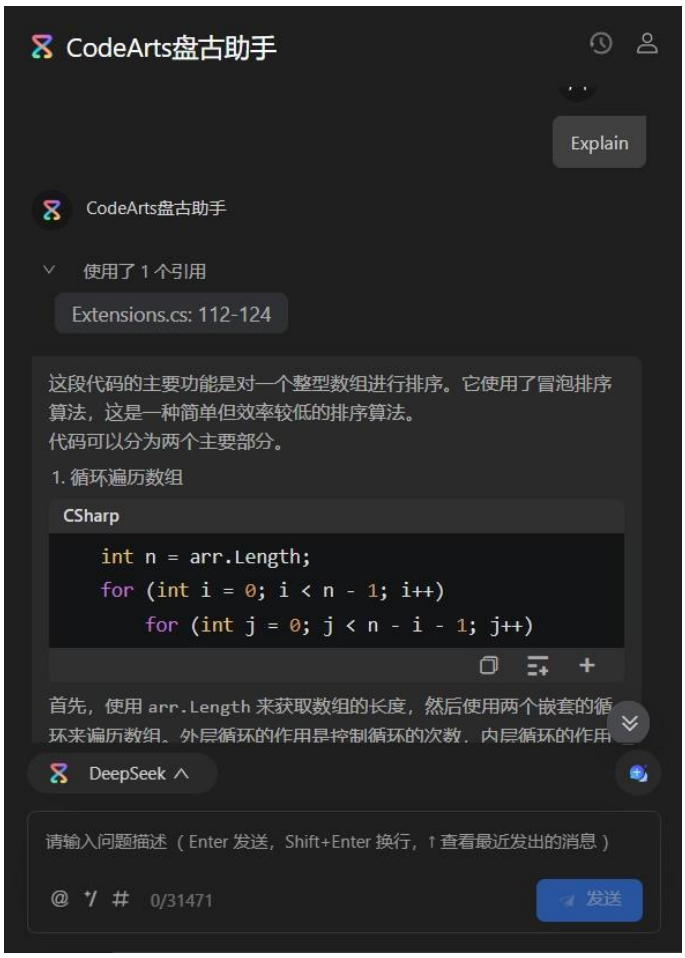
如果开发人员对代码存在疑惑，可以使用代码助手代码解释功能自动分析代码的结构和逻辑，对代码功能进行解释，帮助开发人员理解代码的功能和实现方式。

通过问答功能解释代码

1、选中示例代码中“true”方法代码，单击右键，选择菜单“CodeArts 盘古助手：Add to Chat”或使用快捷键 Ctrl+Shift+X 将代码添加至研发对话窗口。

2、在研发对话窗口输入框中输入“/”，在弹出菜单中选择“/explain”，或单击研发对话窗口中“Code Explain”，单击【发送】按钮发送。

3、代码助手将对代码进行解释，通过文字描述帮助开发人员理解代码。针对本次选中的代码，代码助手给的解释是：这段代码的主要功能是对一个整型数组进行排序。它使用了冒泡排序算法，这是一种简单但效率较低的排序算法。



2.3 注释代码

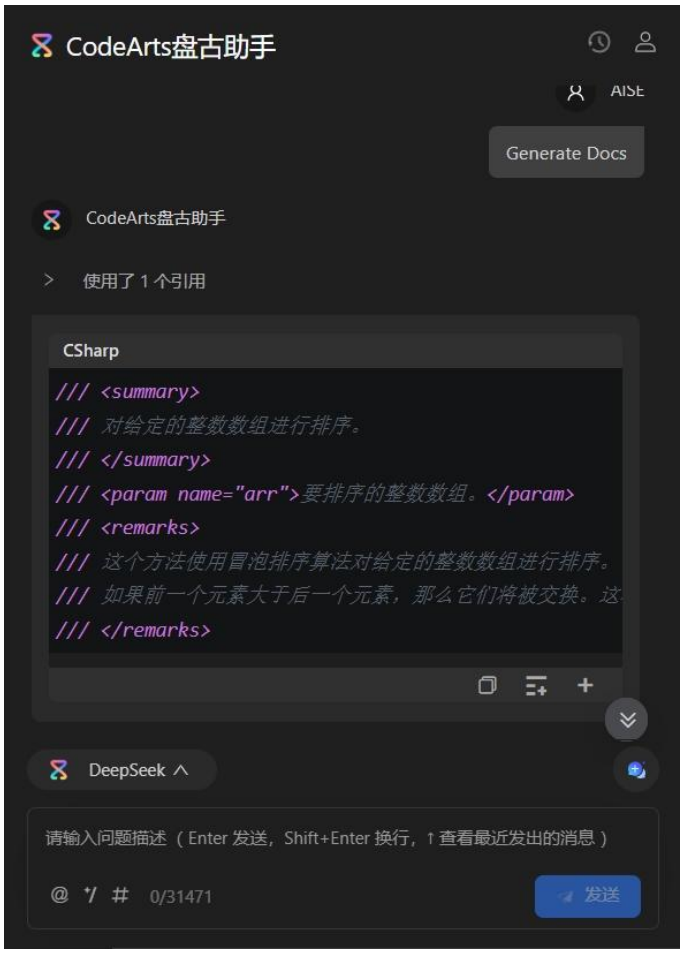
代码开发完成后，使用代码助手代码注释功能可以为代码添加详细的注释说明，包括函数、变量、类的作用、参数、返回值信息，帮助开发人员更好地理解代码逻辑和实现方式，提高代码可读性和可维护性，同时也方便后续的代码维护和修改工作。

注释代码

1、选中示例代码，单击右键，选择菜单“CodeArts 盘古助手：Add to Chat”或使用快捷键 Ctrl+Shift+X 将代码添加至研发对话窗口。

2、在研发对话窗口输入框中输入“/”，在弹出菜单中选择“/comment”或单击研发对话窗口中“Code Comment”，单击【发送】按钮发送。

3、代码助手将对代码进行注释，通过文字描述帮助开发人员理解代码。



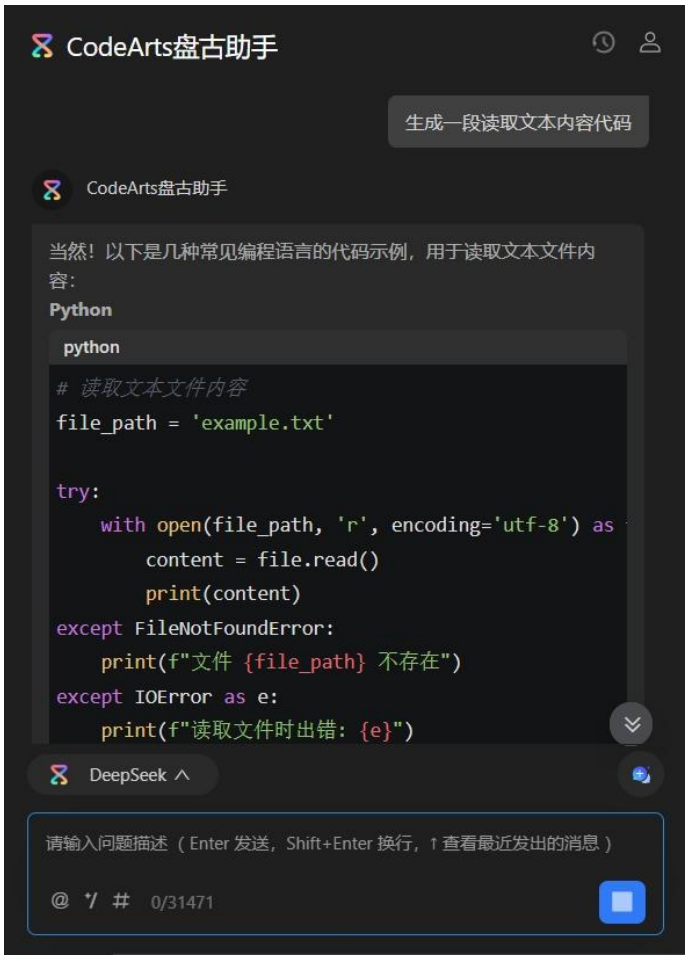
2.4 研发知识问答

在代码助手研发对话窗口中，用户可以随时提出问题，而系统则会快速检索研发相关知识，并提供匹配答案，从而帮助用户高效地解决问题。

研发知识问答

1、在研发对话窗口中输入研发相关问题“生成一段读取文本内容代码”。

2、代码助手将使用 Python 或其他语言生成一段读取文本内容的代码。


CodeArts盘古助手
生成一段读取文本内容代码
CodeArts盘古助手
当然！以下是几种常见编程语言的代码示例，用于读取文本文件内容：
Python
python
# 读取文本文件内容
file_path = 'example.txt'
try:
with open(file_path, 'r', encoding='utf-8') as
content = file.read()
print(content)
except FileNotFoundError:
print(f"文件 {file_path} 不存在")
except IOError as e:
print(f"读取文件时出错: {e}")
DeepSeek
请输入问题描述（Enter 发送，Shift+Enter 换行，↑查看最近发出的消息）
0/31471

三、应用场景

代码智能推荐与补全

- 高效编写新代码：在编写代码时，代码助手能够基于对代码上下文的理解，实时推荐相关的代码片段、函数、变量名等。例如，当用户在编写一个循环结构时，它会推荐常用的循环语法和相关函数，帮助用户快速完成代码的编写，提高编码速度和准确性。
- 减少重复劳动：对于一些常用的代码模式，如条件判断、异常处理等，代码助手可以自动补全代码，避免重复编写相同的代码，节省时间和精力。
- 多行代码推荐：除了单行代码补全，代码助手还支持多行代码推荐。例如，在编写一个复杂的算法或函数时，它可以根据用户的意图和上下文，推荐完整的代码块，帮助用户快速构建代码逻辑。

代码快速熟悉与理解

- 功能与逻辑快速理解：当接手新的项目或研究开源代码时，代码助手可以从功能、目的、使用场景和主要逻辑多个维度帮助用户快速理解代码、复杂数据结构与算法分析。例如，在阅读一个复杂的系统代码时，代码助手可以详细解释其功能和实现逻辑。
- 解读非标准化代码和注释：有时候，用户可能会遇到一些非标准化的代码或没有详细注释的代码。代码助手可以解读这些代码，提供通俗易懂的注释和说明，帮助用户理解代码的意图和功能。例如，在一些历史遗留代码中，可能存在一些晦涩难懂的代码，代码助手会解释这些代码的功能、目的、使用场景和主要逻辑。

代码调试与优化

- 快速定位代码错误：在代码运行报错时，代码助手通过分析堆栈信息和代码逻辑，能够快速定位代码中的错误和异常。例如，当运行代码报错时，在报错处单击代码助手图标，它会根据堆栈信息自动识别出可能导致问题的代码行或模块，并提供相关的上下文信息和建议，帮助用户快速找到错误根源，节省调试时间。
- 性能优化建议：代码助手能够分析代码的性能瓶颈，提供优化建议。例如，当用户检查代码时，它会识别出可能导致性能问题的代码片段，如低效的循环、不必要的计算等，并建议用户使用更高效的算法或数据结构来优化代码，提高程序的运行速度和响应能力。
- 代码可读性与可维护性优化：代码助手可以帮助用户提高代码的可读性和可维护性。例如，它会根据代码的结构和逻辑，建议用户对代码进行优化，如简化复杂的逻辑、提取重复的代码片段、增强注释和文档的清晰度等，使代码更加简洁、易读和易维护，降低后续代码修改和扩展的难度。

单元测试用例生成

- 功能验证测试用例生成：在编写新功能或修改现有功能时，代码助手可以基于代码实现自动为功能模块创建单元测试用例。例如，当用户编写一个处理用户登录的函数时，它会生成涵盖正常登录、无效用户名、错误密码等各种情况的测试用例，确保函数行为的正确性。
- 边界条件测试用例生成：代码助手能够识别代码中的边界条件，比如数组为空、字符串为空或达到最大长度等情况，并自动生成相应的测试用例。这些测试用例有助于检测代码在极端情况下的健壮性，提前发现潜在的边界问题。
- 异常处理测试用例生成：在开发过程中，异常处理是容易被忽略但又非常关键的环节。代码助手会根据代码逻辑，自动为可能引发异常的代码路径生成测试用例，如模拟网络错误、数据库连接失败等场景，确保代码能够正确捕获和处理异常。

- 高质量测试用例生成：代码助手能够分析代码逻辑和业务需求，生成既覆盖正常情况又兼顾异常场景的高质量测试用例。例如，对于一个电商网站的购物车功能，它会生成测试用例来验证添加商品、更新商品数量、删除商品等操作在不同条件下的正确性，以及应对库存不足、价格变更等异常情况的能力。

四、功能特性

- 支持多种编程语言，并能根据开发者键入的函数签名和注释自动生成函数体。
- 支持根据行级注释或代码上下文信息自动生成与描述场景匹配的代码。
- 可根据开发者当前光标位置的前后语句片段进行代码填空和补全。
- 支持跨文件生成与任务相关的代码。
- 可根据用户需求内容生成行级、函数级注释信息，能够帮助开发人员高效补充代码注释。
- 可根据输入的代码和错误信息，得到错误原因并给出修复方案。
- 支持生成高覆盖率的单元测试代码，包括单个方法和类级别的单元测试框架代码。
- 可根据提问来检索研发相关知识，提供答案。
- 支持对代码进行函数级检查功能，可及时、主动发现编码缺陷，提升代码质量和安全性。

五、约束与限制

本节介绍了 SmartCode AI 代码助手中的限制，如表 5-1 所示

表 5-1 使用限制说明

指标类型	指标项	限制说明
IDE 版本	版本要求	目前支持 Visual Studio 2019 和 Visual Studio 2022 两个版本，具体版本要求如下 Visual Studio 2019: ● 16.10 及以上。 Visual Studio 2022 ● 17.12 及以上。
add to chat 的代码长度	代码长度（字符）	代码长度<=10000。
对话框的输入文本长度	文本长度（字符）	<=1200。
模型的 token 数限制	数量限制（token）	<=4096，当输入及输出总的 token 数达到上 限后，模型无法返回新的内容，可能导致返 回代码或文本不完整。
单用户访问速度限制	速度限制（请求数/每分钟）	单用户访问速度限制 60 次/每分钟。