

SysY2022E 扩展语言编辑器

SysY2022E Language Support

用户使用手册

成员信息: 李晓坤 信息安全 信安 211 U202141863

王若凡 信息安全 信安 211 U202141852

王宠爱 信息安全 信安 211 U202141853

成豪 信息安全 信安 211 U202141880

指导教师: 崔 晓 龙

目录

1. 引言	3
1.1. 编写目的	3
1.2. 项目背景	3
1.3. 定义	3
1.3.1. 编辑器	3
1.4. 参考资料	3
2. 软件概述	5
2.1. 背景	5
2.2. 功能	5
2.3. 性能	6
2.3.1. 精度	6
2.3.2. 时间特性	6
2.3.3. 灵活性	6
2.4. 安全保密	6
3. 运行环境	7
3.1. 硬件	7
3.2. 操作系统	7
3.3. 支持软件	7
3.3.1. Visual Studio Code	7
3.3.2. NET Framework	7
4. 使用说明	8
4.1. 安装和初始化	8
4.2. 启动服务端和客户端	8
4.2.1. 服务端	9
4.2.2. 客户端	9
4.3. 系统具体操作	9
4.3.1. 语法高亮和悬浮提示	9
4.3.2. 语法检查与错误提示	9
4.3.4. 静态语义检查	10
4.3.5. 修复建议	11
4.3.6. 程序错误检查	11
4.3.7. 代码重构	11
4.4. 出错和恢复	12
4.4.1. IDE 集成	13

4.4.2. 语法高亮和悬浮提示	13
4.4.3. 语法检查和错误提示	13
4.4.4. 静态语义检查	13
4.4.5. 修复建议和程序错误检查	14
4.4.6. 代码重构	14

1. 引言

1.1. 编写目的

为了让用户能够轻松上手并高效使用我们的编辑器，我们编写了此文档。手册中包含了编辑器的安装、配置、界面说明、功能介绍、操作指南以及常见问题解答等内容，旨在为用户提供全面的帮助和支持，确保用户使用 SysY2022E 扩展语言编辑器时能够流畅地进行编程、代码编辑和项目管理，从而提升工作效率和用户体验。

1.2. 项目背景

本项目源于对现有 SysY2022 编程语言的改进和扩展需求的迫切要求，旨在为广大开发者打造一个更强大、更灵活的代码编辑和开发环境。通过扩展的 SysY2022 语言（我们称之为 SysY2022E），本项目致力于开发一款全新的编辑器，不仅支持 SysY2022E 的全部语法特性，还提供了丰富的代码编辑、调试和测试功能。

1.3. 定义

1.3.1. 编辑器

编辑器是软体程序，一般是指用来修改电脑档案的编写软件，但也有人称 PE2、HE4（汉书）……等文书软件为编辑器。常见的编辑器有文本编辑器、网页编辑器、源程序编辑器、图像编辑器，声音编辑器和视频编辑器等。

程序编辑器是指用来进行编辑程序的软体程序，能把存在计算机中的源程序显示在屏幕上，然后根据需要进行增加、删除、替换和联结等操作。

1.4. 参考资料

- [1] 卢慧琼, 李百龄, 张溶溶. C 语言的结构编辑器[J]. 计算机研究与发展, 1986, (07):22-31.
- [2] 宋国钦. 编辑器设计技术[J]. 现代计算机, 1996, (05):21-23.
- [3] 微软 c++ language support 文档:

<https://learn.microsoft.com/lt-lt/cpp/overview/visual-cpp-language-conform>

[ance?view=msvc-170.](#)

- [4] VScode 插件开发文档:

<https://rackar.github.io/vscode-ext-doccn/get-started/your-first-extension.html#%E5%BC%80%E5%8F%91%E6%8F%92%E4%BB%B6>.

- [5] LSP 官方文档:

<https://vscode.github.net.cn/api/language-extensions/language-server-extension-guide#google-vignette>.

- [6] LSP 官方示例项目:

<https://github.com/microsoft/vscode-extension-samples/tree/main/lsp-sample>
.

- [7] ANTLR4 官方文档:

<https://pragprog.com/titles/tpantlr2/the-definitive-antlr-4-reference/>.

- [8] ANTLR4 官方 github 项目: <https://github.com/antlr/antlr4>.

- [9] 编译原理教材: <https://book.douban.com/subject/26736235/>.

- [10] SysY2022 编译器大赛项目参考:

<https://code.educoder.net/ph7n2ofui/SysyCompiler-Arm>.

- [11] Clangd 官方文档: <https://clangd.llvm.org/>.

- [12] Clangd 官方项目: <https://github.com/clangd/vscode-clangd>.

2. 软件概述

2.1. 背景

SysY2022, 作为 C 语言的一个子集, 原本是为了“全国大学生系统能力大赛-编译器大赛”而特别设计的编程语言。由于其简洁的语法和易于扩展的特性, SysY2022 很快在学术和教育领域获得了关注。然而, 随着用户对该语言的深入了解和广泛使用, 原始的编程环境已经无法满足日益增长的复杂编程需求。

为了满足用户对于更强大、更灵活编程环境的需求, SysY2022E 扩展语言编辑器应运而生。这款编辑器不仅继承了原始 SysY2022 编辑器的核心功能, 更在语法支持、代码编辑、调试和测试等方面进行了全面的扩展和优化。

2.2. 功能

SysY2022E 扩展语言编辑器主要包括以下功能:

- 1) **IDE 集成:** SysY2022 扩展语言编辑器能够与 Visual Studio Code 集成, 允许用户通过 vscode 直接编辑、运行和调试 SysY2022E 程序。集成后, 编辑器应提供与 vscode 一致的界面风格和用户体验, 同时保留 SysY2022E 语言的特定功能。
- 2) **语法高亮和悬浮提示:** 实时对 SysY2022E 程序进行语法高亮, 使得代码更易于阅读和理解。同时, 当鼠标悬停在标识符(如变量名、函数名等)上时, 编辑器应能够显示相应的信息或提示, 这些信息包括但不限于该标识符代表的对象和类型。
- 3) **语法检查与错误提示:** 内置的语法检查器会实时分析代码, 检测语法错误; 这包括但不限于变量声明、函数定义、表达式书写方面的错误。当检测到语法错误时, 编辑器会在代码中相应的位置标出错误, 并提供错误提示信息。
- 4) **静态语义检查:** 通过内置的静态语义分析器, 在编译之前对 SysY2022E 程序进行详细的语义检查。这些检查不仅限于语法层面, 还包括对代码逻辑、类型、作用域等方面的深入分析。
- 5) **修复建议:** 自动分析 SysY2022E 程序中检测到的语法错误和静态语义错误, 并基于这些错误提供针对性的修复建议。
- 6) **程序错误检查:** 自动检测 SysY2022E 程序中存在的常见错误, 以帮助用户提高代码质量和减少潜在的运行时问题。
- 7) **代码重构:** 提供全面的代码重构支持, 帮助用户改善代码结构、提高代码质量和维护性。

2.3. 性能

2.3.1. 精度

- 1) 输入数据精度：准确接收用户输入的 SysY2022E 程序代码，包括字符、数字、特殊符号等，确保无乱码、丢失或重复字符的现象；数值的输入精度与 SysY2022E 语言的规范一致；准确解析并处理其他类型的数据输入（如布尔值、枚举类型等）。
- 2) 输出数据精度：准确显示代码，能够输出详细的错误信息，包括错误的类型、位置、描述等，确保信息的准确性和清晰度，以及提供准确的信息和建议。
- 3) 传输过程中的精度：能保证保存加载过程中文件完好，以及与 IDE 集成过程中接口传输数据的完整一致。

2.3.2. 时间特性

- 1) 响应时间：迅速响应用户操作，时间被控制在毫秒级，用户界面流畅。
- 2) 更新处理时间：在毫秒级内完成代码分析和建议生成，且检查过程快速且准确，不影响用户的正常编辑操作。
- 3) 数据转换和传送时间：具有较快的数据转换、导入速度以及处理速度。
- 4) 计算时间：采用高效的算法和数据结构有效减少计算时间，支持实时反馈，同时能够确保结果的准确性和可靠性。

2.3.3. 灵活性

系统能适应如下变化，并能及时重新部署投入运行。

- 1) 部分硬件的变化（如打印机）；
- 2) 网络环境的变化；
- 3) 运行环境的变化：支持跨平台运行，能够在 Windows、Linux、macOS 等多种操作系统上稳定运行。

2.4. 安全保密

本系统对用户的个人信息等隐私完全保密。

3. 运行环境

3.1. 硬件

PC 机，4G 内存，双核处理器及以上均可。

3.2. 操作系统

Windows: Windows 10 或更高版本。

3.3. 支持软件

3.3.1. Visual Studio Code

从 [Visual Studio Code 官网](#) 下载并安装最新版。

推荐使用最新稳定版本，确保兼容性和功能完整性。

3.3.2. NET Framework

确保安装最新的 .NET Framework 版本（通常安装 VS Code 时会自动安装）。

4. 使用说明

4.1. 安装和初始化

步骤 1: 安装 vscode:

- ◆ 安装 Visual Studio Code;
- ◆ 从 [Visual Studio Code 官网](#) 下载并安装 VS Code。

步骤 2: 在 vscode 中通过 vsix 安装 sysY 扩展插件:

- ◆ 在扩展中, 点击右上角 ... 选择从 VSIX 安装:

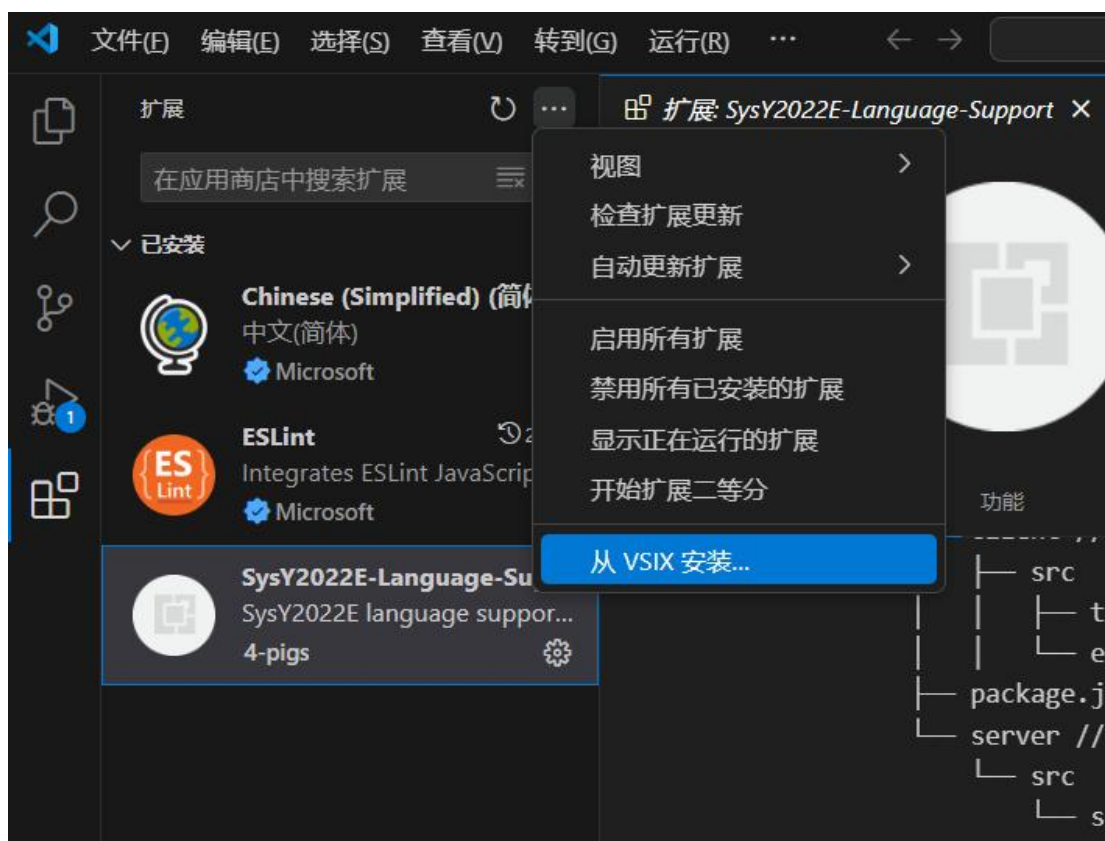


图 1 从 VSIX 安装扩展

在弹出的文件选择界面, 找到 SysY2022E-Language-Support-1.0.0.vsix 文件, 点击安装即可进行安装。

4.2. 启动服务端和客户端

4.2.1. 服务端

服务器默认启动。

4.2.2. 客户端

相应插件安装后，编辑器即可自动启用。

4.3. 系统具体操作

从扩展安装完成 SysY2022E-Language-Support-1.0.0.vsix 插件后，编辑器即可自动启用。

4.3.1. 语法高亮和悬浮提示

把鼠标放在相应程序上即可查看语法高亮与悬浮提示信息。

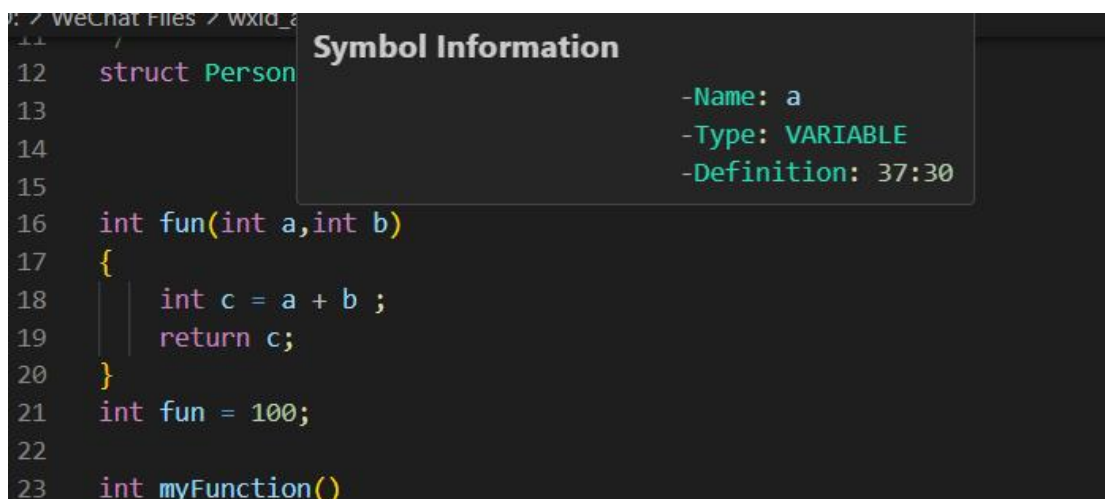


图 2 查看语法高亮和悬浮提示

4.3.2. 语法检查与错误提示

- ◆ 对常量 qq 进行修改给出错误提示。

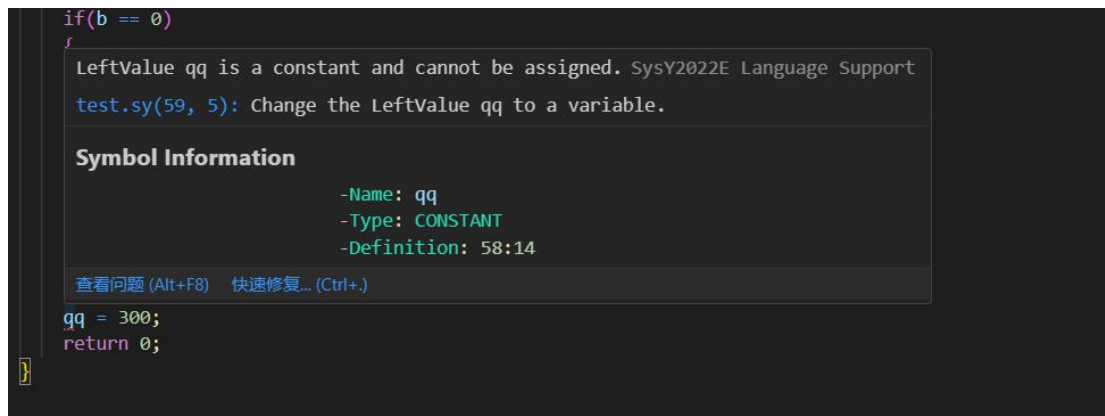


图 3 修改常量的错误提示

- ◆ 变量 m 未定义给出相应错误提示。

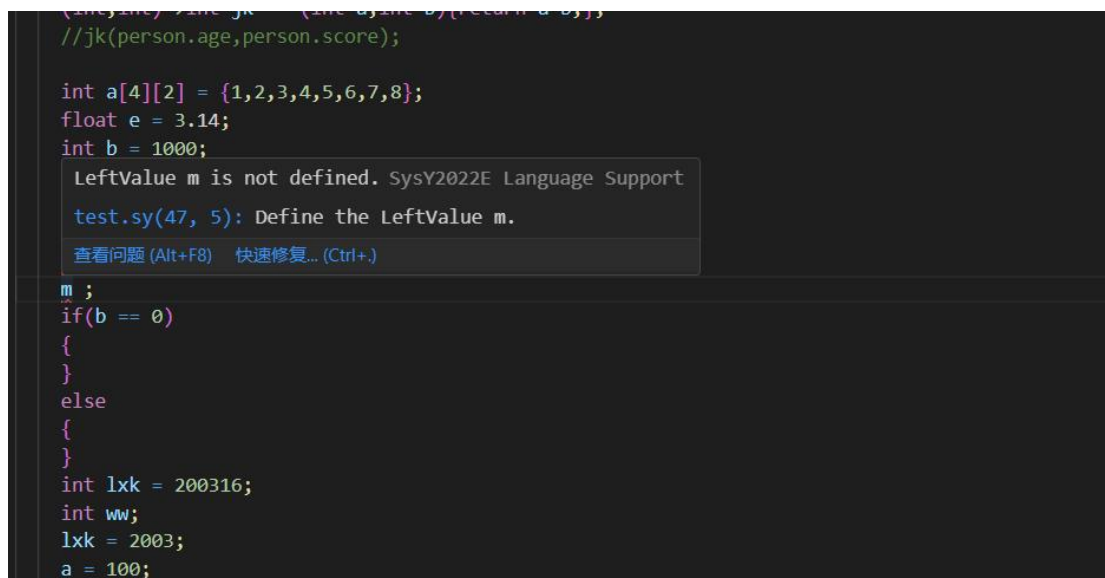


图 4 修改未定义变量的错误提示

4.3.3. 静态语义检查

- ◆ 在主函数中尝试给出 fun 函数错误的形参数量给出提示。

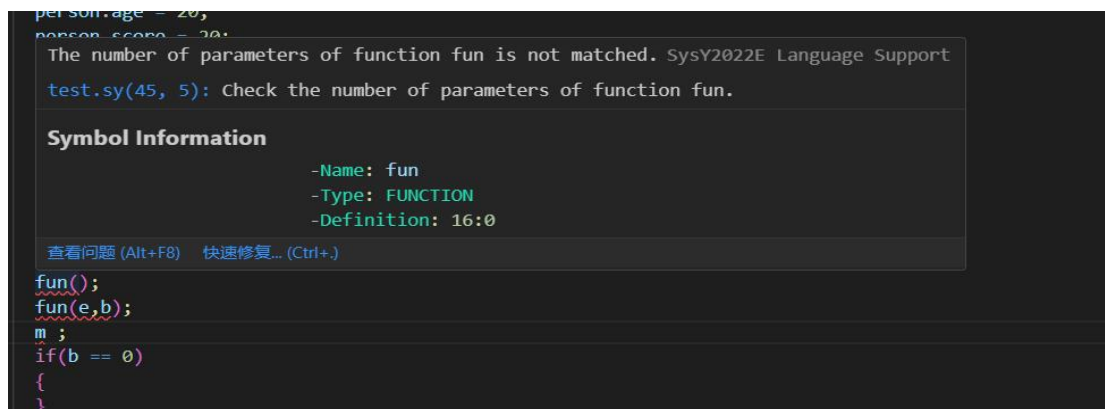


图 5 形参数量错误提示

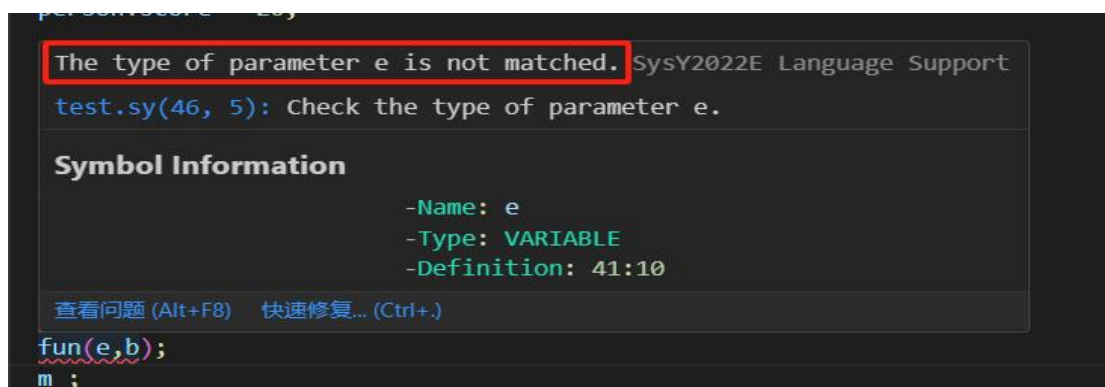


图 6 形参类型错误提示

4.3.4. 修复建议

- ◆ 对主函数中对 fun 函数错误的形参类型的建议。

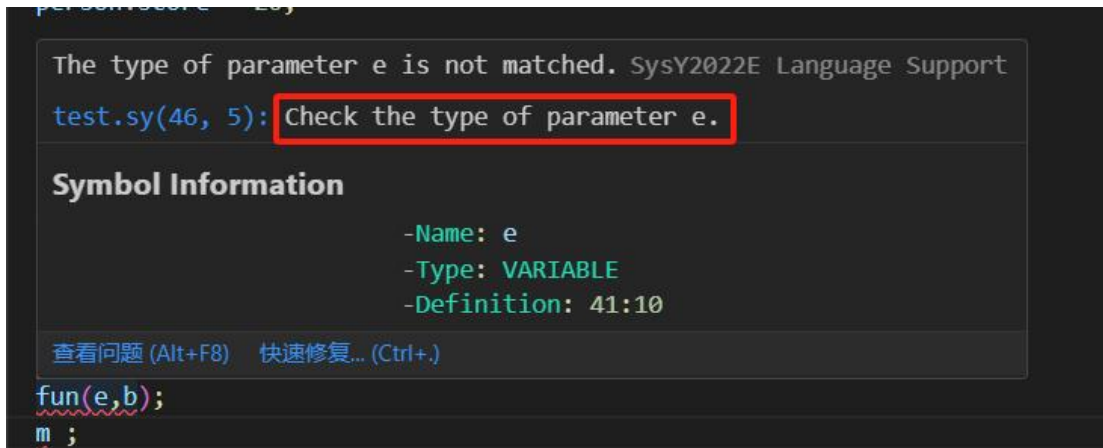


图 7 修改错误形参类型的建议

4.3.5. 程序错误检查

- ◆ 运行程序后，在问题栏检查出警告与错误信息。

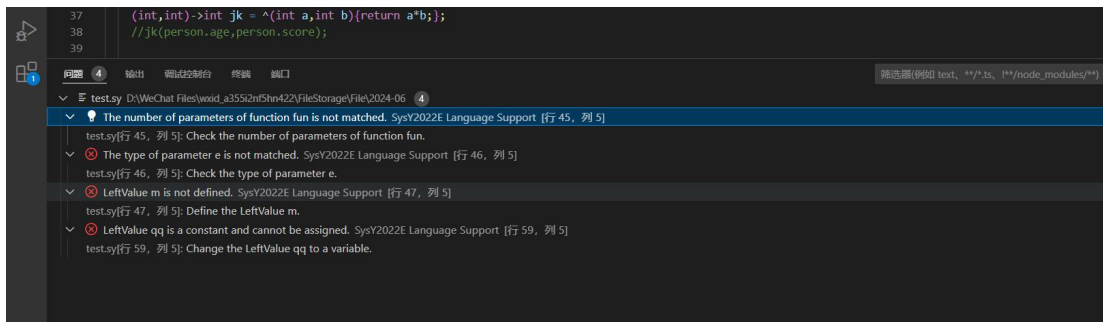


图 8 显示警告与错误信息

4.3.6. 代码重构

- ◆ 对函数 myFuction 进行重构，可在多种重构操作中进行选择。



图 9 三种重构选择

- ◆ 转化为大写操作。

```
void myFUNCTION()
{
    int temp=100;
}
float testFun()
```

图 10 转化为大写

- ◆ 抽取为新函数操作。

```
void void newFunction () {
myFunction()
}
myFuction()
{
    int temp = 100;
}
```

图 11 抽取为新函数

```
void myFunction()
{
    int temp=100;
}
float testFun()
{
    return 100 ;
}
int main()
{
    Person person;
    person.age = 20;
    person.score = 20;

    (int,int)->int jk = ^(int a,int b){return a*b;};
    //jk(person.age,person.score);

    int a[4][2] = {1,2,3,4,5,6,7,8};
    int temp = 100;
```

图 12 内联函数

4.4. 出错和恢复

4.4.1. IDE 集成

- 1) 问题: SysY2022E 扩展没有正确安装或无法在 VS Code 中正常工作。
- 2) 解决方案:
 - ◆ 检查 VSIX 文件是否正确下载并且版本兼容。
 - ◆ 确认 VS Code 是最新版本并重启 VS Code。
 - ◆ 尝试重新安装扩展: 打开 VS Code, 转到扩展视图 (Ctrl+Shift+X)。
 - ◆ 点击右上角的省略号 (...) 选择 “Install from VSIX...”。
 - ◆ 选择 VSIX 文件进行重新安装。

4.4.2. 语法高亮和悬浮提示

- 1) 问题: 语法高亮和悬浮提示功能无法正常工作。
- 2) 解决方案:
 - ◆ 1. 确认文件类型正确识别。确保文件扩展名和 SysY2022E 扩展语言编辑器支持的文件类型匹配。
 - ◆ 2. 重新加载窗口 (Ctrl+Shift+P -> Reload Window)。关闭所有打开的文件并重新打开。

4.4.3. 语法检查和错误提示

- 1) 问题: 语法检查和错误提示没有显示或不准确。
- 2) 解决方案:
 - ◆ 确认 SysY2022E 扩展已正确加载。
 - ◆ 查看问题面板来检查是否存在语法错误 (Ctrl+Shift+M)。
 - ◆ 确保代码是最新保存的状态。

4.4.4. 静态语义检查

- 1) 问题: 静态语义检查功能未正常运行或提示错误。
- 2) 解决方案:
 - ◆ 确保所有配置文件都已正确设置, 如 .sy 配置文件 (如果有的话)。
 - ◆ 重新加载窗口 (Ctrl+Shift+P -> Reload Window)。
 - ◆ 重启 VS Code, 这通常会重新加载扩展和其功能。

4.4.5. 修复建议和程序错误检查

- 1) 问题：修复建议未显示或无法应用。
- 2) 解决方案：
 - ◆ 确认您选中的代码区域包含需要修复的部分。
 - ◆ 右键点击选中的代码部分，确保“快速修复”选项可用。
 - ◆ 确保您使用的是最新版本的 SysY2022E 扩展。

4.4.6. 代码重构

- 1) 问题：代码重构功能无法正常使用。
- 2) 解决方案：
 - ◆ 确认选中的代码区域适用于重构。
 - ◆ 右键点击选中的代码部分，确保“重构”选项可用。
 - ◆ 确保您使用的是最新版本的 SysY2022E 扩展。
 - ◆ 如果重构菜单没有出现，尝试重启 VS Code。
- 3) 如果上述方法都无法解决您的问题，尝试以下步骤：
 - ◆ 清除缓存：清除 VS Code 缓存可能有助于解决一些持续性问题。
 - ◆ 查看日志：打开输出面板 (View -> Output)，选择相关的日志来获取可能的错误信息。
 - ◆ 重新安装：如果所有其他方法都失败了，卸载并重新安装 VS Code 可能有助于解决问题。
 - ◆ 当您面临问题时，确保检查扩展的官方文档或联系支持人员，可能有已知的问题或解决方案。