



(12) 发明专利申请

(10) 申请公布号 CN 116363256 A

(43) 申请公布日 2023. 06. 30

(21) 申请号 202310282197.1

(22) 申请日 2023.03.17

(71) 申请人 武汉大学

地址 430000 湖北省武汉市武昌区珞珈山

(72) 发明人 玄跻峰 彭钰荃 丁正杰

(74) 专利代理机构 武汉智权专利代理事务所

(特殊普通合伙) 42225

专利代理师 张凯

(51) Int. Cl.

G06T 11/20 (2006.01)

G09B 5/02 (2006.01)

G06F 11/36 (2006.01)

G06V 30/10 (2022.01)

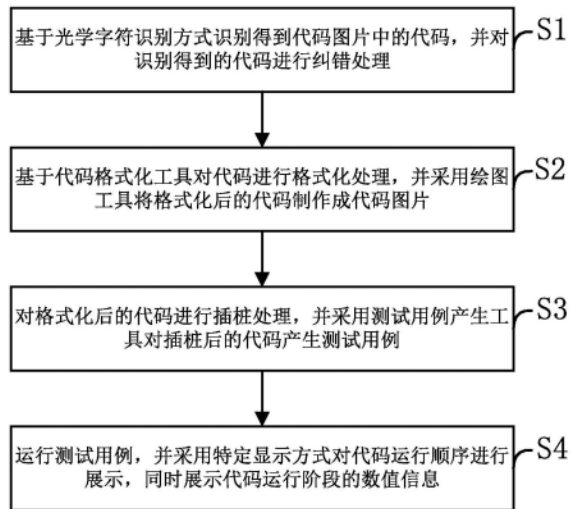
权利要求书2页 说明书7页 附图1页

(54) 发明名称

一种代码图片的可视化展示实现方法及装置

(57) 摘要

本发明公开了一种代码图片的可视化展示实现方法及装置,涉及软件工程技术领域,该方法包括基于光学字符识别方式识别得到代码图片中的代码,并对识别得到的代码进行纠错处理;基于代码格式化工具对代码进行格式化处
理,并采用绘图工具将格式化后的代码制作成代码图片;对格式化后的代码进行插桩处理,并采用测试用例产生工具对插桩后的代码产生测试用例;运行测试用例,并采用特定显示方式对代码运行顺序进行展示,同时展示代码运行阶段的数值信息。本发明能够有效帮助用户增进对于代码图片中代码片段的理解,简化了用户实现代码的流程。



1. 一种代码图片的可视化展示实现方法,其特征在于,具体包括以下步骤:
基于光学字符识别方式识别得到代码图片中的代码,并对识别得到的代码进行纠错处理;
基于代码格式化工具对代码进行格式化处理,并采用绘图工具将格式化后的代码制作成代码图片;
对格式化后的代码进行插桩处理,并采用测试用例产生工具对插桩后的代码产生测试用例;
运行测试用例,并采用特定显示方式对代码运行顺序进行展示,同时展示代码运行阶段的数值信息。
2. 如权利要求1所述的一种代码图片的可视化展示实现方法,其特征在于,所述对识别得到的代码进行纠错处理,具体为:
基于启发式规则对代码进行纠错处理,以将光学字符识别时识别的错误字符替换为正确字符。
3. 如权利要求1所述的一种代码图片的可视化展示实现方法,其特征在于:所述代码图片中,代码间采用等宽字体。
4. 如权利要求3所述的一种代码图片的可视化展示实现方法,其特征在于,所述对格式化后的代码进行插桩处理,具体步骤包括:
采用源代码分析和转换工具对代码进行遍历,在每个语句前后插入打印该代码位置编号以及其参数值的代码。
5. 如权利要求4所述的一种代码图片的可视化展示实现方法,其特征在于,所述在每个语句前后插入打印该代码位置编号以及其参数值的代码,具体为:
对于非循环的所有语句,在语句前插入打印该语句行编号的代码;
对于循环语句,则在语句前插入打印该循环开始与结束位置的代码;
对于赋值语句、初始值语句,以及输入特定参数名的调用函数语句,则另外在语句后插入打印该值的代码。
6. 如权利要求4所述的一种代码图片的可视化展示实现方法,其特征在于:
在采用源代码分析和转换工具对代码进行遍历的同时,获取当前代码区块的位置信息;
所述位置信息包括代码起始行与结束字节编号,以及代码结束行与结束字节编号。
7. 如权利要求1所述的一种代码图片的可视化展示实现方法,其特征在于:在采用测试用例产生工具对插桩后的代码产生测试用例时,在产生的测试用例中限制代入源代码的参数。
8. 如权利要求7所述的一种代码图片的可视化展示实现方法,其特征在于,当采用测试用例产生工具对插桩后的代码产生测试用例之后,还包括:
采用源代码分析和转换工具遍历测试用例,判断测试用例中被测函数的参数是否完整,以挑选得到具备预设可视化理解效果的测试用例。
9. 如权利要求1所述的一种代码图片的可视化展示实现方法,其特征在于:所述数值信息包括代码行编号、循环语句位置编号和代码中各项参数的数值。
10. 一种代码图片的可视化展示实现装置,其特征在于,包括:

识别模块,其用于基于光学字符识别方式识别得到代码图片中的代码,并对识别得到的代码进行纠错处理;

第一处理模块,其用于基于代码格式化工具对代码进行格式化处理,并采用绘图工具将格式化后的代码制作成代码图片;

第二处理模块,其用于对格式化后的代码进行插桩处理,并采用测试用例产生工具对插桩后的代码产生测试用例;

执行模块,其用于运行测试用例,并采用特定显示方式对代码运行顺序进行展示,同时展示代码运行阶段的数值信息。

一种代码图片的可视化展示实现方法及装置

技术领域

[0001] 本发明涉及软件工程技术领域,具体涉及一种代码图片的可视化展示实现方法及装置。

背景技术

[0002] 在以数据结构与算法分析为首的教科书中,经常出现不易理解的代码。以往,学生在阅读此类书籍时,由于教科书中的代码为静态的文字,学生们对于教科书中的代码具体是如何运行会产生疑问,只能采用手动方式将书中代码在电脑上实际敲打出来以运行。例如,学生不理解教科书上一个排序的算法,只能手动将代码实现并设计测试用例带入该算法函数,并查看运行结果。

[0003] 然而,上述实现方式对于学生理解代码还是效果不佳,主要原因是:(1)对于运行时代码的执行顺序不了解;(2)对于代码运行时(例如进行排序)各项参数的变动没办法掌握;(3)手动敲打代码方式极为耗时。

发明内容

[0004] 针对现有技术中存在的缺陷,本发明的目的在于提供一种代码图片的可视化展示实现方法及装置,能够有效帮助用户增进对于代码图片中代码片段的理解,简化了用户实现代码的流程。

[0005] 为达到以上目的,本发明提供一种代码图片的可视化展示实现方法,具体包括以下步骤:

[0006] 基于光学字符识别方式识别得到代码图片中的代码,并对识别得到的代码进行纠错处理;

[0007] 基于代码格式化工具对代码进行格式化处理,并采用绘图工具将格式化后的代码制作成代码图片;

[0008] 对格式化后的代码进行插桩处理,并采用测试用例产生工具对插桩后的代码产生测试用例;

[0009] 运行测试用例,并采用特定显示方式对代码运行顺序进行展示,同时展示代码运行阶段的数值信息。

[0010] 在上述技术方案的基础上,所述对识别得到的代码进行纠错处理,具体为:

[0011] 基于启发式规则对代码进行纠错处理,以将光学字符识别时识别的错误字符替换为正确字符。

[0012] 在上述技术方案的基础上,所述代码图片中,代码间采用等宽字体。

[0013] 在上述技术方案的基础上,所述对格式化后的代码进行插桩处理,具体步骤包括:

[0014] 采用源代码分析和转换工具对代码进行遍历,在每个语句前后插入打印该代码位置编号以及其参数值的代码。

[0015] 在上述技术方案的基础上,所述在每个语句前后插入打印该代码位置编号以及其

参数值的代码,具体为:

[0016] 对于非循环的所有语句,在语句前插入打印该语句行编号的代码;

[0017] 对于循环语句,则在语句前插入打印该循环开始与结束位置的代码;

[0018] 对于赋值语句、初始值语句,以及输入特定参数名的调用函数语句,则另外在语句后插入打印该值的代码。

[0019] 在上述技术方案的基础上,

[0020] 在采用源代码分析和转换工具对代码进行遍历的同时,获取当前代码区块的位置信息;

[0021] 所述位置信息包括代码起始行与结束字节编号,以及代码结束行与结束字节编号。

[0022] 在上述技术方案的基础上,在采用测试用例产生工具对插桩后的代码产生测试用例时,在产生的测试用例中限制代入源代码的参数。

[0023] 在上述技术方案的基础上,当采用测试用例产生工具对插桩后的代码产生测试用例之后,还包括:

[0024] 采用源代码分析和转换工具遍历测试用例,判断测试用例中被测函数的参数是否完整,以挑选得到具备预设可视化理解效果的测试用例。

[0025] 在上述技术方案的基础上,所述数值信息包括代码行编号、循环语句位置编号和代码中各项参数的数值。

[0026] 本发明提供了一种代码图片的可视化展示实现装置,包括:

[0027] 识别模块,其用于基于光学字符识别方式识别得到代码图片中的代码,并对识别得到的代码进行纠错处理;

[0028] 第一处理模块,其用于基于代码格式化工具对代码进行格式化,并采用绘图工具将格式化后的代码制作成代码图片;

[0029] 第二处理模块,其用于对格式化后的代码进行插桩处理,并采用测试用例产生工具对插桩后的代码产生测试用例;

[0030] 执行模块,其用于运行测试用例,并采用特定显示方式对代码运行顺序进行展示,同时展示代码运行阶段的数值信息。

[0031] 与现有技术相比,本发明的优点在于:能够有效帮助用户增进对于代码图片中代码片段的理解,简化了用户实现代码的流程,使得用户不必手动输入并实现代码,简化了用户理解代码的流程,自动产生测试用例带入代码图片中代码片段,自动产生可视化展示代码的运行语句顺序与相对应参数值的变化,具有自动化、低成本、易于理解等特点。

附图说明

[0032] 为了更清楚地说明本申请实施例中的技术方案,下面将对实施例描述中所需要使用的附图作简单地介绍,显而易见地,下面描述中的附图仅仅是本申请的一些实施例,对于本领域普通技术人员来讲,在不付出创造性劳动的前提下,还可以根据这些附图获得其他的附图。

[0033] 图1为本发明实施例中一种代码图片的可视化展示实现方法的流程图。

具体实施方式

[0034] 为使本申请实施例的目的、技术方案和优点更加清楚,下面将结合本申请实施例中的附图,对本申请实施例中的技术方案进行清楚、完整地描述,显然,所描述的实施例是本申请的一部分实施例,而不是全部的实施例。

[0035] 本发明提供一种代码图片的可视化展示实现方法,该方法可基于智能设备实现,结合智能设备帮助用户(如学生、程序员等)快速理解文本中的代码示例,使用者只需使用智能设备对着目标代码进行截取图片,就能对此代码片段以高亮代码行的方式对代码运行过程进行可视化展示,并展示各项参数值实时的变化,具有自动化、低成本、易于理解等特点;以可视化展示程序运行过程的方式增进用户(如学生、程序员等)对于代码图片(如教科书上的代码照片、屏幕代码截图等)中代码片段的理解。

[0036] 参见图1所示,本发明实施例提供的一种代码图片的可视化展示实现方法,具体包括以下步骤:

[0037] S1:基于光学字符识别方式识别得到代码图片中的代码,并对识别得到的代码进行纠错处理;

[0038] 即对于代码图片进行光学字符识别,得到代码的字符串,并通过程序分析技术对识别得到的代码字符串进行纠错,使其可编译。

[0039] 本发明中,对识别得到的代码进行纠错处理,具体为:基于启发式规则对代码进行纠错处理,以将光学字符识别时识别的错误字符替换为正确字符。即基于启发式规则来替换光学字符识别所产生的常见错误,其技术形式是基于静态程序分析的代码纠错,以达到可编译的效果。

[0040] S2:基于代码格式化工具对代码进行格式化,并采用绘图工具将格式化后的代码制作成代码图片;代码图片中,代码间采用等宽字体,以便掌握各行或各语句在图片上的位置方便后续进行高亮可视化展示。

[0041] S3:对格式化后的代码进行插桩处理,并采用测试用例产生工具对插桩后的代码产生测试用例;对代码进行插桩处理,可使得代码在运行时能以字符串的形式打印出相关信息。

[0042] 本发明中,对格式化后的代码进行插桩处理,具体步骤包括:采用源代码分析和转换工具对代码进行遍历,在每个语句前后插入打印该代码位置编号以及其参数值的代码。

[0043] 本发明中,在每个语句前后插入打印该代码位置编号以及其参数值的代码,具体为:

[0044] 对于非循环的所有语句,在语句前插入打印该语句行编号的代码;

[0045] 对于循环语句,则在语句前插入打印该循环开始与结束位置的代码;

[0046] 对于赋值语句、初始值语句,以及输入特定参数名的调用函数语句,则另外在语句后插入打印该值的代码。

[0047] 进一步的,在采用源代码分析和转换工具对代码进行遍历的同时,获取当前代码区块的位置信息;位置信息包括代码起始行与结束字节编号,以及代码结束行与结束字节编号。

[0048] 本发明中,在采用测试用例产生工具对插桩后的代码产生测试用例时,在产生的测试用例中代入源代码的参数。即在使用测试用例产生工具时,为了使可视化理解代码的

效果更好,对产生的测试用例带入源代码的参数进行限制。

[0049] 本发明中,当采用测试用例产生工具对插桩后的代码产生测试用例之后,还包括:采用源代码分析和转换工具遍历测试用例,判断测试用例中被测函数的参数是否完整,以挑选得到具备预设可视化理解效果的测试用例。

[0050] S4:运行测试用例,并采用特定显示方式对代码运行顺序进行展示,同时展示代码运行阶段的数值信息。

[0051] 即在运行测试用例时,通过遍历产生的与之相关信息的字符串,以高亮显示代码方式对代码运行顺序进行展示,同时展示代码运行阶段的数值信息,数值信息包括代码行编号、循环语句位置编号和代码中各项参数的数值。

[0052] 具体的,本发明所述可视化展示实现方法可应用于智能设备(例如智能手机),以智能设备的应用的形式呈现。使用者在阅读文本材料时,若遇到需要理解的代码片段,即可打开智能设备应用,该智能设备应用会使用智能设备自带的摄像头权限,使用者只需要将画面对准代码片段并按截屏按钮就可以看到弹出的代码可视化展现。

[0053] 本发明结合了光学字符识别工具与测试用例生成工具,以代码插桩的方式收集代码信息,最后以可视化的形式展现。整体可以分成四个阶段:光学字符识别与代码补全纠错阶段、遍历代码进行插桩阶段、测试用例生成与选择阶段、输出结果收集与可视化展示阶段。

[0054] 对于光学字符识别与代码补全纠错阶段,首先使用光学字符识别技术将图片中的代码文字转换为字符串,由于光学字符识别技术可能存在精度不足的问题,在代码出现错误的情况下将无法编译,故进行代码纠错,可以使用启发式规则来将显而易见以及重复率高的错误进行修正。

[0055] 对于遍历代码进行插桩阶段,为了在运行代码时得出所需的信息,使用源代码分析和转换工具遍历代码,并在每一行代码之前插入打印该行代码号的语句;在每一个赋值语句、初始化语句与输入特定参数名的调用函数语句之后插入打印该参数的语句;在循环的前后插入标明循环种类与编号及其运行位置的语句。

[0056] 为了作可视化展示,利用代码格式化工具将原代码进行代码格式化,使代码风格统一。将格式化过后的代码使用绘图功能存取成代码图片,以此图片作为可视化展示的基底,并且为了能高亮出代码片段在图片中的位置,在此使用等宽字体。

[0057] 在遍历代码的同时,由于已经遍历了代码,故可以得出当前代码行或者当前代码片段在整个代码中的位置,例如此行在第3行的第5个字节到第25个字节,并根据此位置信息与等宽字体的高度或宽度相乘,以之前所制作的可视化基底图片为基础,制作在基底图片上高亮目标代码片段的图片,并以对应目标代码片段命名,例如高亮第三行代码的图片命名为“3.jpg”。

[0058] 对于测试用例生成与选择阶段,使用测试用例生成工具对插桩后的代码生成测试用例。测试用例生成工具会根据设定的参数生成多个测试用例。由于工具生成的测试用例不一定有助于可视化理解代码,选择更好的测试用例就显得至关重要,在此使用源代码分析和转换工具检查测试用例代码,检查是否真的有待入非空的值,限制数组最小长度等等,若符合条件,则存取当前测试用例作为可视化展示使用,最后,运行通过筛选的测试用例。

[0059] 对于输出结果收集与可视化展示阶段,由于已运行了测试用例,此时所插入的代

码会将所需的信息顺序(包含代码行数、参数数值与循环状态)打印出。将此字符流读入并分解相关信息,即可将之前制作的高亮代码片段图片按照运行顺序高亮显示,进行可视化展示,并且显示运行到该代码片段时所有参数值的状态。

[0060] 以下对本发明代码图片的可视化展示实现方法的具体实施流程进行说明。

[0061] (1)使用光学字符识别工具(例如ABBYY、Tesseract)取得代码图片中的代码;

[0062] (2)对于识别得到的代码进行纠错与代码补全,并把代码片段存成可运行的类,使用启发式规则将常见的光学字符识别错误替换为正确的字词,例如常见的I、l、1的错误问题,将length、loop等词替换为正确的length、loop;

[0063] (3)对代码使用代码格式化工具(例如源代码分析和转换工具Spoon的PrettyPrint功能),并将格式化后的代码使用绘图工具(如java.awt.Graphics.drawString())将代码片段制作成图片,以此图片作为可视化的基础;

[0064] (4)使用源代码分析和转换工具(例如spoon)将步骤(3)所格式化的代码遍历并进行插桩,加入将打印信息输出至文字档(在此存为sequence.txt)的代码。对于插桩部份,重点针对下面三种类型:

[0065] A.在每一行之前插入探针,打印代码行编号,例如out.write("!LINE!"+"#,#");

[0066] B.在每一个赋值语句、初始化语句与已被纪录的参数名被输入至其他函数的语句之后插入探针(包括for语句的initial等等),打印该参数值。例如out.write("p="+ "#,#"),其中若是数组,则将数组的所有值输出,例如out.write("a="+ java.util.Arrays.toString(a)+"#,#");

[0067] C.针对每一个循环种类,在其开始与结束插入探针,以显示目前状态与该循环编号。例如for循环外插入探针使其打印"forA1",例如out.write("forA1#,#")、for循环开始前插入探针使其打印"forB1",例如out.write("forB1#,#")、结束时插入探针,使其打印"forC1",例如out.write("forC1#,#")。

[0068] (5)在遍历代码插桩的同时,取得该代码区块的位置信息,包括该代码起始行与结束字节编号、结束行与结束字节编号。每一次遍历时基于步骤(3)所制作的代码图片,根据等宽字体的宽度与高度相乘,即可以得出该代码块的坐标。利用该坐标信息以绘图方法(例如Java graphics.fillRect()),在代码图片上高亮出该代码块。最后使用Java I/O将此图另存为一图片,其档案名需对应步骤(4)探针所打印的字符。

[0069] (6)对于插桩后的代码,利用测试用例生成工具(例如evosuite)对其生成测试用例。为了提高可视化理解的效果,对于测试用例生成加入条件,例如evosuite指令中增加限制最大integer值不超过10,以及不要产生assertion,并且尽可能多产生一些测试用例。

[0070] (7)对于步骤(6)所产生的测试类,其中一个测试类会包含许多测试用例,使用源代码分析和转换工具对其中的每个测试用例进行遍历,对于每个测试用例,记录赋值语句所赋予的值和参数名,并且对应当前测试用例调用被测函数的语句,若带入的值不为空值、且若带入的值为数组,其长度大于3,则认为此测试用例对可视化理解是有帮助的,即可存取此测试用例单独至一个新的类,依序存为test0.java、test1.java等等。

[0071] (8)将运行步骤(7)所存取的测试用例类的指令储存成批次档,运行该批次档即可

得出打印出信息的文字档(sequence.txt)。使用Java I/O读取该文字流并遍历,并根据步骤(4)插入的探针所打印的值分别作处理。若是代码行或者是循环中的位置,则转换成步骤(5)所生成的对应图片路径,利用排程方法(例如timer())设置每一秒进行图片播放,若是打印的参数值,则将各参数值存储或者更新至一个数据结构(在此使用Map<String, ArrayList<String>>),并且将该数据结构在画面中展示出来。

[0072] 本发明的核心点在于:1、将光学字符识别技术、源代码分析和转换工具与测试用例自动生成工具相结合,以可视化的形式展现帮助用户理解代码图片中的代码片段;2、基于光学字符识别技术和代码纠错的可编译性恢复,基于启发式规则,通过静态程序分析技术纠正光学字符识别技术识别后的代码的识别错误,使得原本不完整或者残缺的代码片段可编译运行;3、基于测试用例生成技术和选取技术的程序执行的可视化理解,构造自动的测试用例生成,并依据限制输入被测代码的参数范围的策略来选取更适合可视化理解的测试用例;4、结合智能设备的应用程序,以动态的高亮执行位置展示显示测试用例所经过的代码区块的方式,结合智能设备自带的摄像头拍摄代码图片,并以高亮代码区块的方式进行展示代码运行顺序,简化了自行运行代码的繁琐步骤。

[0073] 本发明实施例的代码图片的可视化展示实现方法,能够有效帮助用户增进对于代码图片中代码片段的理解,简化了用户实现代码的流程,使得用户不必手动输入并实现代码,简化了用户理解代码的流程,自动产生测试用例带入代码图片中代码片段,自动产生可视化展示代码的运行语句顺序与相对应参数值的变化,具有自动化、低成本、易于理解等特点。

[0074] 在一种可能的实施方式中,本发明实施例还提供一种可读存储介质,可读存储介质位于PLC(Programmable Logic Controller,可编程逻辑控制器)控制器中,可读存储介质上存储有计算机程序,该程序被处理器执行时实现以下所述一种代码图片的可视化展示实现方法的步骤:

[0075] 基于光学字符识别方式识别得到代码图片中的代码,并对识别得到的代码进行纠错处理;

[0076] 基于代码格式化工具对代码进行格式化,并采用绘图工具将格式化后的代码制作成代码图片;

[0077] 对格式化后的代码进行插桩处理,并采用测试用例产生工具对插桩后的代码产生测试用例;

[0078] 运行测试用例,并采用特定显示方式对代码运行顺序进行展示,同时展示代码运行阶段的数值信息。

[0079] 存储介质可以采用一个或多个计算机可读的介质的任意组合。计算机可读介质可以是计算机可读信号介质或者计算机可读存储介质。计算机可读存储介质例如可以是但不限于:电、磁、光、电磁、红外线、或半导体的系统、装置或器件,或者任意以上的组合。计算机可读存储介质的更具体的例子(非穷举的列表)包括:具有一个或多个导线的电连接、便携式计算机磁盘、硬盘、随机存取存储器(RAM)、只读存储器(ROM)、可擦式可编程只读存储器(EPROM或闪存)、光纤、便携式紧凑磁盘只读存储器(CD-ROM)、光存储器件、磁存储器件、或者上述的任意合适的组合。在本文件中,计算机可读存储介质可以是任何包含或存储程序的有形介质,该程序可以被指令执行系统、装置或者器件使用或者与其结合使用。

[0080] 计算机可读的信号介质可以包括在基带中或者作为载波一部分传播的数据信号，其中承载了计算机可读的程序代码。这种传播的数据信号可以采用多种形式，包括但不限于电磁信号、光信号或上述的任意合适的组合。计算机可读的信号介质还可以是计算机可读存储介质以外的任何计算机可读介质，该计算机可读介质可以发送、传播或者传输用于由指令执行系统、装置或者器件使用或者与其结合使用的程序。计算机可读介质上包含的程序代码可以用任何适当的介质传输，包括但不限于：无线、电线、光缆、RF等等，或者上述的任意合适的组合。

[0081] 可以以一种或多种程序设计语言或其组合来编写用于执行本发明操作的计算机程序代码，所述程序设计语言包括面向对象的程序设计语言，诸如Java、Smalltalk、C++，还包括常规的过程式程序设计语言—诸如“C”语言或类似的设计语言。程序代码可以完全地在用户计算机上执行、部分地在用户计算机上执行、作为一个独立的软件包执行、部分在用户计算机上部分在远程计算机上执行、或者完全在远程计算机或服务器上执行。在涉及远程计算机的情形中，远程计算机可以通过任意种类的网络，包括局域网 (LAN) 或广域网 (WAN)，连接到用户计算机，或者，可以连接到外部计算机 (例如利用因特网服务提供商来通过因特网连接)。

[0082] 本发明实施例提供的一种代码图片的可视化展示实现装置，包括识别模块、第一处理模块、第二处理模块和执行模块。

[0083] 识别模块用于基于光学字符识别方式识别得到代码图片中的代码，并对识别得到的代码进行纠错处理；第一处理模块用于基于代码格式化工具对代码进行格式化处理，并采用绘图工具将格式化后的代码制作成代码图片；第二处理模块用于对格式化后的代码进行插桩处理，并采用测试用例产生工具对插桩后的代码产生测试用例；执行模块用于运行测试用例，并采用特定显示方式对代码运行顺序进行展示，同时展示代码运行阶段的数值信息。

[0084] 以上所述仅是本申请的具体实施方式，使本领域技术人员能够理解或实现本申请。对这些实施例的多种修改对本领域的技术人员来说将是显而易见的，本文中所定义的一般原理可以在不脱离本申请的精神或范围的情况下，在其它实施例中实现。因此，本申请将不会被限制于本文所示的这些实施例，而是要符合与本文所申请的原理和新颖特点相一致的最宽的范围。

[0085] 本发明是参照根据本发明实施例的方法、设备(系统)和计算机程序产品的流程图和/或方框图来描述的。应理解可由计算机程序指令实现流程图和/或方框图中的每一流程和/或方框、以及流程图和/或方框图中的流程和/或方框的结合。可提供这些计算机程序指令到通用计算机、专用计算机、嵌入式处理机或其他可编程数据处理设备的处理器以产生一个机器，使得通过计算机或其他可编程数据处理设备的处理器执行的指令产生用于实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的装置。

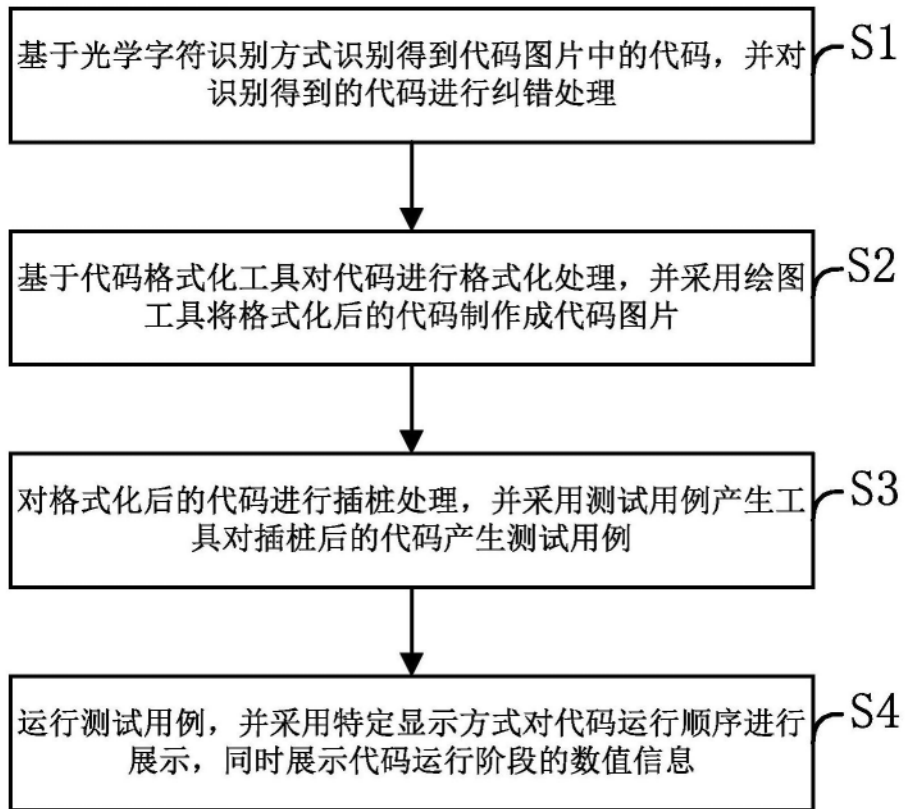


图1