

CodeArts 智能助手

用户指南

文档版本 01
发布日期 2025-09-15



版权所有 © 华为云计算技术有限公司 2025。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为云计算技术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，华为云计算技术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。

华为云计算技术有限公司

地址：贵州省贵安新区黔中大道交兴功路华为云数据中心 邮编：550029

网址：<https://www.huaweicloud.com/>

目录

1 CodeArts 智能助手使用流程.....	1
2 申请并授权使用 CodeArts 智能助手.....	3
3 管理 CodeArts 智能助手.....	7
3.1 统计看板.....	7
3.2 知识库中心.....	9
3.2.1 知识库概述.....	9
3.2.2 新建知识库.....	10
3.2.3 新建知识集.....	12
3.2.4 管理知识集文件.....	14
3.2.5 查看知识库操作日志.....	15
3.3 租户成员管理.....	16
3.4 项目成员管理.....	17
4 使用 CodeArts 编码智能助手插件.....	18
4.1 安装并登录 CodeArts 编码智能助手插件.....	18
4.1.1 在 JetBrains 系列 IDE 安装 CodeArts 编码智能助手插件.....	18
4.1.2 在 VSCode 安装 CodeArts 编码智能助手插件.....	20
4.1.3 在 CodeArts IDE 安装 CodeArts 编码智能助手.....	20
4.2 使用 CodeArts 编码智能助手生成代码.....	21
4.2.1 通过问答描述生成代码.....	21
4.2.2 通过编辑器上下文生成代码.....	24
4.3 使用 CodeArts 编码智能助手解释代码.....	26
4.4 使用 CodeArts 编码智能助手注释代码.....	28
4.5 使用 CodeArts 编码智能助手调试代码.....	31
4.6 使用 CodeArts 编码智能助手翻译代码.....	34
4.7 使用 CodeArts 编码智能助手检查代码.....	36
4.8 使用 CodeArts 编码智能助手优化代码.....	38
4.8.1 优化选中代码.....	38
4.8.2 修复问题代码.....	41
4.9 使用 CodeArts 编码智能助手生成单元测试用例.....	43
4.10 使用 CodeArts 编码智能助手进行研发问答.....	49
4.11 使用知识库增强 CodeArts 编码智能助手能力.....	53

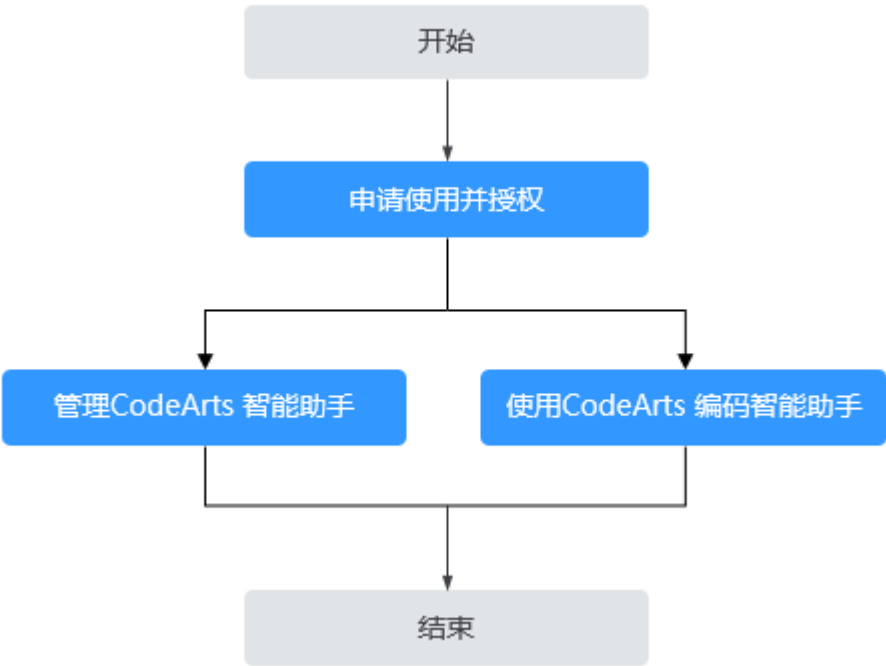
4.12 使用 AgentMode 智能体进行研发问答..... 55

1 CodeArts 智能助手使用流程

使用流程

介绍CodeArts 智能助手使用流程。

图 1-1 CodeArts 智能助手使用流程



流程说明如下：

表 1-1 CodeArts 智能助手使用流程

操作	说明
申请使用并授权	获取CodeArts 智能助手使用权限。

操作	说明
管理CodeArts 智能助手	CodeArts 智能助手的权限管理、知识库管理、租户级管理等。
使用CodeArts 编码智能助手	在IDE中安装CodeArts 编码智能助手插件，触发CodeArts 编码智能助手代码生成、研发知识问答、单元测试用例生成、代码解释、代码注释、代码翻译、代码调试、代码检查功能。

2 申请并授权使用 CodeArts 智能助手

前提条件

已[注册华为账号并开通华为云](#)。

申请 CodeArts 编码智能助手

在使用CodeArts 编码智能助手之前，您需要先获取使用权限（因资源有限，仅向企业用户开放，个人用户暂不开放）。

步骤1 进入[CodeArts 编码智能助手申请页面](#)。

步骤2 填写申请信息，单击“申请公测”提交申请（当前仅支持IAM租户进行申请）。

- 系统将在2至3个工作日内处理您的申请并授权试用。如果您在这段时间后仍未获得权限，请及时联系华为云客服，以便我们协助您确认服务开通状态。
- 租户成功申请CodeArts 智能助手后，请参考[租户成员管理](#)将成员添加至租户成员清单，被添加的成员才能使用CodeArts 智能助手。

----结束

管理项目级权限

CodeArts支持在项目下统一配置项目各角色对当前项目下CodeArts 编码智能助手资源的默认操作权限。

步骤1 登录[CodeArts控制台](#)。

步骤2 单击“前往工作台”，进入CodeArts首页。

步骤3 单击项目名称，进入项目。

步骤4 单击左侧导航“设置 > 权限管理”，进入“权限管理”页面。

CodeArts 编码智能助手相关资源集中在“智能开发助手”节点下，包括知识库、提示词模板、知识库资产和知识库知识。

----结束

- **知识库权限**

初始状态下，项目下各角色具有的知识库权限如下表所示：

表 2-1 项目级知识库权限

角色	查看	修改	删除	创建
项目管理员	√	√	√	√
项目经理	√	√	√	√
产品经理	√	×	×	√
测试经理	√	×	×	√
运维经理	√	×	×	√
系统工程师	√	×	×	√
Committer	√	×	×	√
开发人员	√	×	×	√
测试人员	√	×	×	√
参与者	√	×	×	×
浏览者	√	×	×	×

- 提示词模板权限
初始状态下，项目下各角色具有的提示词模板权限如下表所示：

表 2-2 项目级提示词模板权限

角色	创建	修改	删除	查询	下线	检索	调试	评估
项目管理员	√	√	√	√	√	√	√	√
项目经理	√	√	√	√	√	√	×	×
产品经理	√	√	√	√	√	√	×	×
测试经理	√	√	√	√	√	√	×	×
运维经理	√	√	√	√	√	√	×	×
系统工程师	√	√	√	√	√	√	×	×
Committer	√	√	√	√	√	√	×	×

角色	创建	修改	删除	查询	下线	检索	调试	评估
开发人员	√	√	√	√	√	√	×	×
测试人员	√	√	√	√	√	√	×	×
参与者	√	√	√	√	×	√	×	×
浏览者	×	×	×	√	×	√	×	×

- 知识库资产权限**
初始状态下，项目下各角色具有的知识库资产权限如下表所示：

表 2-3 项目级知识库资产权限

角色	查看	修改	删除	创建	初始化
项目管理员	√	√	√	√	√
项目经理	√	√	√	√	√
产品经理	√	×	×	√	√
测试经理	√	×	×	√	√
运维经理	√	×	×	√	√
系统工程师	√	×	×	√	√
Committer	√	×	×	√	√
开发人员	√	×	×	√	√
测试人员	√	×	×	√	√
参与者	√	×	×	×	×
浏览者	√	×	×	×	×

- 知识库知识权限**
初始状态下，项目下各角色具有的知识库知识权限如下表所示：

表 2-4 项目级知识库知识权限

角色	查看	更新	删除	创建	上传	检索
项目管理员	√	√	√	√	√	√

角色	查看	更新	删除	创建	上传	检索
项目经理	√	√	√	√	√	√
产品经理	√	×	×	√	×	√
测试经理	√	×	×	√	×	√
运维经理	√	×	×	√	×	√
系统工程师	√	×	×	√	×	√
Committer	√	×	×	√	×	√
开发人员	√	×	×	√	×	√
测试人员	√	×	×	√	×	√
参与者	√	×	×	×	×	√
浏览者	√	×	×	×	×	√

3 管理 CodeArts 智能助手

3.1 统计看板

CodeArts 编码智能助手统计看板

CodeArts 智能助手支持企业租户查询插件使用的统计看板。看板对用户数、代码生成及接纳的次数、行数做了统计分析 & 细粒度的趋势分析。

图 3-1 CodeArts 编码智能助手统计看板

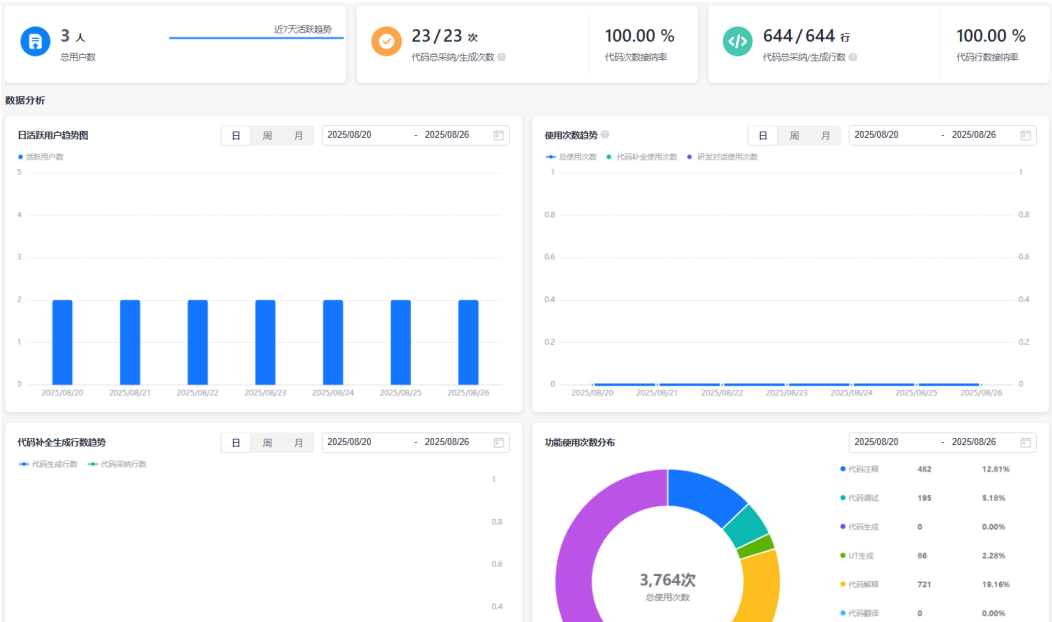


表 3-1 CodeArts 编码智能助手统计看板

看板内容	说明
总览数据	● 总用户数：企业授权的用户总数。 ● 代码总采纳次数：用户对代码生成功能的总采纳次数（包括单行生成和多行生成）。 ● 代码总生成次数：CodeArts 编码智能助手代码生成功能的总生成次数（包括单行生成和多行生成）。 ● 代码次数接纳率：代码总接纳次数 / 代码总生成次数。 ● 代码总采纳行数：用户对代码生成功能的总采纳行数（包括单行生成和多行生成）。 ● 代码总生成行数：CodeArts 编码智能助手代码生成功能的总生成行数（包括单行生成和多行生成）。 ● 代码行数接纳率：代码总采纳行数 / 代码总生成行数。

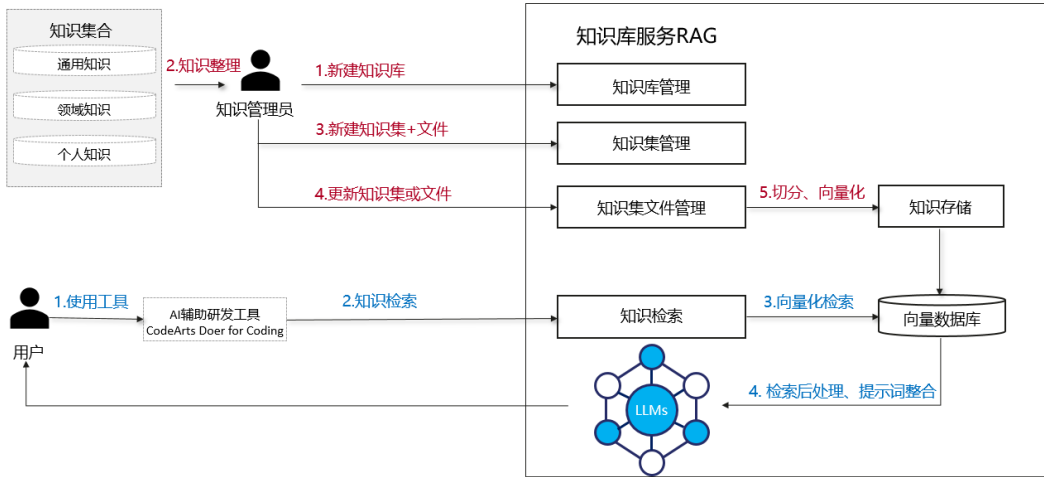
看板内容	说明
数据分析	● 日活跃用户数趋势图 用户可以自定义时间段，然后按照日、周、月粒度查看在此时间段内的活跃用户数及变化趋势。 说明 活跃用户数：单位时间内，用户使用过CodeArts 编码智能助手插件的总人数。 ● 使用次数趋势 用户可以自定义时间段，然后按照日、周、月粒度查看在此时间段内用户使用插件功能的总次数、使用代码补全功能的次数、研发对话（代码解释、代码翻译、代码注释、代码调试、UT生成、研发问答等）的总使用次数及趋势变化。 ● 代码补全生成行数趋势 用户可以自定义时间段，然后按照日、周、月粒度查看在此时间段内代码补全生成行数及变化趋势。 ● 功能使用次数分布 用户在所选时间段内，使用CodeArts 编码智能助手插件在不同应用场景下（代码解释、代码翻译、代码注释、代码调试、代码生成、UT生成、研发问答）的使用次数及占比。 ● 成员使用次数详情 成员在所选时间段内，使用CodeArts 编码智能助手插件在不同应用场景下（代码解释、代码翻译、代码注释、代码调试、代码生成、UT生成、研发问答等）的使用次数及总使用次数。 ● 成员代码生成使用详情 - 代码生成次数：成员在所选时间内生成代码的次数（包括单行生成和多行生成）。 - 代码采纳次数：成员在所选时间内采纳代码生成的次数（包括单行生成和多行生成）。 - 代码次数采纳率：代码采纳次数 / 代码生成次数。 - 代码生成行数：成员在所选时间内生成代码的行数（包括单行生成和多行生成）。 - 代码采纳行数：成员在所选时间内采纳代码生成的行数（包括单行生成和多行生成）。 - 代码行数采纳率：代码采纳行数 / 代码生成行数。

3.2 知识库中心

3.2.1 知识库概述

知识库是将租户上传的数据进行切片、模型向量化处理、索引过滤后入库形成的知识集合。在CodeArts 编码智能助手插件UT生成、代码续写和研发知识问答场景下，可以增强其准确性。

图 3-2 知识库整体架构

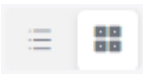


基本概念

- **知识库**：知识的集合，按照实际使用维度进行划分，通常基于知识库进行检索召回。接口和数据库都采用UTF-8字符集编码。
- **知识集**：知识库的二级目录，可按照知识来源划分。
- **文件**：知识载体，知识的导入到检索的端到端效果，需要应用正确的加载切片配置、合理的向量化模型、正确的索引过滤配置以及对应的检索策略。
- **切片**：加载切分后的文本块；向量化、存储、召回的单位。
- **操作日志**：记录知识集初始化、更新（文件增删）、删除任务的执行状态。
- **导入状态**：记录文件的状态，包含上传中、加载失败、入库中、入库成功、入库失败、删除中、删除失败七种状态。

3.2.2 新建知识库

步骤1 登录[知识库管理页面](#)，可查看当前租户下所有知识库。

- 知识库可根据类型和可见范围进行筛选。
 - 类型
 - 所有：展示当前用户可见的知识库。
 - 我创建的：展示当前用户创建的知识库。
 - 可见范围
 - 个人：展示当前用户创建的个人级知识库，仅当前用户可见。
 - 项目：展示对应项目关联的知识库，项目下所有成员可见。
 - 租户：展示租户级的知识库，租户下所有成员可见。
- 单击“新建知识库”旁的搜索框，可以根据知识库名称关键字搜索知识库。
- 单击右上角，可切换知识库显示视图。

步骤2 单击“新建知识库”，参考[表3-2](#)完成知识库配置，单击“确定”。

图 3-3 新建知识库

新建知识库

* 知识库名称

1-18字符，支持中文、英文字母、数字、下划线、中划线，不能以下划线开头

* 可见范围

☒ 个人可见

☐ 项目可见

☐ 租户可见

标签

点击选择或新建标签，最多可创建20个标签

描述

描述知识库包含的数据内容，详细的描述可以让AI及时、准确的访问知识库的内容

0/300

确定

取消

表 3-2 新建知识库

参数名称	参数说明
知识库名称	知识库的名称。支持中文、英文字母、数字、下划线、中划线，不能以下划线开头。长度为1-18个字符。
可见范围	支持按个人、项目、租户三个维度控制知识库访问权限。 - 个人可见：仅当前用户可见，拥有该知识库的所有权限。 - 项目可见：知识库关联的项目下所有成员可见，对知识库的操作权限由项目角色而定，详见管理项目级权限。 - 租户可见：租户下所有成员可见，对知识库的操作权限由成员租户级角色而定。
关联项目	“可见范围”选择“项目可见”时需选择知识库关联的项目，不可更改。
标签	标记知识库。最多可以创建20个标签。
描述	描述知识库包含的数据内容，详细的描述可以让AI及时准确地访问知识库的内容。

步骤3 知识库创建成功后会在页面上生成一张知识库卡片，创建者默认为知识库owner。

步骤4 （可选）将鼠标移至知识库卡片，可根据需要编辑、删除知识库。

----结束

3.2.3 新建知识集

- 步骤1 登录[知识库管理页面](#)，进入“知识库管理”页面。
- 步骤2 单击知识库卡片，进入知识库详情页。
- 步骤3 单击“新建知识集”，进入“新建知识集”页面，参考[表3-3](#)完成基本信息配置及文件上传。

图 3-4 新建知识集

基本信息

* 知识集名称

1-32字符，不能以下划线开头，仅支持中英文、数字、下划线和中划线

知识类型

☒ 文档 ☐ 结构化格式

主要包含文本内容，可以包括文章、报告、书籍等，如TXT、DOCX、ZIP格式文件

上传文件

数据来源

☒ 本地文件 ☐ 代码

* 文件

选择文件

支持TXT、DOCX、PDF、MD、CSV、XLS、XLSX格式文件，单个文件最大支持20MB，单个知识集配额50MB

表 3-3 新建知识集

参数名称	参数说明
知识集名称	知识集的名称。支持中英文、数字、下划线和中划线，不能以下划线开头。长度为1-32个字符。
知识类型	支持“文档”和“结构化格式”两种类型。 - 文档：主要包含文本内容，可以包括文章、报告、书籍等，如TXT、DOCX、ZIP格式文件。 - 结构化格式：仅支持JSON格式文件。
数据来源	包括“本地文件”和“代码”两种类型。
文件	“数据来源”选择“本地文件”时，请选择对应文件。 “知识类型”为“文档”时，支持TXT、DOCX、PDF、MD、CSV、XLS、XLSX格式文件，单个文件最大支持20MB，单个知识集配额为50MB。 “知识类型”为“结构化文档”时，支持JSON格式文件，单个文件最大支持20MB，单个知识集配额50MB
代码	“数据来源”选择“代码”时，请选择对应代码文件。支持JAVA、PYTHON、C、C++、VUE文件的ZIP格式，单个文件最大支持20MB，单个知识集配额为50MB。

步骤4 单击“下一步”，完成入库配置。

图 3-5 入库配置

入库配置

Embedding模型

 **m3e-base**
该模型是适用于中英文文本的通用型768维度的嵌入模型

 **bge-large-zh-v1.5**
该模型适用于检索、分类、聚类或语义匹配等任务，目前...

 **paraphrase-multilingual-MiniLM-L12-v2**
该模型适用于中英文文本，在文本相似度计算中表现优异...

 **VESO-v1.3**
该模型在代码数据向量化、代码搜索任务中表现突出

高级配置

恢复默认

切分器 ②

基于段落的切分器

分段预计长度 ②

1000

分段重叠长度 ②

0

子切分器 ①

基于字符的切分器

1. 选择Embedding模型，对上传的文档进行向量化，当前提供以下模型可供选择。

表 3-4 Embedding 模型

模型名称	说明
m3e-base	该模型是适用于中英文文本的通用型768维度的嵌入模型。
bge-large-zh-v1.5	该模型适用于检索、分类、聚类或语义匹配等任务，目前仅支持中文文本。
paraphrase-multilingual-MiniLM-L12-v2	该模型适用于中英文文本，在文本相似度计算中表现优异，模型维度364。
VESO-v1.3	该模型在代码数据向量化、代码搜索任务中表现突出。

2. 按需配置“高级配置”中的“切分器”、“分段预计长度”、“分段重叠长度”和“子切分器”。

表 3-5 高级配置

模型名称	说明
切分器	不同的切分器的切片方式和结果不同，在不同的语言场景下可以选择不同的切分器，当前支持以下切分器。 - 基于段落的切分器（默认使用）：按照两个连续的换行符（\n\n）分割或文本段落结构分割。 - 基于行的切分器：按照换行符（\n）分割。 - 基于句子的切分器（适用于英文文本）：将文本按照英文句子分割。 - 基于字符的切分器：逐个字符（含空白字符）分割。
分段预计长度	不应用于docx、md文件；分片长度的取值范围为50-1200，默认值为1000。
分段重叠长度	不应用于docx、md文件；当前分片与上一个分片重叠的文本长度，取值范围0到分片的长度，默认值为0。
子切分器	不应用于docx、md文件；切片长度超过最大长度时应用子切分器。当前支持以下切分器： - 基于字符的切分器（默认使用）：逐个字符（含空白字符）分割。 - 基于句子的切分器（适用于英文文本）：将文本按照英文句子分割。 - 基于单词的切分器（适用于英文文本）：将文本按照空白字符分割。

步骤5 单击“确定”，知识集创建完成，如图3-6所示，为创建好的知识集列表。

图 3-6 知识集列表



步骤6 （可选）在知识集列表，单击“操作”列的，可以删除知识集。

----结束

3.2.4 管理知识集文件

- 步骤1 登录[知识库管理页面](#)，进入“知识库管理”页面。
- 步骤2 单击知识库卡片，进入知识库详情页。
- 步骤3 单击知识集名称，进入知识集详情页，可以看到在新建知识集时已上传的文件。

知识集创建成功后，会自动进行初始化，在知识集详情页面，可以查看知识集文件的导入状态，详见表3-6。

表 3-6 文件导入状态

模型名称	说明
上传中	正在从数据源获取文档内容。
加载失败	文档解析切片失败，通常失败原因为分段清洗配置有误或是源文档损坏。
入库中	执行向量化和索引建立。
入库成功	成功入库。
入库失败	执行向量化和索引建立失败，通常失败原因为索引配置有误。
删除中	执行文档在向量库中的删除过程。
删除失败	执行删除操作失败。

步骤4 单击左上角“导入文件”，可增量上传知识集产文件，支持同时上传多个文件。

图 3-7 导入文件



步骤5 （可选）知识集文件导入后，还可以根据需要进行以下操作。

- 单击文件名称，可以查看文件详情。
- 单击知识集文件“操作”列的，可以删除已上传的知识集文件。
- 单击右上角“刷新”，可以刷新知识集文件列表。
- 单击右上角“重新导入”，可以重新导入当前知识集下所有文件。

----结束

3.2.5 查看知识库操作日志

知识集与文件初始化、更新以及删除都会记录相应的操作日志。

步骤1 登录[知识库管理页面](#)，进入“知识库管理”页面。

步骤2 单击右上角“操作日志”，可查看所有知识库下知识集和文件的操作记录。

图 3-8 操作日志样例

操作日志

序号	知识库名称	知识集名称	操作人	操作类型	状态	创建时间	状态更新时间
1	rag01	asset01	...	初始化	入库成功	2025/08/27...	2025/08/27...
2	rag01	asset01	...	删除	删除成功	2025/08/27...	2025/08/27...

20 条/页， 所有条目: 2 < 1 > 跳至 1 页

更多操作：

- 单击知识库名称，进入知识库详情页，单击右上角“操作日志”，可查看当前知识库下所有知识集和文件的操作记录。
- 单击知识集名称，进入知识集详情页，单击右上角“操作日志”，可查看当前知识集和文件的操作记录。

----结束

3.3 租户成员管理

成功申请CodeArts 智能助手后，需要将成员添加至租户成员清单，被添加的成员才能使用CodeArts 智能助手。

- 步骤1 使用管理员账号（拥有Tenant Administrator或Security Administrator权限）登录[租户成员管理页面](#)。
- 步骤2 单击左上角“从本企业导入成员”，在弹窗中根据实际需要选择需要导入的成员，并设置成员类型。

图 3-9 导入成员

从本企业导入成员

☒ 成员名称	企业用户	类型
☒ test001	c...	CodeArts 编码智能助手
☐ test002	d...	CodeArts 编码智能助手
☐ test003	...	CodeArts 编码智能助手
☐ test008	...	CodeArts 编码智能助手

10 条/页， 所有条目: 36 < 1 2 3 4 >

确定 取消

- 步骤3 单击“确定”，完成成员导入。
- 步骤4 （可选）单击“操作”列，可移除成员。
- 结束

3.4 项目成员管理

CodeArts 智能助手支持对项目成员进行管理（导入和删除）。

- 步骤1 使用租户账号登录[项目成员管理页面](#)。
- 步骤2 单击左上角“从本企业导入成员”，在弹窗中根据实际需要选择需要导入的成员，并设置成员角色。

图 3-10 导入成员



- 步骤3 单击“确定”，完成成员导入。
- 步骤4 （可选）单击“操作”列，可移除成员。
- 结束

4 使用 CodeArts 编码智能助手插件

4.1 安装并登录 CodeArts 编码智能助手插件

4.1.1 在 JetBrains 系列 IDE 安装 CodeArts 编码智能助手插件

JetBrains平台支持多种集成开发环境（IDE），包括IntelliJ IDEA、Pycharm等。本节以IntelliJ IDEA为例介绍JetBrains平台CodeArts 编码智能助手插件的安装流程，其他JetBrains系列的IDE请参考IntelliJ IDEA。

准备工作

已下载并安装IntelliJ IDEA 2021.3或更高版本。

通过 IntelliJ IDEA 插件市场安装 CodeArts 编码智能助手插件

- 步骤1** 打开IntelliJ IDEA，单击左上角菜单栏“File > Settings...”，进入IDE设置界面。
- 步骤2** 在设置界面左侧菜单选择“Plugins”，单击“Marketplace”，在搜索框中输入“CodeArts Doer for Coding”。
- 步骤3** 单击“Install”，IntelliJ IDEA将下载插件并自动安装。
- 步骤4** 安装完成后，单击“Restart IDE”或手动重启IDE使插件安装生效。

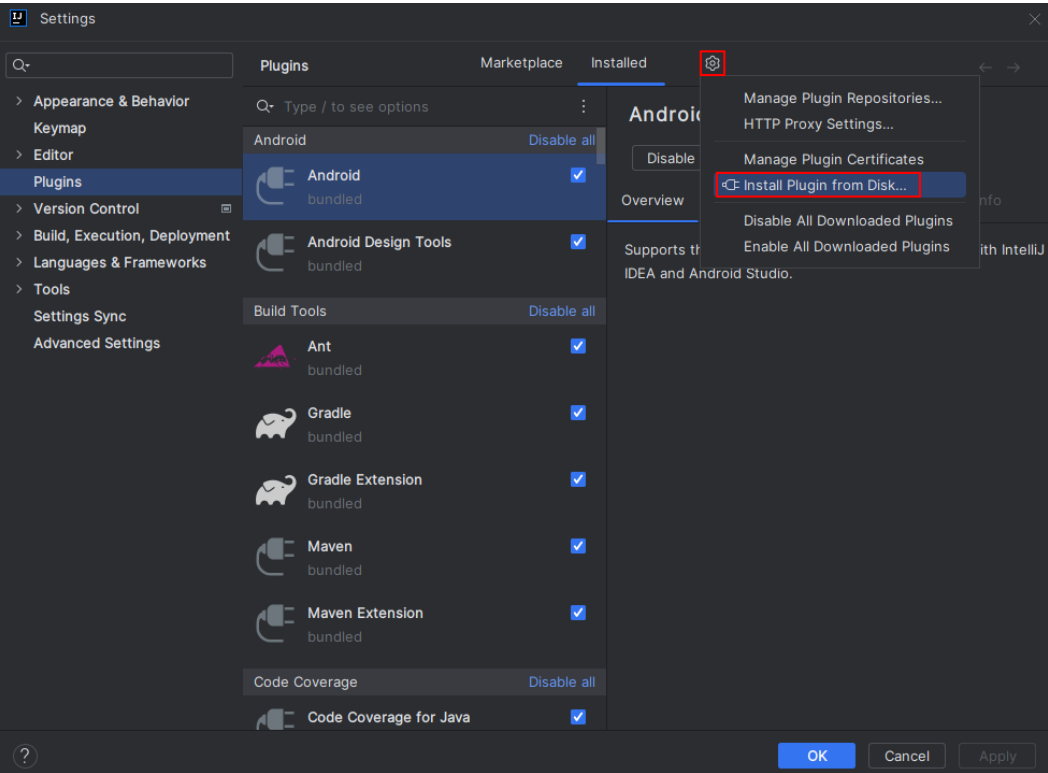
重启后，即可在IntelliJ IDEA侧边栏看到CodeArts 编码智能助手插件图标.

----结束

通过本地文件方式安装 CodeArts 编码智能助手插件

- 步骤1** 进入JetBrains插件市场[CodeArts 编码智能助手插件页面](#)。
- 步骤2** 切换至“Versions”标签页，选择最新版本，单击“Download”，下载离线安装包。
- 步骤3** 打开IntelliJ IDEA，单击左上角菜单栏“File > Settings...”，进入IDE设置界面。
- 步骤4** 在设置界面左侧菜单选择“Plugins”，单击齿轮图标，选择“Install Plugin from Disk...”。

图 4-1 选择从磁盘安装



- 步骤5** 选择已下载CodeArts 编码智能助手安装包，单击“OK”，IntelliJ IDEA将会检查依赖，并完成安装。
- 步骤6** 安装完成后，单击“Restart IDE”或手动重启IDE使插件安装生效。

重启后，即可在IntelliJ IDEA侧边栏看到CodeArts 编码智能助手插件图标.

----结束

登录 CodeArts 编码智能助手

可通过以下三种方式录CodeArts 编码智能助手插件。

- 华为IAM账号登录
输入“租户名”、“用户名”、“密码”登录，账号信息请参考[创建用户并授权使用CodeArts控制台](#)。
- 第三方账号登录
即企业联邦用户登录，账号信息请参考[企业联邦用户登录](#)。
- 华为AKSK登录
访问密钥认证方式进行认证鉴权，即使用AK/SK加密的方法来验证请求发送者身份。
 - AK (Access Key ID)：访问密钥ID，是与私有访问密钥关联的唯一标识符。访问密钥ID和私有访问密钥结合使用，对请求进行加密签名。
 - SK (Secret Access Key)：与访问密钥ID结合使用的密钥，对请求进行加密签名，可标识发送方，并防止请求被修改。访问密钥获取方式请参考[新增访问密钥](#)。

4.1.2 在 VSCode 安装 CodeArts 编码智能助手插件

前提条件

已下载并安装[Visual Studio Code](#)最新版本。

通过 VSCode 插件市场安装 CodeArts 编码智能助手插件

- 步骤1** 打开Visual Studio Code，单击左上角菜单栏“File > Preferences > Extensions”，弹出IDE的插件市场的插件列表。
- 步骤2** 在搜索框中输入“CodeArts Doer for Coding”。
- 步骤3** 单击“Install”，Visual Studio Code将下载插件并自动安装。
- 步骤4** 安装完成后，单击“重新加载窗口”，使插件生效。

----结束

登录 CodeArts 编码智能助手

可通过以下三种方式登录CodeArts 编码智能助手插件。

- 华为IAM账号登录
输入“租户名”、“用户名”、“密码”登录，账号信息请参考[创建用户并授权使用CodeArts控制台](#)。
- 第三方账号登录
即企业联邦用户登录，账号信息请参考[企业联邦用户登录](#)。
- 华为AKSK登录
访问密钥认证方式进行认证鉴权，即使用AK/SK加密的方法来验证请求发送者身份。
 - AK (Access Key ID)：访问密钥ID，是与私有访问密钥关联的唯一标识符。访问密钥ID和私有访问密钥结合使用，对请求进行加密签名。
 - SK (Secret Access Key)：与访问密钥ID结合使用的密钥，对请求进行加密签名，可标识发送方，并防止请求被修改。访问密钥获取方式请参考[新增访问密钥](#)。

4.1.3 在 CodeArts IDE 安装 CodeArts 编码智能助手

前提条件

下载并安装[CodeArts IDE](#)最新版本。

通过 CodeArts IDE 插件市场安装 CodeArts 编码智能助手

- 步骤1** 打开CodeArts IDE，单击左上角菜单栏““File” > “Preferences” > “Extensions””，弹出IDE的插件市场的插件列表。
- 步骤2** 在搜索框中输入“CodeArts Doer for Coding”。
- 步骤3** 单击“Install”，Visual Studio Code将下载插件并自动安装。

步骤4 安装完成后，单击“重新加载窗口”，使插件生效。

----结束

通过本地文件方式安装 CodeArts 编码智能助手

步骤1 进入CodeArts IDE插件市场[CodeArts 编码智能助手插件页面](#)，单击右侧“Download Extension”下载插件。

步骤2 打开CodeArts IDE，单击右上角“扩展”打开扩展管理。

步骤3 单击  图标，选择已下载的CodeArts 编码智能助手插件，单击“安装”。

步骤4 CodeArts IDE弹出“非受信任扩展安装提示”，单击“去扩展市场查找”或“取消”跳出安装，单击“继续安装”进入下一步。

步骤5 “CodeArts IDE”完成安装后弹出安装成功提示，同时弹出CodeArts 编码智能助手免责声明。

根据您的意愿是否“同意”免责声明，同意即可使用插件，关闭将无法使用插件。

----结束

登录 CodeArts 编码智能助手

可通过以下三种方式登录CodeArts 编码智能助手插件。

- 华为IAM账号登录
输入“租户名”、“用户名”、“密码”登录，账号信息请参考[创建用户并授权使用CodeArts控制台](#)。
- 第三方账号登录
即企业联邦用户登录，账号信息请参考[企业联邦用户登录](#)。
- 华为AKSK登录
访问密钥认证方式进行认证鉴权，即使用AK/SK加密的方法来验证请求发送者身份。
 - AK (Access Key ID)：访问密钥ID，是与私有访问密钥关联的唯一标识符。访问密钥ID和私有访问密钥结合使用，对请求进行加密签名。
 - SK (Secret Access Key)：与访问密钥ID结合使用的密钥，对请求进行加密签名，可标识发送方，并防止请求被修改。访问密钥获取方式请参考[新增访问密钥](#)。

4.2 使用 CodeArts 编码智能助手生成代码

4.2.1 通过问答描述生成代码

CodeArts 编码智能助手支持在研发对话窗口通过需求描述、编程语言注释或自然语言描述生成代码。

步骤1 [安装并登录CodeArts 编码智能助手插件](#)。

步骤2 在研发问答窗口输入问题描述，如：使用Java编写一个冒泡排序方法，不包含main方法。

图 4-2 输入问题



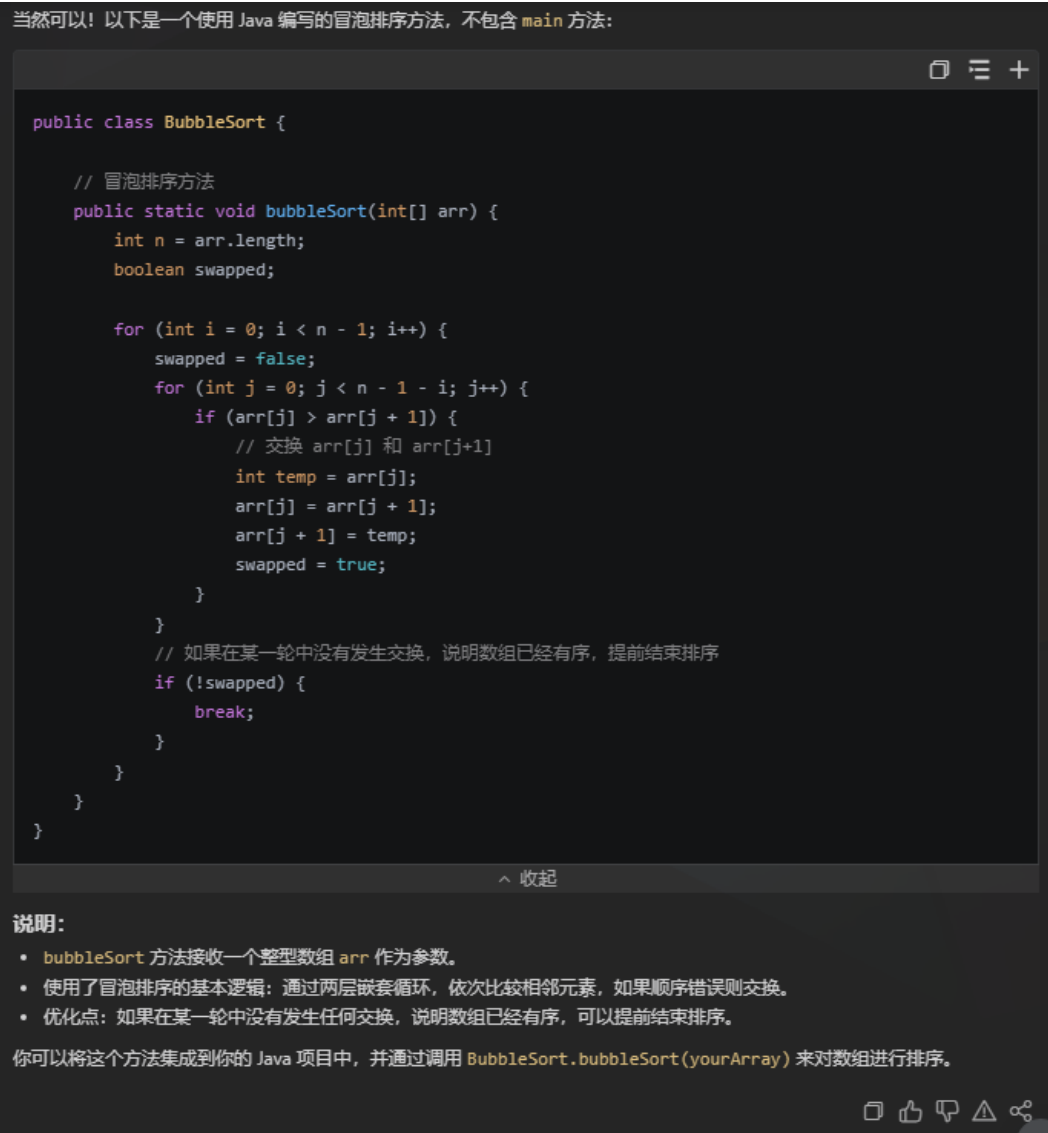
步骤3 单击  或使用“Enter”快捷键发送。

说明

插件在回答过程中输入框禁止发送问题。

步骤4 等待CodeArts 编码智能助手给出回答，回答内容包含描述及代码块，代码块提供复制、插入、另存为文件功能。

图 4-3 回答



----结束

相关操作

表 4-1 相关操作

操作项	说明
复制代码	单击回答框右上角  图标, 可将代码复制至剪贴板。
插入代码	单击回答框右上角  图标, 可将代码插入至当前打开文件光标所在位置。
保存代码	单击回答框右上角  图标, 可新建文件将代码保存为文件。
重新回答	如对当前回答不满意, 可单击回答框右下角  图标, 重新生成答案。

操作项	说明
复制回答	单击回答框右下角  图标，可复制回答内容。
评价回答	单击回答框右下角  或  图标，可对回答给出评价。
投诉举报	单击回答框右下角  ，可对回答结果进行投诉举报。
分享	单击回答框右下角  ，可将回答结果导出为Markdown文件或生成为图片。
新建对话	单击研发问答窗口右上角  图标，可打开新的问答窗口，开启新的问答。
对话历史	单击研发问答窗口右上角图标，可查看历史对话。 - 单击历史对话，可查看该对话的问答结果，并在历史对话基础上继续提问。 - 将鼠标移到历史对话，可编辑、分享、删除历史对话。 - 单击“查看更多”，可查看最近一个月的历史对话。 - 单击搜索框，可输入标题搜索历史对话。 - 勾选历史对话，可删除对应对话。
MCP服务器	单击研发问答窗口右上角图标，可配置MCP服务器。 说明 仅JetBrains系列IDE支持该功能。

4.2.2 通过编辑器上下文生成代码

CodeArts 编码智能助手支持在编辑器中根据代码上下文触发生成代码。可以直接读取上下文触发代码生成，也可以编写注释完善上下文再生成代码，这样生成的结果更准确，更能满足业务需求。

表 4-2 生成代码相关操作

快捷键	说明
停顿	编写代码过程中，自动识别上下文生成代码。
回车	回车后，自动识别上下文生成代码。
Alt+X	单行生成。
Alt+C	多行生成。
Tab	接受生成的代码。
Ctrl+Right	逐词接纳生成的代码。
Escape	取消生成的代码。

快捷键	说明
Alt+R	重新生成代码。
Alt+[	重新生成代码时，切换至上一个生成结果。
Alt+]	重新生成代码时，切换至下一个生成结果。

示例：通过注释生成代码

- 步骤1 安装并登录CodeArts 编码智能助手插件。
- 步骤2 在编辑器中编写符合编程语言规范的注释。

```
/**
 * 冒泡排序
 * @param array int数组
 */
```
- 步骤3 将编辑器中光标移动至注释内容最后，按下快捷键“Alt+C”。
- 步骤4 CodeArts 编码智能助手将会在编辑器中生成一个冒泡排序方法至注释下方。
- 步骤5 开发者可以选择接受或拒绝CodeArts 编码智能助手生成的代码。如果首次生成的代码不完整，可以先按Tab键接受生成的代码，再按快捷键“Alt+C”继续生成代码，直到生成完整代码片段。

图 4-4 由注释生成的代码片段

```
public class BubbleSort {
    /**
     * 冒泡排序
     * @param array int数组
     */
    no usages
    public static void bubbleSort(int[] array) {
        for (int i = 0; i < array.length - 1; i++) {
            for (int j = 0; j < array.length - 1 - i; j++) {
                if (array[j] > array[j + 1]) {
                    int temp = array[j];
                    array[j] = array[j + 1];
                    array[j + 1] = temp;
                }
            }
        }
    }
}
```

----结束

4.3 使用 CodeArts 编码智能助手解释代码

步骤1 安装并登录CodeArts 编码智能助手插件。

步骤2 在编辑器选中待解释的代码。

步骤3 发送选中代码，可通过以下两种方式。

- 方式一
单击右键，选择“CodeArts 编码智能助手 > 代码解释”。
- 方式二
 - a. （可选）单击右键，选择“CodeArts 编码智能助手：将代码添加到聊天窗”或使用“Ctrl+Shift+Q”快捷键，将代码添加到聊天窗。
 - b. 在聊天窗输入“/”，选择“/explain”，单击或使用“Enter”快捷键发送。

说明

插件在回答过程中输入框禁止发送问题。

图 4-5 待解释的代码

```
public class BubbleSort {  
    /**  
     * 冒泡排序  
     * @param array int数组  
     */  
    no usages  
    public static void bubbleSort(int[] array) {  
        for (int i = 0; i < array.length - 1; i++) {  
            for (int j = 0; j < array.length - 1 - i; j++) {  
                if (array[j] > array[j + 1]) {  
                    int temp = array[j];  
                    array[j] = array[j + 1];  
                    array[j + 1] = temp;  
                }  
            }  
        }  
    }  
}
```

步骤4 CodeArts 编码智能助手将会在研发问答窗口回复代码的解释内容。

图 4-6 代码片段解释

代码功能与目的

这段代码实现了一个经典的排序算法——冒泡排序。其主要目的是将输入的整数数组按照升序排列。冒泡排序是一种简单但效率较低的排序算法，适用于小规模数据的排序或作为教学示例。

使用场景

1. 教学演示：冒泡排序因其简单性，常用于算法教学，帮助初学者理解排序的基本思想。

2. 小规模数据排序：对于数据量较小的场景（如数组长度在100以内），冒泡排序的性能尚可接受。

3. 算法对比：在需要与其他高效排序算法（如快速排序、归并排序）进行性能对比时，冒泡排序可作为基准算法。

代码逻辑分条解析

1. 外层循环控制遍历轮数

- 循环变量 `i` 从 0 到 `array.length - 2`，表示需要进行 `n-1` 轮遍历（`n` 为数组长度）。
- 每轮遍历后，最大的元素会被“冒泡”到数组末尾。

2. 内层循环执行相邻元素比较与交换

- 循环变量 `j` 从 0 到 `array.length - 2 - i`，确保每轮遍历的范围逐渐缩小（已排序部分无需重复检查）。
- 比较相邻元素 `array[j]` 和 `array[j + 1]`，若前者大于后者，则交换它们的位置。

3. 交换操作

- 通过临时变量 `temp` 保存当前元素值，完成元素交换：

```
int temp = array[j];
array[j] = array[j + 1];
array[j + 1] = temp;
```

4. 优化点

- 内层循环的终止条件为 `array.length - 1 - i`，避免重复检查已排序的部分（每轮遍历后，末尾 `i` 个元素已有序）。

总结

该代码通过两层嵌套循环实现冒泡排序，时间复杂度为 $O(n^2)$ ，适合教学和小规模数据排序。实际应用中，建议使用更高效的排序算法（如快速排序或归并排序）。

----结束

相关操作

表 4-3 相关操作

操作项	说明
复制代码	单击回答框右上角  图标，可将代码复制至剪贴板。
插入代码	单击回答框右上角  图标，可将代码插入至当前打开文件光标所在位置。
保存代码	单击回答框右上角  图标，可新建文件将代码保存为文件。
重新回答	如对当前回答不满意，可单击回答框右下角  图标，重新生成答案。

文档版本 01 (2025-09-15)

版权所有 © 华为云计算技术有限公司

27

操作项	说明
复制回答	单击回答框右下角  图标，可复制回答内容。
评价回答	单击回答框右下角  或  图标，可对回答给出评价。
投诉举报	单击回答框右下角  图标，可对回答结果进行投诉举报。
分享	单击回答框右下角  图标，可将回答结果导出为Markdown文件或生成为图片。
新建对话	单击研发问答窗口右上角  图标，可打开新的问答窗口，开启新的问答。
对话历史	单击研发问答窗口右上角图标，可查看历史对话。 - 单击历史对话，可查看该对话的问答结果，并在历史对话基础上继续提问。 - 将鼠标移到历史对话，可编辑、分享、删除历史对话。 - 单击“查看更多”，可查看最近一个月的历史对话。 - 单击搜索框，可输入标题搜索历史对话。 - 勾选历史对话，可删除对应对话。
MCP服务器	单击研发问答窗口右上角图标，可配置MCP服务器。 说明 仅JetBrains系列IDE支持该功能。

4.4 使用 CodeArts 编码智能助手注释代码

- 步骤1 安装并登录CodeArts 编码智能助手插件。
- 步骤2 在编辑器选中待注释的代码。
- 步骤3 发送选中代码，可通过以下两种方式。
 - 方式一
单击右键，选择“CodeArts 编码智能助手 > 代码注释”。
 - 方式二
 - （可选）单击右键，选择“CodeArts 编码智能助手：将代码添加到聊天窗”或使用“Ctrl+Shift+Q”快捷键，将代码添加到聊天窗。
 - 在聊天窗输入“/”，选择“/comment”，单击或使用“Enter”快捷键发送。

 说明

插件在回答过程中输入框禁止发送问题。

图 4-7 待注释的代码

```
public class BubbleSort {  
    /**  
     * 冒泡排序  
     * @param array int数组  
     */  
    no usages  
    public static void bubbleSort(int[] array) {  
        for (int i = 0; i < array.length - 1; i++) {  
            for (int j = 0; j < array.length - 1 - i; j++) {  
                if (array[j] > array[j + 1]) {  
                    int temp = array[j];  
                    array[j] = array[j + 1];  
                    array[j + 1] = temp;  
                }  
            }  
        }  
    }  
}
```

步骤4 CodeArts 编码智能助手将会在研发问答窗口中给出注释后的代码。

图 4-8 注释后代码

```
public class BubbleSort {
    /**
     * 冒泡排序
     * @param array int数组
     */
    public static void bubbleSort(int[] array) {
        // 外层循环控制排序的轮数，每轮确定一个最大值
        for (int i = 0; i < array.length - 1; i++) {
            // 内层循环负责每轮的比较和交换操作
            for (int j = 0; j < array.length - 1 - i; j++) {
                // 如果前一个元素大于后一个元素，则交换它们的位置
                if (array[j] > array[j + 1]) {
                    int temp = array[j]; // 临时变量存储当前元素的值
                    array[j] = array[j + 1]; // 将后一个元素的值赋给当前元素
                    array[j + 1] = temp; // 将临时变量的值赋给后一个元素
                }
            }
        }
    }
}
```

----结束

相关操作

表 4-4 相关操作

操作项	说明
复制代码	单击回答框右上角  图标，可将代码复制至剪贴板。
插入代码	单击回答框右上角  图标，可将代码插入至当前打开文件光标所在位置。
保存代码	单击回答框右上角  图标，可新建文件将代码保存为文件。
重新回答	如对当前回答不满意，可单击回答框右下角  图标，重新生成答案。
复制回答	单击回答框右下角  图标，可复制回答内容。
评价回答	单击回答框右下角  或  图标，可对回答给出评价。
投诉举报	单击回答框右下角  图标，可对回答结果进行投诉举报。
分享	单击回答框右下角  图标，可将回答结果导出为Markdown文件或生成图片。

操作项	说明
新建对话	单击研发问答窗口右上角  图标，可打开新的问答窗口，开启新的问答。
对话历史	单击研发问答窗口右上角图标，可查看历史对话。 - 单击历史对话，可查看该对话的问答结果，并在历史对话基础上继续提问。 - 将鼠标移到历史对话，可编辑、分享、删除历史对话。 - 单击“查看更多”，可查看最近一个月的历史对话。 - 单击搜索框，可输入标题搜索历史对话。 - 勾选历史对话，可删除对应对话。
MCP服务器	单击研发问答窗口右上角图标，可配置MCP服务器。 说明 仅JetBrains系列IDE支持该功能。

4.5 使用 CodeArts 编码智能助手调试代码

- 步骤1 安装并登录CodeArts 编码智能助手插件。
- 步骤2 在编辑器选中待调试的代码。
- 步骤3 发送选中代码，可通过以下两种方式。
 - 方式一
单击右键，选择“CodeArts 编码智能助手 > 代码调试（Beta）”。
 - 方式二
 - （可选）单击右键，选择“CodeArts 编码智能助手：将代码添加到聊天窗”或使用“Ctrl+Shift+Q”快捷键，将代码添加到聊天窗。
 - 在聊天窗输入“/”，选择“/debug”，单击或使用“Enter”快捷键发送。

 **说明**

插件在回答过程中输入框禁止发送问题。

图 4-9 待调试的代码

```
public class BubbleSort {  
    /**  
     * 冒泡排序  
     * @param array int数组  
     */  
    2 usages  
    public static void bubbleSort(int[] array) {  
        for (int i = 0; i < array.length - 1; i++) {  
            for (int j = 0; j < array.length - 1 - i; j++) {  
                if (array[j] > array[j + 1]) {  
                    int temp = array[j];  
                    array[j] = array[j + 1];  
                    array[j + 1] = temp;  
                }  
            }  
        }  
    }  
}
```

步骤4 CodeArts 编码智能助手将在研发问答窗口生成一个输入错误日志的输入框。开发者在输入框中输入错误日志或异常信息，单击“生成答案”。

图 4-10 代码调试过程

BubbleSort.java: L3-19

我将根据输入的错误代码和错误日志提供调试运行错误的解决方案。

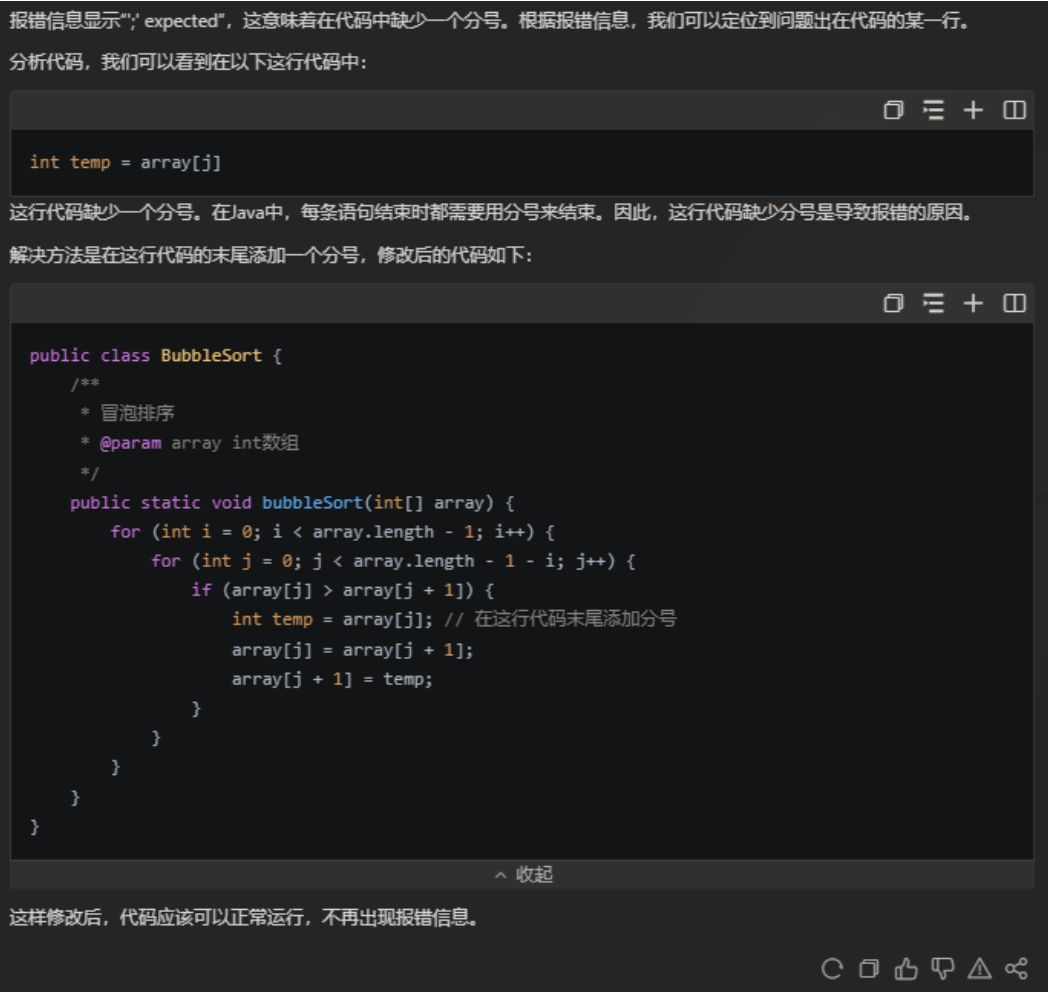
请输入错误日志:

';' expected

生成答案

步骤5 CodeArts 编码智能助手会在研发问答窗口给出异常出现原因及修改后代码片段。

图 4-11 代码调试结果



----结束

相关操作

表 4-5 相关操作

操作项	说明
复制代码	单击回答框右上角  图标，可将代码复制至剪贴板。
插入代码	单击回答框右上角  图标，可将代码插入至当前打开文件光标所在位置。
保存代码	单击回答框右上角  图标，可新建文件将代码保存为文件。
重新回答	如对当前回答不满意，可单击回答框右下角  图标，重新生成答案。
复制回答	单击回答框右下角  图标，可复制回答内容。
评价回答	单击回答框右下角  或  图标，可对回答给出评价。

操作项	说明
投诉举报	单击回答框右下角  ，可对回答结果进行投诉举报。
分享	单击回答框右下角  ，可将回答结果导出为Markdown文件或生成为图片。
新建对话	单击研发问答窗口右上角  图标，可打开新的问答窗口，开启新的问答。
对话历史	单击研发问答窗口右上角图标，可查看历史对话。 - 单击历史对话，可查看该对话的问答结果，并在历史对话基础上继续提问。 - 将鼠标移到历史对话，可编辑、分享、删除历史对话。 - 单击“查看更多”，可查看最近一个月的历史对话。 - 单击搜索框，可输入标题搜索历史对话。 - 勾选历史对话，可删除对应对话。
MCP服务器	单击研发问答窗口右上角图标，可配置MCP服务器。 说明 仅JetBrains系列IDE支持该功能。

4.6 使用 CodeArts 编码智能助手翻译代码

CodeArts 编码智能助手支持将代码翻译为其他语言，本节以CodeArts 编码智能助手生成的JavaScript冒泡排序代码作为示例进行介绍。

- 步骤1 安装并登录CodeArts 编码智能助手插件。
- 步骤2 在编辑器选中待翻译的代码。
- 步骤3 单击右键，选择“CodeArts 编码智能助手：将代码添加到聊天窗”或使用快捷键“Ctrl+Shift+Q”或将代码添加到聊天窗。

```
function bubbleSort(array) {
  const len = array.length;
  // 外层循环控制排序的轮数
  for (let i = 0; i < len - 1; i++) {
    // 内层循环负责每轮的比较和交换操作
    for (let j = 0; j < len - 1 - i; j++) {
      // 如果前一个元素大于后一个元素，则交换它们的位置
      if (array[j] > array[j + 1]) {
        // 交换两个元素的值
        let temp = array[j];
        array[j] = array[j + 1];
        array[j + 1] = temp;
      }
    }
  }
  return array;
}
```

- 步骤4 在聊天窗输入翻译代码要求，如：将这段代码翻译为Java。

- 步骤5 单击或使用“Enter”快捷键发送。

📖 说明

插件在回答过程中输入框禁止发送问题。

步骤6 CodeArts 编码智能助手将会在研发问答窗口中输出翻译后的代码内容。

图 4-12 代码翻译结果



----结束

相关操作

表 4-6 相关操作

操作项	说明
复制代码	单击回答框右上角  图标，可将代码复制至剪贴板。

操作项	说明
插入代码	单击回答框右上角  图标，可将代码插入至当前打开文件光标所在位置。
保存代码	单击回答框右上角  图标，可新建文件将代码保存为文件。
重新回答	如对当前回答不满意，可单击回答框右下角  图标，重新生成答案。
复制回答	单击回答框右下角  图标，可复制回答内容。
评价回答	单击回答框右下角  或  图标，可对回答给出评价。
投诉举报	单击回答框右下角  图标，可对回答结果进行投诉举报。
分享	单击回答框右下角  图标，可将回答结果导出为Markdown文件或生成为图片。
新建对话	单击研发问答窗口右上角  图标，可打开新的问答窗口，开启新的问答。
对话历史	单击研发问答窗口右上角图标，可查看历史对话。 - 单击历史对话，可查看该对话的问答结果，并在历史对话基础上继续提问。 - 将鼠标移到历史对话，可编辑、分享、删除历史对话。 - 单击“查看更多”，可查看最近一个月的历史对话。 - 单击搜索框，可输入标题搜索历史对话。 - 勾选历史对话，可删除对应对话。
MCP服务器	单击研发问答窗口右上角图标，可配置MCP服务器。 说明 仅JetBrains系列IDE支持该功能。

4.7 使用 CodeArts 编码智能助手检查代码

CodeArts 编码智能助手支持检查代码是否存在缺陷或常见编码问题。本节以修改 CodeArts 编码智能助手生成的Java冒泡排序代码（将第二循环进入条件修改为“j<array.length+i-1”）作为示例进行介绍。

步骤1 安装并登录CodeArts 编码智能助手插件。

步骤2 在编辑器选中修改后的代码。

```
public static void bubbleSort(int[] array) {
    for (int i = 0; i < array.length - 1; i++) {
        for (int j = 0; j < array.length + 1 - i; j++) {
            if (array[j] > array[j + 1]) {
                int temp = array[j];
                array[j] = array[j + 1];
                array[j + 1] = temp;
            }
        }
    }
}
```

```
}  
}  
}
```

步骤3 单击右键，选择“CodeArts 编码智能助手：将代码添加到聊天窗”或使用快捷键“Ctrl+Shift+Q”将代码添加到聊天窗。

步骤4 在聊天窗输入检查代码要求，如：检查这段代码是否存在缺陷。

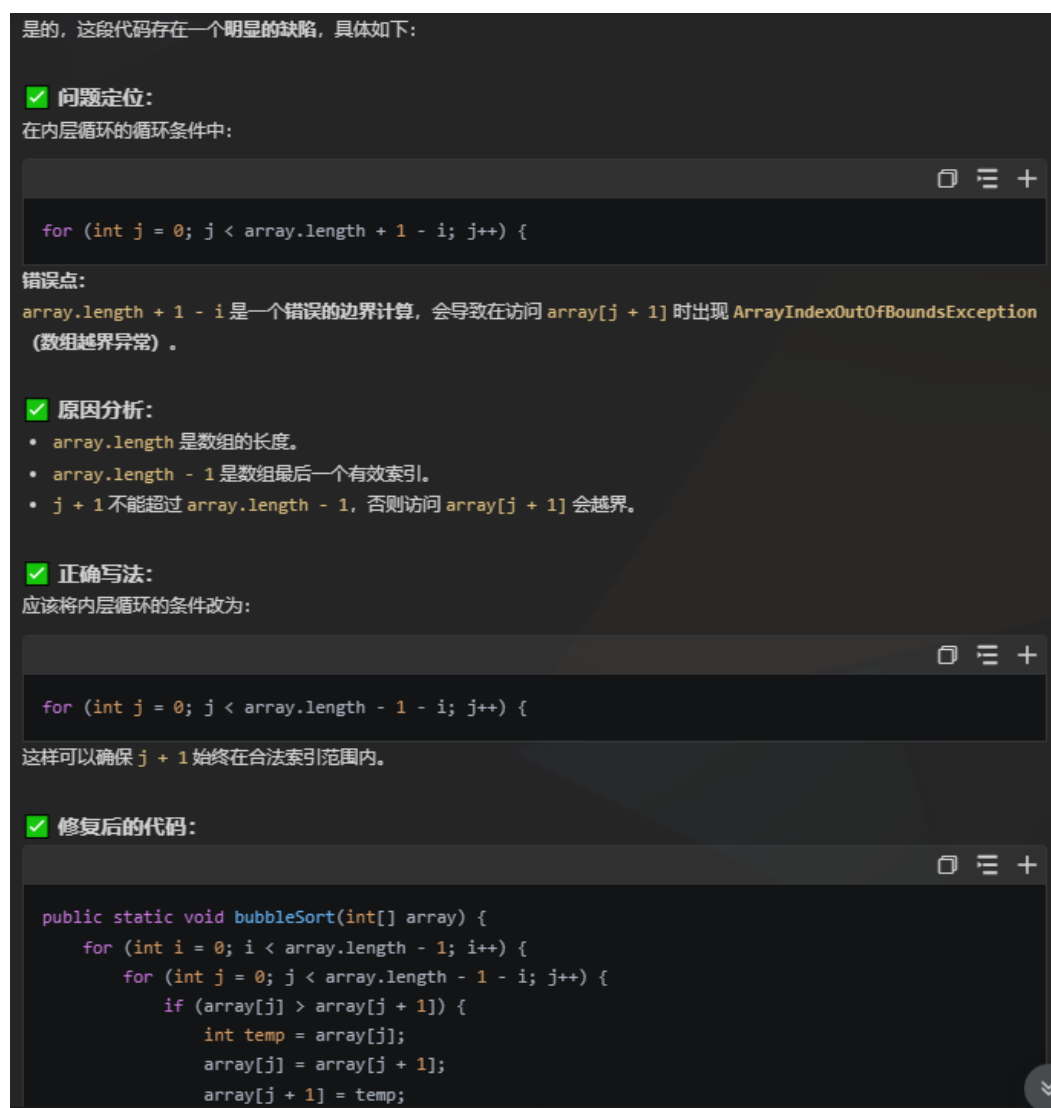
步骤5 单击或使用“Enter”快捷键发送。

说明

插件在回答过程中输入框禁止发送问题。

步骤6 CodeArts 编码智能助手将会在研发问答窗口中输出对这段代码的检查结果。

图 4-13 代码检查结果



----结束

相关操作

表 4-7 相关操作

操作项	说明
复制代码	单击回答框右上角  图标，可将代码复制至剪贴板。
插入代码	单击回答框右上角  图标，可将代码插入至当前打开文件光标所在位置。
保存代码	单击回答框右上角  图标，可新建文件将代码保存为文件。
重新回答	如对当前回答不满意，可单击回答框右下角  图标，重新生成答案。
复制回答	单击回答框右下角  图标，可复制回答内容。
评价回答	单击回答框右下角  或  图标，可对回答给出评价。
投诉举报	单击回答框右下角  图标，可对回答结果进行投诉举报。
分享	单击回答框右下角  图标，可将回答结果导出为Markdown文件或生成为图片。
新建对话	单击研发问答窗口右上角  图标，可打开新的问答窗口，开启新的问答。
对话历史	单击研发问答窗口右上角图标，可查看历史对话。 - 单击历史对话，可查看该对话的问答结果，并在历史对话基础上继续提问。 - 将鼠标移到历史对话，可编辑、分享、删除历史对话。 - 单击“查看更多”，可查看最近一个月的历史对话。 - 单击搜索框，可输入标题搜索历史对话。 - 勾选历史对话，可删除对应对话。
MCP服务器	单击研发问答窗口右上角图标，可配置MCP服务器。 说明 仅JetBrains系列IDE支持该功能。

4.8 使用 CodeArts 编码智能助手优化代码

4.8.1 优化选中代码

- 步骤1 安装并登录CodeArts 编码智能助手插件。
- 步骤2 在编辑器选中代码。

步骤3 调试选中代码，可通过以下两种方式。

- 方式一
单击右键，选择“CodeArts 编码智能助手 > 代码优化（Beta）”。
- 方式二
 - a. （可选）单击右键，选择“CodeArts 编码智能助手：将代码添加到聊天窗”或使用“Ctrl+Shift+Q”快捷键，将代码添加到聊天窗。
 - b. 在聊天窗输入“/”，选择“/optimize”，单击或使用“Enter”快捷键发送。

说明

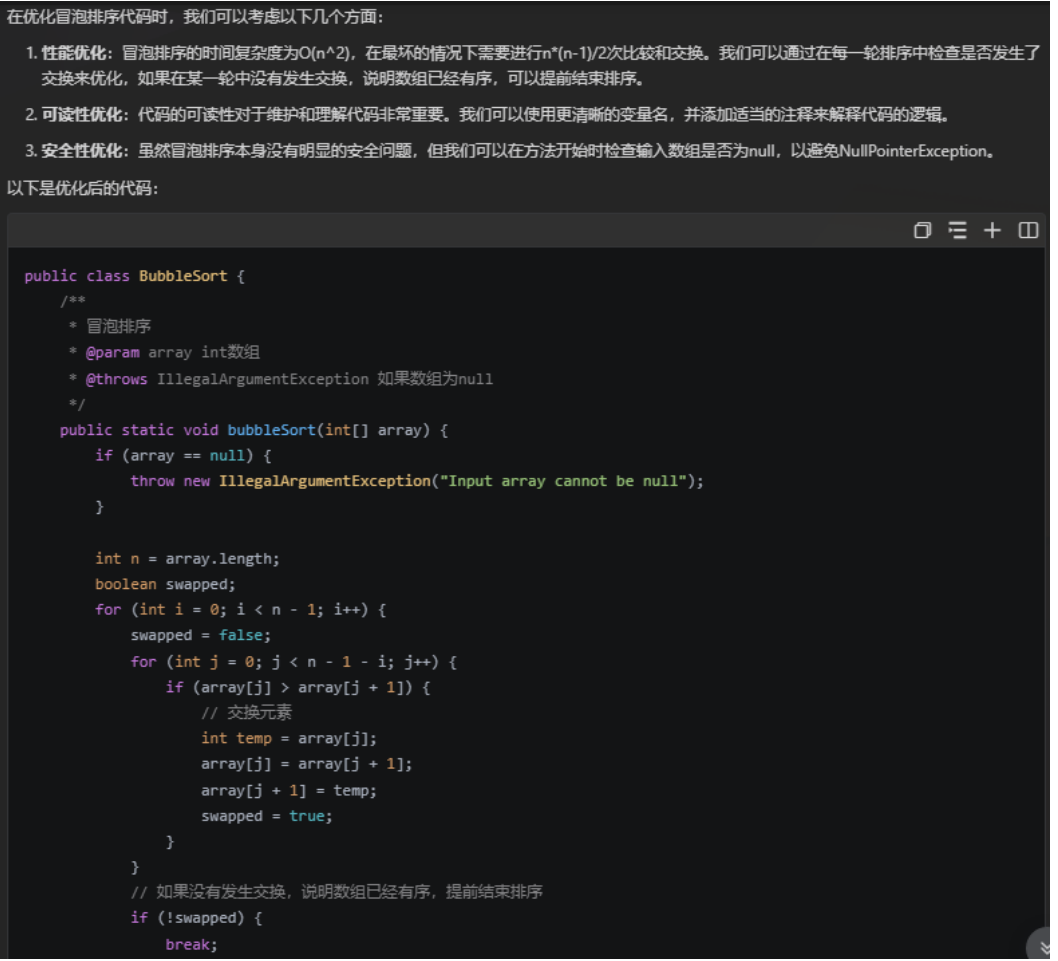
插件在回答过程中输入框禁止发送问题。

图 4-14 待优化的代码

```
public class BubbleSort {  
    /**  
     * 冒泡排序  
     * @param array int 数组  
     */  
    2 usages  
    public static void bubbleSort(int[] array) {  
        for (int i = 0; i < array.length - 1; i++) {  
            for (int j = 0; j < array.length - 1 - i; j++) {  
                if (array[j] > array[j + 1]) {  
                    int temp = array[j];  
                    array[j] = array[j + 1];  
                    array[j + 1] = temp;  
                }  
            }  
        }  
    }  
}
```

步骤4 CodeArts 编码智能助手将在研发问答窗口回复代码优化后的内容。

图 4-15 优化结果



----结束

相关操作

表 4-8 相关操作

操作项	说明
复制代码	单击回答框右上角  图标，可将代码复制至剪贴板。
插入代码	单击回答框右上角  图标，可将代码插入至当前打开文件光标所在位置。
保存代码	单击回答框右上角  图标，可新建文件将代码保存为文件。
重新回答	如对当前回答不满意，可单击回答框右下角  图标，重新生成答案。
复制回答	单击回答框右下角  图标，可复制回答内容。
评价回答	单击回答框右下角  或  图标，可对回答给出评价。

操作项	说明
投诉举报	单击回答框右下角  ，可对回答结果进行投诉举报。
分享	单击回答框右下角  ，可将回答结果导出为Markdown文件或生成为图片。
新建对话	单击研发问答窗口右上角  图标，可打开新的问答窗口，开启新的问答。
对话历史	单击研发问答窗口右上角图标，可查看历史对话。 - 单击历史对话，可查看该对话的问答结果，并在历史对话基础上继续提问。 - 将鼠标移到历史对话，可编辑、分享、删除历史对话。 - 单击“查看更多”，可查看最近一个月的历史对话。 - 单击搜索框，可输入标题搜索历史对话。 - 勾选历史对话，可删除对应对话。
MCP服务器	单击研发问答窗口右上角图标，可配置MCP服务器。 说明 仅JetBrains系列IDE支持该功能。

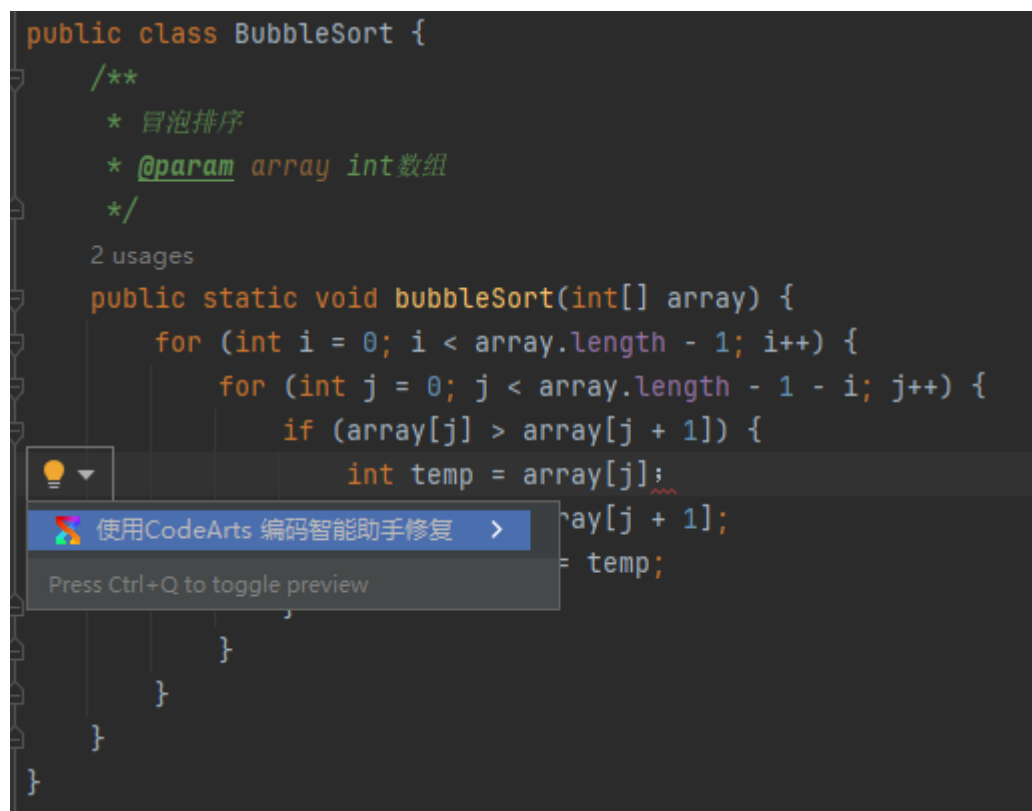
4.8.2 修复问题代码

如果代码中有缺陷，CodeArts 编码智能助手可以对代码进行修复。

步骤1 安装并登录CodeArts 编码智能助手插件。

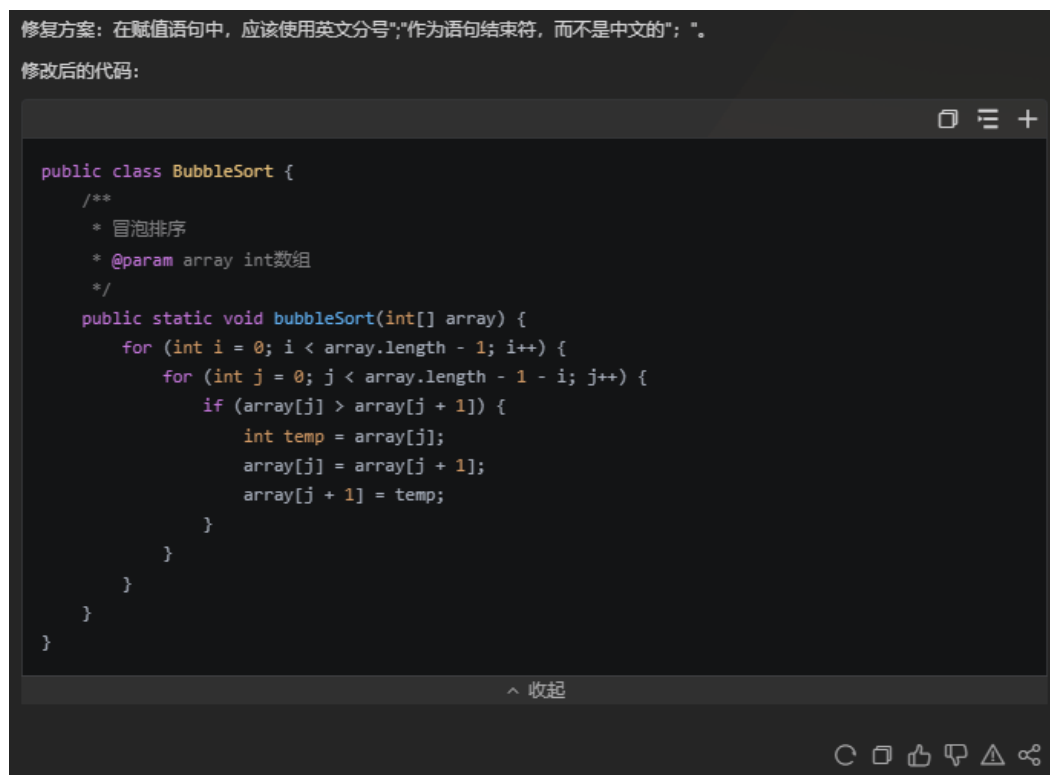
步骤2 在编辑器单击问题代码前的提示图标.

图 4-16 代码修复



步骤3 选择“使用CodeArts 编码智能助手修复”，自动将问题代码发送至CodeArts 编码智能助手研发问答窗口并给出修复后的代码。

图 4-17 修复后代码



步骤4 将修复后的代码拷贝至待修复的代码块处。

----结束

4.9 使用 CodeArts 编码智能助手生成单元测试用例

CodeArts 编码智能助手支持自动分析代码，找出适合进行单元测试的部分，并生成测试用例，帮助开发人员快速验证代码的正确性和稳定性。这些测试用例可以覆盖正常情况、异常情况和边界情况，帮助开发人员验证代码，保证代码在各种情况下都能够正确运行。CodeArts 编码智能助手支持生成Java、C、C++、Python、JavaScript等语言测试用例，示例以Java语言为例。

约束与限制

- CodeArts 编码智能助手生成的单元代码可能包含未引入的框架或包，需要手动引入。
- 不支持对abstract、private、main、constructor、toString、getter、setter、empty方法生成单元测试用例。

为方法生成单元测试用例

以Java语言以及CodeArts 编码智能助手生成的冒泡排序代码作为示例介绍生成测试用例。

步骤1 安装并登录CodeArts 编码智能助手插件。

步骤2 在编辑器选中通过CodeArts 编码智能助手生成的冒泡排序方法。

```
public static void bubbleSort(int[] array) {
    for (int i = 0; i < array.length - 1; i++) {
```

```
for (int j = 0; j < array.length - 1 - i; j++) {  
    if (array[j] > array[j + 1]) {  
        int temp = array[j];  
        array[j] = array[j + 1];  
        array[j + 1] = temp;  
    }  
}  
}
```

步骤3 选中代码生成单元测试，可通过以下两种方式。

- 方式一：单击右键，选择菜单“CodeArts 编码智能助手 > 单元测试”。
- 方式二：在研发问答窗口输入“/”，在弹出的菜单中选择“/tests”，单击或使用“Enter”快捷键发送。

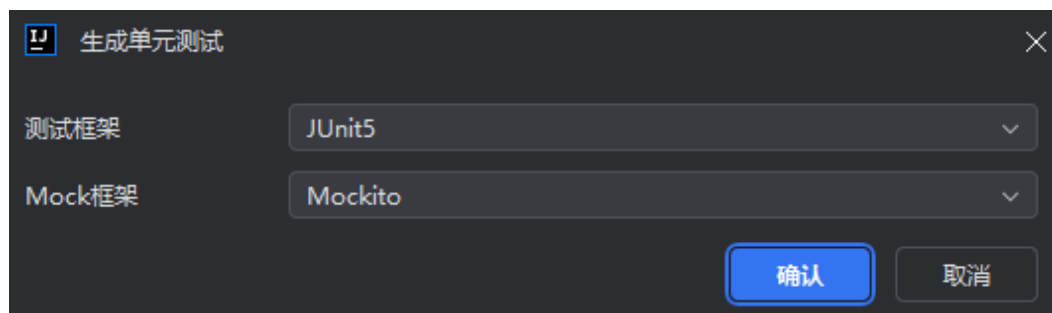
说明

插件在回答过程中输入框禁止发送问题。

步骤4 弹出“生成单元测试”对话框，选择“测试框架”和“Mock框架”，单击“确认”。

- 测试框架：包括JUnit4，JUnit5。
- Mock框架：目前仅支持Mockito。

图 4-18 生成单元测试



步骤5 CodeArts 编码智能助手将会在研发问答窗口输出测试用例代码及代码解释内容，开发者可以将生成的测试用例代码另存为文件后运行。

图 4-19 生成的单元测试用例

为了测试**bubbleSort**方法，我们可以编写几个单元测试用例，以确保方法在各种情况下都能正确排序数组。以下是一些可能的测试用例：

1. 测试空数组：确保方法能够处理空数组而不抛出异常。
2. 测试单元素数组：确保方法能够处理只有一个元素的数组。
3. 测试已排序数组：确保方法能够正确处理已经排序的数组。
4. 测试逆序数组：确保方法能够正确处理元素顺序完全相反的数组。
5. 测试包含重复元素的数组：确保方法能够正确处理包含重复元素的数组。
6. 测试随机数组：确保方法能够正确处理随机顺序的数组。

以下是使用JUnit框架编写的测试用例：

```
import org.junit.Test;
import static org.junit.Assert.*;

public class BubbleSortTest {

    @Test
    public void testEmptyArray() {
        int[] array = {};
        BubbleSort.bubbleSort(array);
        assertEquals(new int[] {}, array);
    }

    @Test
    public void testSingleElementArray() {
        int[] array = {1};
        BubbleSort.bubbleSort(array);
        assertEquals(new int[] {1}, array);
    }

    @Test
    public void testSortedArray() {
        int[] array = {1, 2, 3, 4, 5};
        BubbleSort.bubbleSort(array);
        assertEquals(new int[] {1, 2, 3, 4, 5}, array);
    }
}
```

----结束

为类生成单元测试用例

步骤1 安装并登录CodeArts 编码智能助手插件。

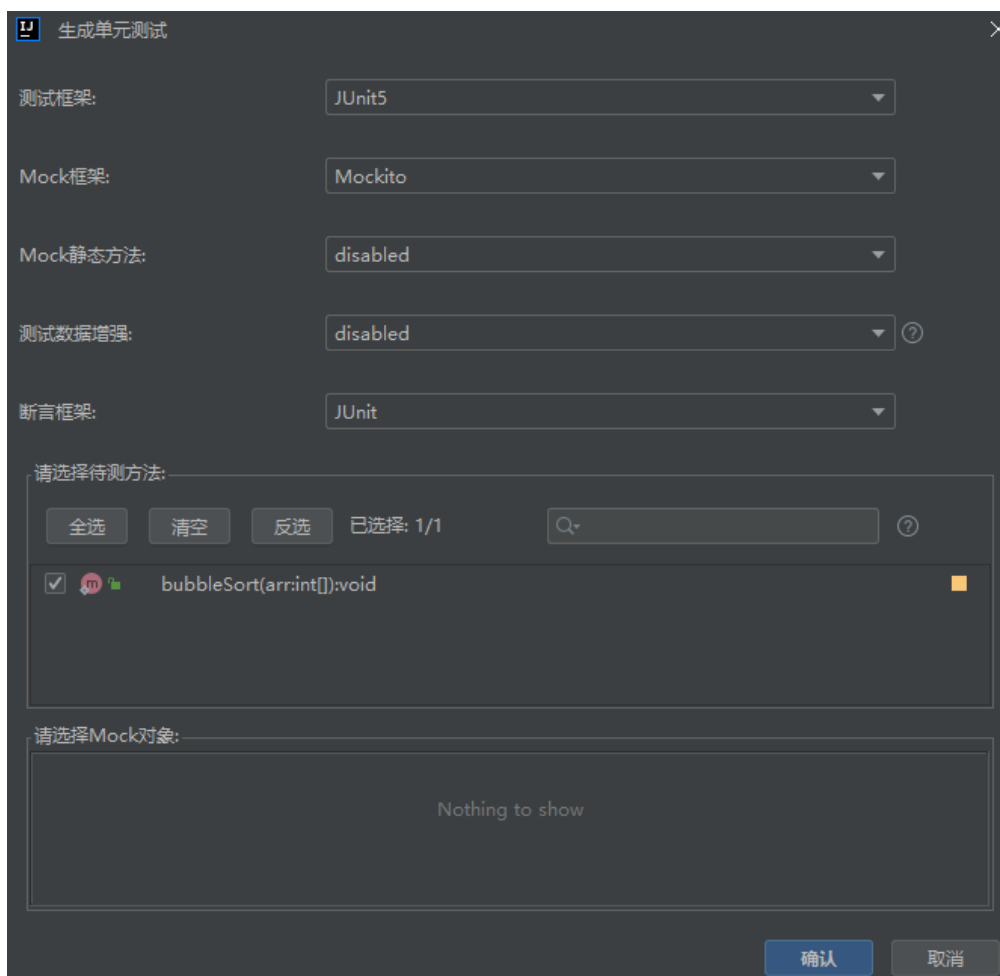
步骤2 在编辑器中代码类任意空白处，单击右键，选择菜单“CodeArts 编码智能助手 > 单元测试”。

步骤3 弹出“生成单元测试”对话框，根据需要选择以下内容。

- 测试框架：包括JUnit4，JUnit5。
- Mock框架：目前仅支持Mockito。
- Mock静态方法：是否启用模拟静态方法。
- 测试数据增强：是否增强边界数据。
- 断言框架：选择断言方式。
- 选择待测方法：选择生成测试用例方法。

- 选择Mock对象：根据实际情况选择Mock对象。

图 4-20 生成单元测试



步骤4 单击“确认”，CodeArts 编码智能助手将会根据表单内容，生成单元测试用例文件。

----结束

批量生成单元测试用例

步骤1 安装并登录CodeArts 编码智能助手插件。

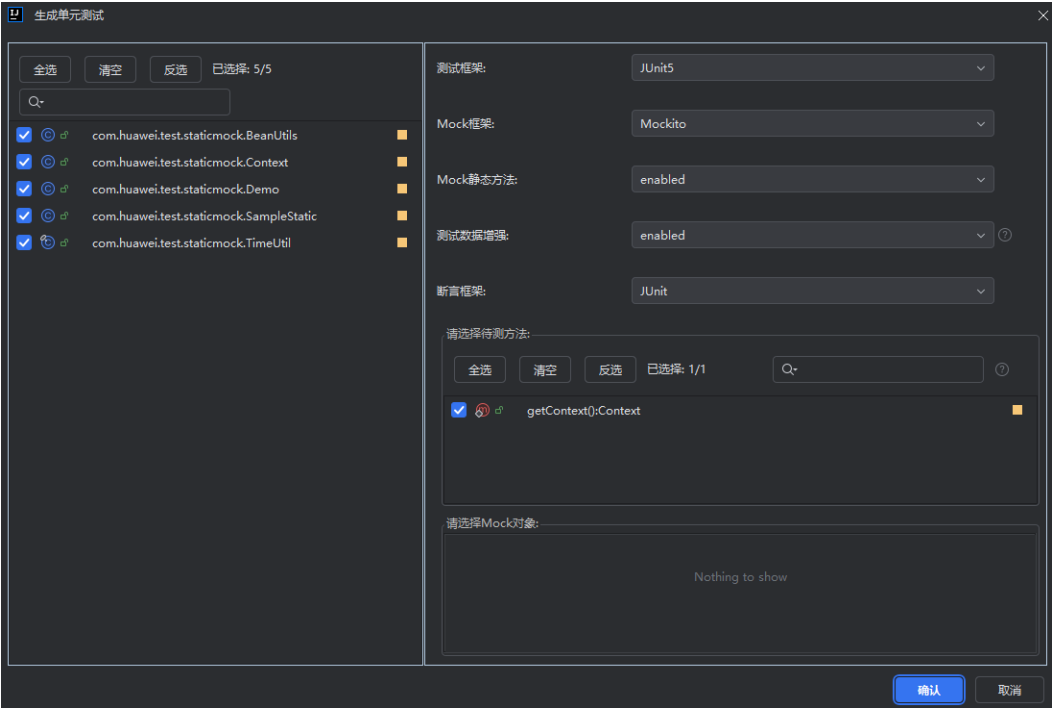
步骤2 开启目录级单元测试用例生成。

1. 打开IDE，单击左上角菜单栏“File > Settings...”，进入IDE设置界面。
2. 在设置界面左侧菜单选择“CodeArts 编码智能助手 > 单元测试生成”，勾选“开启目录级单元测试用例生成”。

步骤3 找到项目中需要批量生成测试用例的文件夹，单击右键，选择菜单“CodeArts 编码智能助手 > 单元测试”。

步骤4 弹出“生成单元测试”对话框，根据需要选择对应内容。

图 4-21 批量生成单元测试用例



- 步骤5** 单击“确认”，CodeArts 编码智能助手将会根据表单内容，批量生成单元测试用例文件。
- 结束

修复测试用例

- 步骤1** 安装并登录CodeArts 编码智能助手插件。
- 步骤2** 在编辑器打开一个测试文件，在文件空白地方单击右键，选择菜单“CodeArts 编码智能助手 > 测试用例修复（Beta）”。
- 步骤3** CodeArts 编码智能助手将会在研发问答窗口中给出修复后测试用例。

图 4-22 修复测试用例



----结束

相关操作

表 4-9 相关操作

操作项	说明
复制代码	单击回答框右上角  图标，可将代码复制至剪贴板。
插入代码	单击回答框右上角  图标，可将代码插入至当前打开文件光标所在位置。
保存代码	单击回答框右上角  图标，可新建文件将代码保存为文件。

操作项	说明
重新回答	如对当前回答不满意，可单击回答框右下角  图标，重新生成答案。
复制回答	单击回答框右下角  图标，可复制回答内容。
评价回答	单击回答框右下角  或  图标，可对回答给出评价。
投诉举报	单击回答框右下角  ，可对回答结果进行投诉举报。
分享	单击回答框右下角  ，可将回答结果导出为Markdown文件或生成图片。
新建对话	单击研发问答窗口右上角  图标，可打开新的问答窗口，开启新的问答。
对话历史	单击研发问答窗口右上角图标，可查看历史对话。 - 单击历史对话，可查看该对话的问答结果，并在历史对话基础上继续提问。 - 将鼠标移到历史对话，可编辑、分享、删除历史对话。 - 单击“查看更多”，可查看最近一个月的历史对话。 - 单击搜索框，可输入标题搜索历史对话。 - 勾选历史对话，可删除对应对话。
MCP服务器	单击研发问答窗口右上角图标，可配置MCP服务器。 说明 仅JetBrains系列IDE支持该功能。

4.10 使用 CodeArts 编码智能助手进行研发问答

在CodeArts 编码智能助手研发对话窗口中，用户可以随时提出问题，CodeArts 编码智能助手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。同时支持自定义人设，设置个性化提示词，CodeArts 编码智能助手将根据个性化指令提供更符合个人预期的回答。

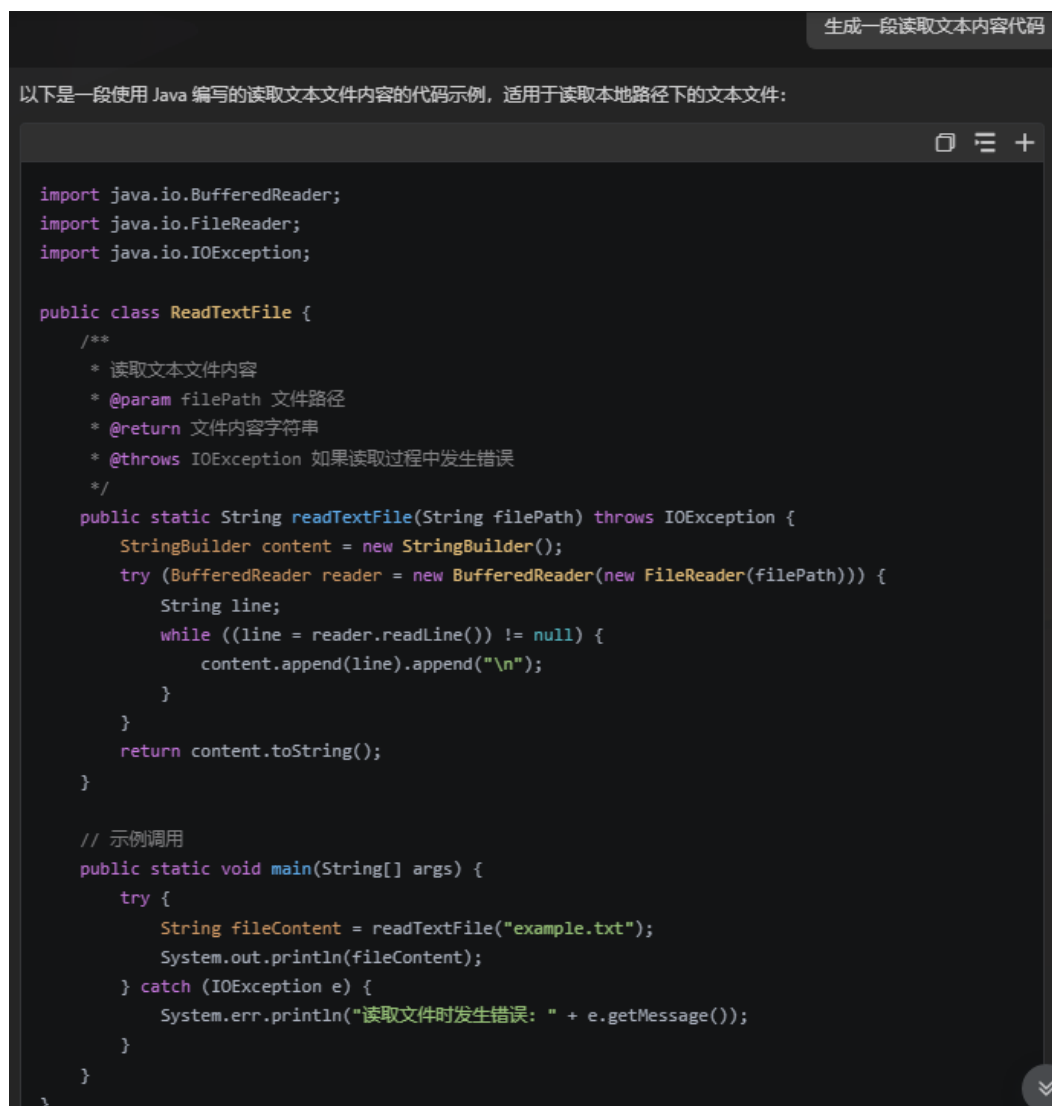
约束限制

CodeArts 编码智能助手支持回答Java、C、C++、Python、JavaScript等主流编程语言相关问题，对非研发相关问题，可能无法准确回答。

研发知识问答

- 步骤1 安装并登录CodeArts 编码智能助手插件。
- 步骤2 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”。
- 步骤3 CodeArts 编码智能助手将使用Java或其他语言生成一段读取文本内容的代码。

图 4-23 回答结果



----结束

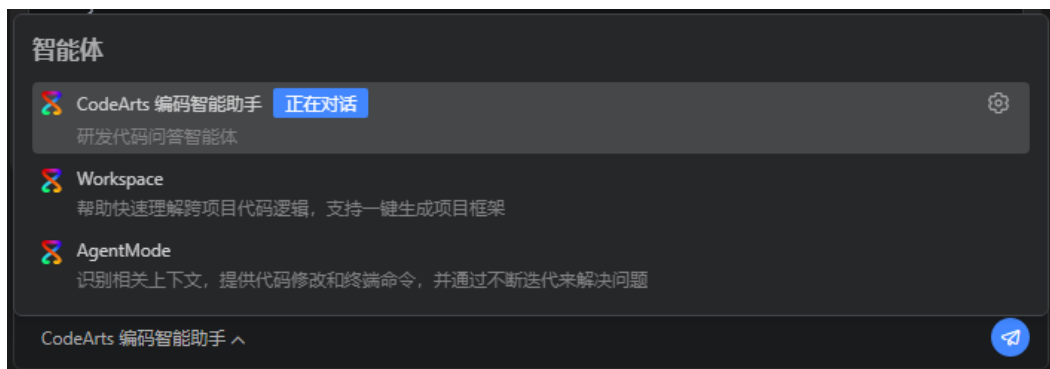
个人自定义人设研发知识问答

当前用户设置的个性化提示词，作用于当前用户。

步骤1 安装并登录CodeArts 编码智能助手插件。

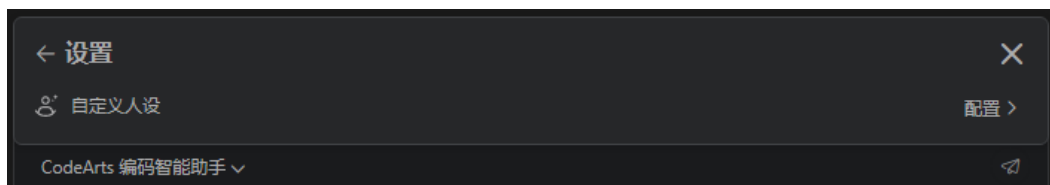
步骤2 在研发问答窗口单击“CodeArts 编码智能助手”旁的下拉箭头。

图 4-24 配置智能体



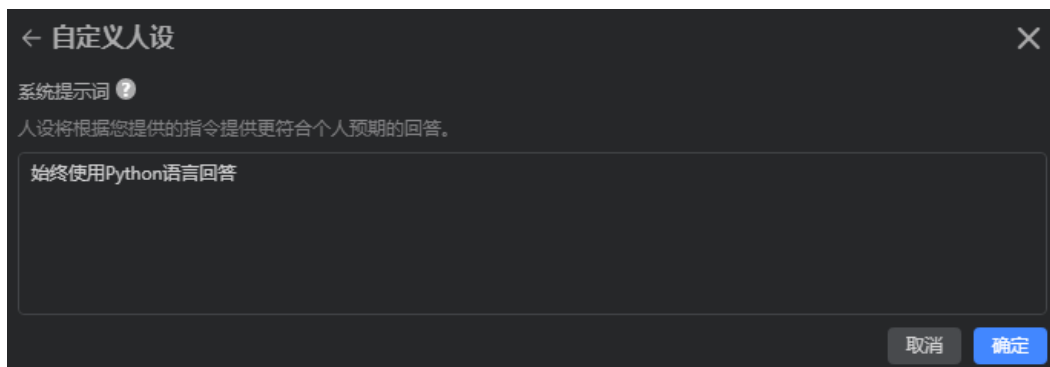
步骤3 将鼠标移动到智能体（可切换不同智能体），单击右侧，可自定义人设。

图 4-25 自定义人设



步骤4 单击“配置”，在“自定义人设”窗口输入自定义指令，如“始终使用Python语言回答”。

图 4-26 自定义提示词



步骤5 设置完成后，单击“确定”。

步骤6 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”。

步骤7 CodeArts 编码智能助手将使用Python语言生成一段读取文本内容的代码。

图 4-27 个性化回答



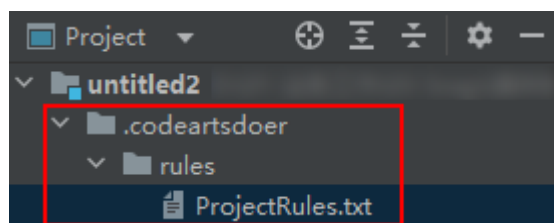
----结束

项目级自定义人设研发知识问答

用户在项目下设置的个性化提示词，作用于当前项目。

- 步骤1 安装并登录CodeArts 编码智能助手插件。
- 步骤2 在项目根目录下创建“.codeartsdoer”文件夹。
- 步骤3 在“.codeartsdoer”文件夹下创建“rules”文件夹。
- 步骤4 在“rules”文件夹下创建“ProjectRules.txt”文件，然后在文件中设置个性化提示词（内容的存储格式需为纯文本）。

图 4-28 自定义人设

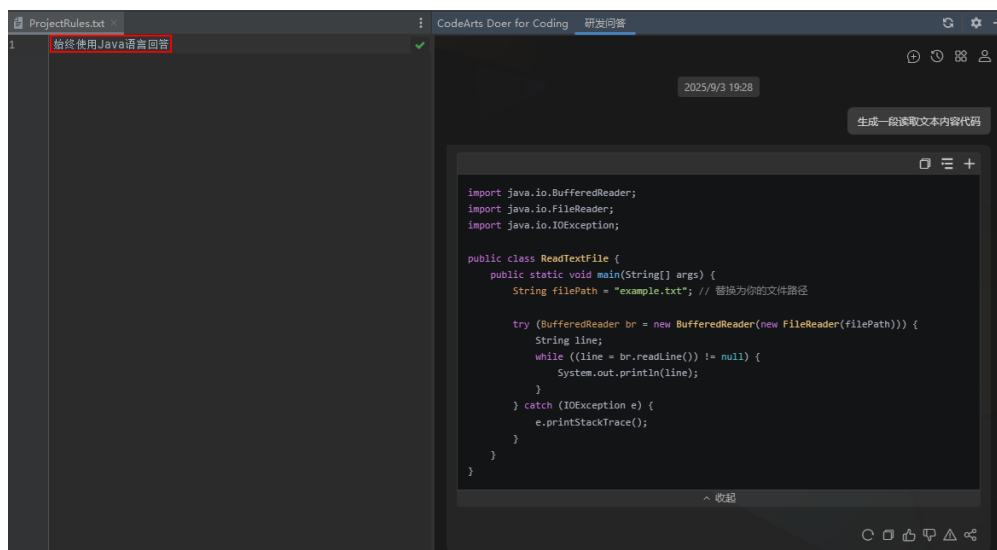


- 步骤5 设置完成后，在研发对话窗口中输入研发相关问题“生成一段读取文本内容代码”，CodeArts 编码智能助手将结合提示词生成一段读取文本内容的代码。

- 示例1

个性化提示词为“始终使用Java语言回答”，回答结果如下。

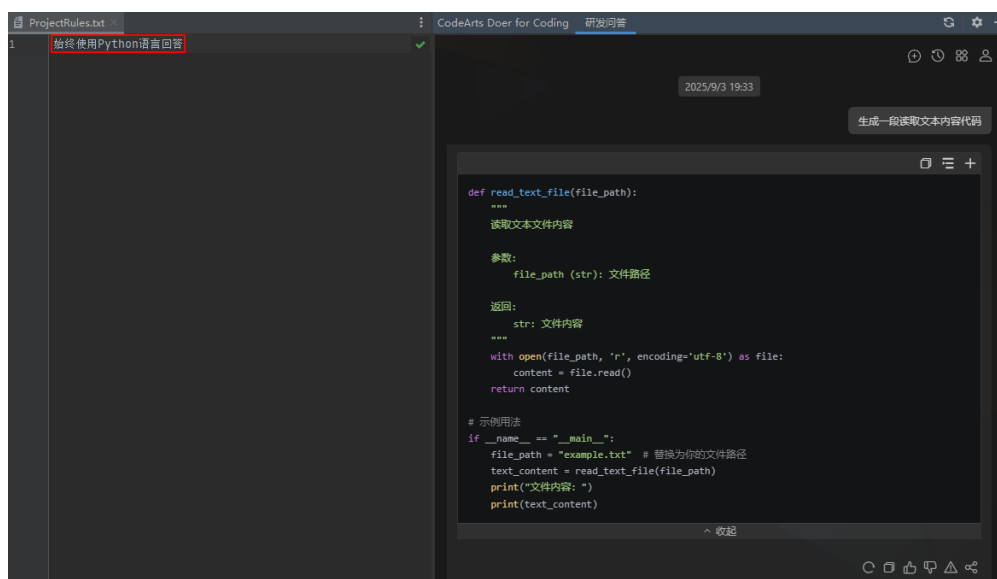
图 4-29 个性化回答



- 示例2

个性化提示词为“始终使用Python语言回答”，回答结果如下。

图 4-30 个性化回答



----结束

4.11 使用知识库增强 CodeArts 编码智能助手能力

CodeArts 编码智能助手支持企业租户管理员创建知识库、自定义知识库内容，利用知识库增强“生成代码”和“研发知识问答”场景能力。

操作步骤

- 步骤1 安装并登录CodeArts 编码智能助手插件。
- 步骤2 在研发问答窗口输入“#”，在列表中选择“Knowledge Bases”调用知识库。
- 步骤3 依次选择在知识库管理中已创建好的知识库及知识集，选定完毕后，即可使用CodeArts 编码智能助手生成代码或使用CodeArts 编码智能助手进行研发知识问答。

图 4-31 调用 RAG 知识库



----结束

4.12 使用 AgentMode 智能体进行研发问答

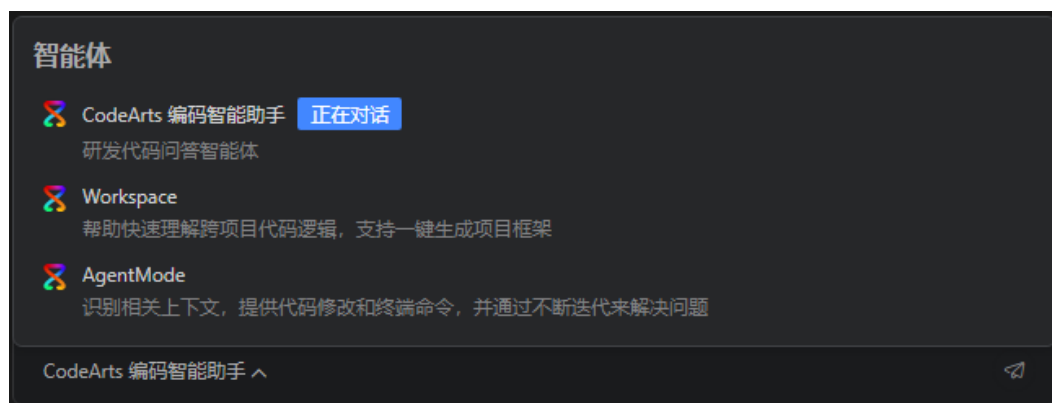
AgentMode智能体支持识别上下文，提供代码修改，并通过多轮对话解决问题。

操作步骤

步骤1 安装并登录CodeArts 编码智能助手插件。

步骤2 在研发问答窗口单击“CodeArts 编码智能助手”旁的下拉箭头，切换到AgentMode智能体。

图 4-32 切换智能体



步骤3 在对话框中输入想要完成的任务，单击或使用“Enter”快捷键发送。

说明

插件在回答过程中输入框禁止发送问题。

图 4-33 输入任务



步骤4 AgentMode智能体会给出相应执行计划，请根据实际需要进行操作。

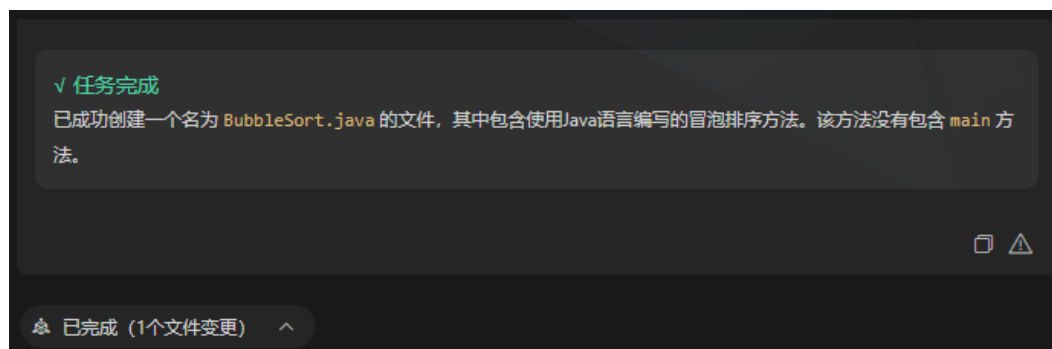
- 单击“同意”，允许智能体继续执行。
- 单击“拒绝”，拒绝智能体执行当前请求。
- 单击，修改智能体给出的执行计划，如：不包含main方法。然后单击“同意”。
- 在底部问答框输入修改指导，如：不包含main方法。然后单击“同意”。

图 4-34 输入指令



步骤5 经过多轮对话交互,直到得到满意回答。对于生成文件的回答会将文件放于暂存区。

图 4-35 生成文件



步骤6 单击文件旁的箭头,展开文件信息,请根据实际需要进行操作。

- 单击文件名,检视生成的内容。

- 将鼠标移至文件，单击右侧 ☒ 接受文件，单击 ☐ 拒绝文件。
- 存在多个文件可以直接单击“全部接受”接受全部文件，单击“全部拒绝”拒绝全部文件。

说明

接受文件后会将文件保存至项目根目录。

图 4-36 文件处理



----结束