



(12)发明专利申请

(10)申请公布号 CN 110688103 A

(43)申请公布日 2020.01.14

(21)申请号 201910959952.9

(22)申请日 2019.10.10

(71)申请人 北京东软望海科技有限公司

地址 100176 北京市大兴区北京经济技术
开发区宏达北路12号B楼二区夹07室

(72)发明人 梁国华

(74)专利代理机构 北京市立方律师事务所

11330

代理人 张筱宁

(51)Int.Cl.

G06F 8/33(2018.01)

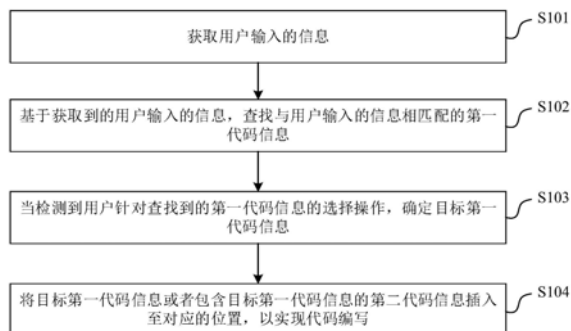
权利要求书2页 说明书9页 附图4页

(54)发明名称

代码编写方法、装置、电子设备及计算机可读存储介质

(57)摘要

本申请实施例提供了一种代码编写方法、装置、电子设备及计算机可读存储介质。该方法包括：获取用户输入的信息，基于获取到的用户输入的信息，查找与用户输入的信息相匹配的第一代码信息，当检测到用户针对查找到的第一代码信息的选择操作，确定目标第一代码信息，将目标第一代码信息或者包含目标第一代码信息的第二代码信息插入至对应的位置，以实现代码编写。本申请实施例实现了一方面减少了手动代码编写耗费的时间和人力成本；另一方面降低复制粘贴编程人员之前已编写的代码的可能性限制，从而减少复制粘贴的耗费时间并降低了复制粘贴的复杂度，提升了用户的体验。



1. 一种代码编写方法,其特征在于,包括:
获取用户输入的信息;
基于获取到的用户输入的信息,查找与所述用户输入的信息相匹配的第一代码信息;
当检测到用户针对查找到的第一代码信息的选择操作,确定目标第一代码信息;
将所述目标第一代码信息或者包含所述目标第一代码信息的第二代码信息插入至对应的位置,以实现代码编写。
2. 根据权利要求1所述的方法,其特征在于,所述确定目标第一代码信息,之前还包括:
基于查找到的与所述用户输入的信息相匹配的第一代码信息,查找包含第一代码信息的第二代码信息。
3. 根据权利要求1所述的方法,其特征在于,所述确定目标第一代码信息,之后还包括:
查找包含所述目标第一代码信息的第二代码信息。
4. 根据权利要求1-3任一项所述的方法,其特征在于,若查找到与所述用户输入的信息相匹配的第一代码信息为至少两个,
所述查找与所述用户输入的信息相匹配的第一代码信息,之后还包括:
按照命中次数,确定查找到的各个第一代码信息分别对应的排序信息;
按照所述查找到的各个第一代码信息分别对应的排序信息,显示所述各个第一代码信息;
其中,所述命中次数为用户的选择操作选择各个第一代码信息的次数。
5. 根据权利要求1-4任一项所述的方法,其特征在于,所述将所述目标第一代码信息或者包含所述目标第一代码信息的第二代码信息插入至对应的位置,以实现代码编写的步骤,包括:
通过预设方式,将所述目标第一代码信息或者包含所述目标第一代码信息的第二代码信息插入至对应的位置,以实现代码编写;
其中,预设方式包括以下至少一项:
将所述目标第一代码信息中除所述用户输入的信息之外的内容或者包含所述目标第一代码信息的第二代码信息中除所述用户输入的信息之外的内容插入至所述用户输入的信息之后;
将所述目标第一代码信息或者包含所述目标第一代码信息的第二代码信息替换所述用户输入的信息。
6. 根据权利要求1-5任一项所述的方法,其特征在于,所述当检测到用户针对查找到的第一代码信息的选择操作,确定目标第一代码信息,之后还包括:
更新所述目标第一代码信息的命中次数。
7. 根据权利要求1-6任一项所述的方法,其特征在于,若查找到的与所述用户输入的信息相匹配第一代码信息包含至少一个,且至少一个第一代码信息中存在至少一个满足预设条件的第一代码信息,该满足预设条件的第一代码信息存在第二代码信息;
所述基于获取到的用户输入的信息,查找与所述用户输入的信息相匹配的第一代码信息,之后还包括:
确定每个第二代码信息对应的功能信息,所述每个第二代码信息对应的功能信息用于表征所述每个第二代码信息所实现的功能;

当检测到用户针对功能信息的选择操作,将选择操作选择的功能信息对应的第二代码信息插入至对应的位置。

8.一种代码编写装置,其特征在于,包括:

获取模块,用于获取用户输入的信息;

第一查找模块,用于基于获取到的用户输入的信息,查找与所述用户输入的信息相匹配的第一代码信息;

第一确定模块,用于当检测到用户针对查找到的第一代码信息的选择操作,确定目标第一代码信息;

第一插入模块,用于将所述目标第一代码信息或者包含所述目标第一代码信息的第二代码信息插入至对应的位置,以实现代码编写。

9.一种电子设备,其特征在于,其包括:

一个或多个处理器;

存储器;

一个或多个应用程序,其中所述一个或多个应用程序被存储在所述存储器中并被配置为由所述一个或多个处理器执行,所述一个或多个程序配置用于:执行根据权利要求1~7任一项所述的代码编写方法。

10.一种计算机可读存储介质,其上存储有计算机程序,其特征在于,该程序被处理器执行时实现权利要求1~7任一项所述的代码编写方法。

代码编写方法、装置、电子设备及计算机可读存储介质

技术领域

[0001] 本申请涉及软件开发技术领域,具体而言,本申请涉及一种代码编写方法、装置、电子设备及计算机可读存储介质。

背景技术

[0002] 随着互联网的不断发展,信息化已经融入我们生活的方方面面,越来越多的人参与到信息技术行业中,参与到编程工作中,编程人员在对应的开发工具上通过编写代码来实现某些功能。

[0003] 现有技术中编程人员还是通过手动编写代码来实现具体的功能,或者编程人员查找自己之前已编写的代码中是否存在实现相同功能的代码,若存在,可以复制粘贴到对应的位置。

[0004] 但是上述方案中通过手动编写代码需要耗费编程人员大量的时间,耗费大量的人力成本;再者由于编程人员之前已编写过实现相同功能的代码可能性较低,因此通过复制粘贴编程人员之前已编写的代码的可能性同样较低,再者即使编程人员之前编写过实现相同功能的代码,也还需要编程人员查找具体的代码位置,然后在进行复制粘贴,耗费时间同样较长,复杂度同样较高,进一步导致用户体验较差。

发明内容

[0005] 本申请提供了一种代码编写方法、装置、电子设备及计算机可读存储介质,可以解决以上至少一种技术问题。

[0006] 第一方面,提供了一种代码编写的方法,该方法包括:

[0007] 获取用户输入的信息;

[0008] 基于获取到的用户输入的信息,查找与用户输入的信息相匹配的第一代码信息;

[0009] 当检测到用户针对查找到的第一代码信息的选择操作,确定目标第一代码信息;

[0010] 将目标第一代码信息或者包含目标第一代码信息的第二代码信息插入至对应的位置,以实现代码编写。

[0011] 第二方面,提供了一种代码编写装置,该装置包括:

[0012] 获取模块,用于获取用户输入的信息;

[0013] 第一查找模块,用于基于获取到的用户输入的信息,查找与用户输入的信息相匹配的第一代码信息;

[0014] 第一确定模块,用于当检测到用户针对查找到的第一代码信息的选择操作,确定目标第一代码信息;

[0015] 第一插入模块,用于将目标第一代码信息或者包含目标第一代码信息的第二代码信息插入至对应的位置,以实现代码编写。

[0016] 第三方面,提供了一种电子设备,该电子设备包括:

[0017] 一个或多个处理器;

[0018] 存储器；

[0019] 一个或多个应用程序，其中一个或多个应用程序被存储在存储器中并被配置为由一个或多个处理器执行，一个或多个程序配置用于：执行根据第一方面所示的代码编写方法对应的操作。

[0020] 第四方面，提供了一种计算机可读存储介质，其上存储有计算机程序，该程序被处理器执行时实现第一方面所示的代码编写方法。

[0021] 本申请提供的技术方案带来的有益效果是：

[0022] 本申请提供了一种代码编写方法、装置、电子设备及计算机可读存储介质，与现有技术相比，本申请通过获取用户输入的信息，基于获取到的用户输入的信息，查找与用户输入的信息相匹配的第一代码信息，当检测到用户针对查找到的第一代码信息的选择操作，确定目标第一代码信息，将目标第一代码信息或者包含目标第一代码信息的第二代码信息插入至对应的位置，以实现代码编写，一方面减少了手动代码编写耗费的时间和人力成本；另一方面降低复制粘贴编程人员之前已编写的代码的可能性限制，从而减少复制粘贴的耗费时间并降低了复制粘贴的复杂度，提升了用户的体验。

附图说明

[0023] 为了更清楚地说明本申请实施例中的技术方案，下面将对本申请实施例描述中所需要使用的附图作简单地介绍。

[0024] 图1为本申请实施例提供的一种代码编写方法的流程示意图；

[0025] 图2为本申请实施例提供的一种代码编写装置的结构示意图；

[0026] 图3为本申请实施例提供的一种电子设备的结构示意图；

[0027] 图4为本申请实施例提供的一种代码编写方法的显示界面图；

[0028] 图5为本申请实施例提供的另一种代码编写方法的显示界面图；

[0029] 图6为本申请实施例提供的一种Eclipse与“代码提示助手”之间的交互结构示意图。

具体实施方式

[0030] 下面详细描述本申请的实施例，所述实施例的示例在附图中示出，其中自始至终相同或类似的标号表示相同或类似的元件或具有相同或类似功能的元件。下面通过参考附图描述的实施例是示例性的，仅用于解释本申请，而不能解释为对本申请的限制。

[0031] 本技术领域技术人员可以理解，除非特意声明，这里使用的单数形式“一”、“一个”、“所述”和“该”也可包括复数形式。应该进一步理解的是，本申请的说明书中使用的措辞“包括”是指存在所述特征、整数、步骤、操作、元件和/或组件，但是并不排除存在或添加一个或多个其他特征、整数、步骤、操作、元件、组件和/或它们的组。应该理解，当我们称元件被“连接”或“耦接”到另一元件时，它可以直接连接或耦接到其他元件，或者也可以存在中间元件。此外，这里使用的“连接”或“耦接”可以包括无线连接或无线耦接。这里使用的措辞“和/或”包括一个或更多个相关联的列出项的全部或任一单元和全部组合。

[0032] 为使本申请的目的、技术方案和优点更加清楚，下面将结合附图对本申请实施方式作进一步地详细描述。

[0033] 下面以具体地实施例对本申请的技术方案以及本申请的技术方案如何解决上述技术问题进行详细说明。下面这几个具体的实施例可以相互结合,对于相同或相似的概念或过程可能在某些实施例中不再赘述。下面将结合附图,对本申请的实施例进行描述。

[0034] 本申请实施例提供了一种代码编写方法,如图1所示,该方法包括:

[0035] 步骤S101,获取用户输入的信息。

[0036] 对于本申请实施例,获取用户在预设输入界面输入的信息。

[0037] 步骤S102,基于获取到的用户输入的信息,查找与用户输入的信息相匹配的第一代码信息。

[0038] 对于本申请实施例,第一代码信息可以为字段信息,也可以为类名称信息,还可以为方法名称信息,如“SystemException”或者“Filer”等,在本申请实施例中不做限定。

[0039] 步骤S103,当检测到用户针对查找到的第一代码信息的选择操作,确定目标第一代码信息。

[0040] 步骤S104,将目标第一代码信息或者包含目标第一代码信息的第二代码信息插入至对应的位置,以实现代码编写。

[0041] 对于本申请实施例,第二代码信息为包含第一代码信息的实现预设功能的功能代码块。

[0042] 本申请实施例提供了一种代码编写方法,与现有技术相比,本申请实施例通过获取用户输入的信息,基于获取到的用户输入的信息,查找与用户输入的信息相匹配的第一代码信息,当检测到用户针对查找到的第一代码信息的选择操作,确定目标第一代码信息,将目标第一代码信息或者包含目标第一代码信息的第二代码信息插入至对应的位置,以实现代码编写,一方面减少了手动代码编写耗费的时间和人力成本;另一方面降低复制粘贴编程人员之前已编写的代码的可能性限制,从而减少复制粘贴的耗费时间并降低了复制粘贴的复杂度,提升了用户的体验。

[0043] 本申请实施例的另一种可能实现方式,步骤S103之前还可以包括:基于查找到的与用户输入的信息相匹配的第一代码信息,查找包含第一代码信息的第二代码信息,一个第二代码信息中包含一个第一代码信息。

[0044] 对于本申请实施例,与用户输入的信息相匹配的第一代码信息包含至少一个,针对任一个第一代码信息,查找包含任一个第一代码信息的第二代码信息。

[0045] 例如,用户输入的信息为“FileR”,基于查找到的与用户输入的信息相匹配的第一代码信息分别为“FileReader”、“FileReadEvent”以及“FileResolverImpl”,查找包含第一代码信息“FileReader”的第二代码信息、查找包含第一代码信息“FileReadEvent”的第二代码信息,以及查找包含第一代码信息“FileResolverImpl”的第二代码信息。

[0046] 进一步地,当检测到用户针对查找到的第一代码信息的选择操作,确定目标第一代码信息,如目标第一代码信息为“FileReader”。

[0047] 本申请实施例的另一种可能实现方式,步骤S103之后还可以包括:查找包含目标第一代码信息的第二代码信息。

[0048] 对于本申请实施例,与用户输入的信息相匹配的第一代码信息包含至少一个,当检测到用户针对查找到的至少一个第一代码信息的选择操作,确定目标第一代码信息,并查找包含目标第一代码信息的第二代码信息。

[0049] 例如,如图4所示,用户输入的信息为A1:“FileR”,基于查找到的与用户输入的信息相匹配的第一代码信息为A3,具体地,A3包括:“FileReader”、“Filer”、“FileReadEvent”、“FileResolverImpl”以及“FilerException”,当检测到用户针对查找到的第一代码信息的选择操作,确定目标第一代码信息为“FileReader”时,查找包含目标第一代码信息“FileReader”的第二代码信息,如图5所示,包含目标第一代码信息“FileReader”的第二代码信息为B。

[0050] 本申请实施例的另一种可能实现方式,若查找到与用户输入的信息相匹配的第一代码信息为至少两个,所述步骤S102,之后还可以包括:按照命中次数,确定查找到的各个第一代码信息分别对应的排序信息;按照查找到的各个第一代码信息分别对应的排序信息,显示各个第一代码信息。

[0051] 其中,命中次数为用户的选择操作选择各个第一代码信息的次数。

[0052] 对于本申请实施例,按照命中次数和/或预设排序规则,确定查找到的各个第一代码信息分别对应的排序信息,按照查找到的各个第一代码信息分别对应的排序信息,显示各个第一代码信息,能够优先显示开发人员常用的第一代码信息,利于开发人员选择,减少时间的浪费。其中,预设排序规则可以为第一代码信息的长度,也可以为字母顺序,在本申请实施例中不做限定。

[0053] 例如,第一代码信息分别为“FileReader”、“FileReadEvent”以及“FileResolverImpl”,其中,“FileReader”的命中次数为2,“FileReadEvent”和“FileResolverImpl”的命中次数为5,若排序按照优先命中次数,其次为第一代码信息的长度的规则,“FileReader”的排序信息为第三位,“FileReadEvent”的排序信息为第二位,“FileResolverImpl”的排序信息为第一位,即,三个第一代码信息的排序为:“FileResolverImpl”、“FileReadEvent”和“FileReader”。

[0054] 本申请实施例的另一种可能实现方式,步骤S104具体可以包括以下任一项:若存在包含目标第一代码信息的第二代码信息,则将包含目标第一代码信息的第二代码信息插入至对应的位置;若不存在包含目标第一代码信息的第二代码信息,则将目标第一代码信息插入至对应的位置。

[0055] 例如,目标第一代码信息为“FileReader”时,若存在包含“FileReader”的第二代码信息,则将包含“FileReader”的第二代码信息插入至对应的位置;若不存在包含“FileReader”的第二代码信息,则将“FileReader”插入至对应的位置。

[0056] 本申请实施例的另一种可能实现方式,步骤S104具体可以包括:通过预设方式,将目标第一代码信息或者包含目标第一代码信息的第二代码信息插入至对应的位置,以实现代码编写。

[0057] 其中,预设方式包括以下至少一项:将目标第一代码信息中除用户输入的信息之外的内容或者包含目标第一代码信息的第二代码信息中除用户输入的信息之外的内容插入至用户输入的信息之后;将目标第一代码信息或者包含目标第一代码信息的第二代码信息替换用户输入的信息。

[0058] 例如,用户输入的信息为“FileR”,若目标第一代码信息为“FileReader”,则将目标第一代码信息中除“FileR”之外的内容,即“eader”,或者包含目标第一代码信息“FileReader”的第二代码信息中除“FileR”之外的内容,插入至“FileR”之后。

[0059] 又如,用户输入的信息为“FileR”,若目标第一代码信息为“FileReader”,则用目标第一代码信息,即“FileReader”,或者包含目标第一代码信息“FileReader”的第二代码信息,替换“FileR”。

[0060] 本申请实施例的另一种可能实现方式,步骤S103之后还可以包括:更新目标第一代码信息的命中次数。

[0061] 对于本申请实施例,目标第一代码信息的命中次数的更新方式为:对目标第一代码信息的命中次数增加1。

[0062] 本申请实施例的另一种可能实现方式,代码编写方法还可以包括:存储用户输入的多个第二代码信息以及各个第二代码信息分别对应的第一代码信息,并创建各个第一代码信息与各个第二代码信息之间的对应关系。

[0063] 对于本申请实施例,将用户输入的多个第二代码信息存储至第一存储区域,将各个第二代码信息分别对应的第一代码信息存储至第二存储区域。其中,第一存储区域可以与第二存储区域为同一个存储区域,也可以为两个不同的存储区域,在本申请实施例中不做限定。进一步地,用户可将第一存储区域存储的多个第二代码信息、第二存储区域存储的与各个第二代码信息分别对应的第一代码信息,以及各个第一代码信息与各个第二代码信息之间的对应关系以预设导出方式导出至其他用户对应的终端设备中。

[0064] 对于本申请实施例,各个第一代码信息与各个第二代码信息之间的对应关系可以为索引关系,以利用预设搜索引擎基于该索引关系查找包含第一代码信息的第二代码信息,或者查找包含目标第一代码信息的第二代码信息,还可以为映射关系等,在本申请实施例中不做限定。

[0065] 对于本申请实施例,存储用户输入的多个第二代码信息以及各个第二代码信息分别对应的第一代码信息,并创建各个第一代码信息与各个第二代码信息之间的对应关系,可以在代码编写方法之前执行,也可以在查找包含第一代码信息的第二代码信息之前执行,还可以在查找包含目标第一代码信息的第二代码信息之前执行,在本申请实施例中不做限定。

[0066] 本申请实施例的另一种可能实现方式,若查找到的与用户输入的信息相匹配第一代码信息包含至少一个,且至少一个第一代码信息中存在至少一个满足预设条件的第一代码信息,该满足预设条件的第一代码信息存在第二代码信息,步骤S102之后具体还可以包括:确定每个第二代码信息对应的功能信息,当检测到用户针对功能信息的选择操作,将选择操作选择的的功能信息对应的第二代码信息插入至对应的位置。

[0067] 其中,每个第二代码信息对应的功能信息用于表征每个第二代码信息所实现的功能。

[0068] 对于本申请实施例,满足预设条件的第一代码信息即为存在至少一个第一代码信息,该至少一个第一代码信息存在其对应的第二代码信息。如,第一代码信息“system”存在其对应的第二代码信息“system{a=1;}”,则第一代码信息“system”即为满足预设条件的第一代码信息。

[0069] 例如,如图4所示,用户输入的信息为A1:“FileR”,与A1相匹配的第一代码信息为A3,其中,A3包括五个第一代码信息,其中三个第一代码信息分别存在对应的第二代码信息,则确定三个第二代码信息分别对应的功能信息A2,具体地,A2包括“读文件”、“写文件”

以及“读写文件”，当检测到用户针对“读文件”的选择操作，将选择操作选择的“读文件”对应的第二代码信息插入至对应的位置，如图5所示，选择操作选择的“读文件”对应的第二代码信息为B。

[0070] 对于本申请实施例，将选择操作选择的功能信息对应的第二代码信息插入至对应的位置的方式为：将选择操作选择的功能信息对应的第二代码信息中除用户输入的信息之外的信息插入至用户输入的信息之后，或者，用选择操作选择的功能信息对应的第二代码信息替换用户输入的信息。

[0071] 对于本申请实施例，上述实施例可以由终端设备执行，也可以由服务器执行，也可以部分由终端设备执行，部分由服务器执行。在本申请实施例中不做限定。

[0072] 上述从方法步骤的角度具体阐述了代码编写方法，下面结合具体的应用场景介绍代码编写方法。

[0073] 如图6所示，本申请实施例提供了一种开发工具Eclipse，其包括的单元为：代码编辑器、代码提示框和代码块信息库；本申请实施例还提供了一种可以在Eclipse中运行的插件工具“代码提示助手”，其中，“代码提示助手”包括的单元为：代码输入监控、代码分析器、代码搜索引擎、本地工程、代码块缓存、代码信息交互、代码块管理界面以及代码块同步。

[0074] 对于本申请实施例，代码编辑器用于获取用户输入的信息；代码输入监控用于监控代码编辑器获取到的用户输入的信息，并调用代码分析器；代码分析器可以用于调用代码搜索引擎从本地工程中查找与用户输入的信息相匹配的第一代码信息，以及从代码块缓存中查找包含第一代码信息的第二代码信息，并将第一代码信息以及包含第一代码信息的第二代码信息发送至代码提示框，以使用户从第一代码信息中选择目标第一代码信息，代码分析器也可以调用代码搜索引擎从本地工程中查找与用户输入的信息相匹配的第一代码信息并发送至代码提示框，以使用户从第一代码信息中选择目标第一代码信息，并调用代码搜索引擎从代码块缓存中查找包含目标第一代码信息的第二代码信息并发送至代码提示框；代码提示框用于将目标第一代码信息或者包含目标第一代码信息的第二代码信息插入至对应的位置，以实现代码编写。

[0075] 对于本申请实施例，代码块管理界面可以接收用户输入的第一代码信息及其对应的第二代码信息；代码块同步可以从其他设备中获取第一代码信息及其对应的第二代码信息；代码信息交互可以建立第一代码信息与第二代码信息之间的对应关系，并保存至代码块信息库或者代码块缓存中。在本申请实施例中，代码分析器还可以通过代码信息交互从代码块信息库中获取包含第一代码信息的第二代码信息，或者包含目标第一代码信息的第二代码信息。

[0076] 上述详细介绍了本申请方法实施例，下面从虚拟模块或者虚拟单元的角度介绍代码编写装置，具体如下所示：

[0077] 本申请实施例提供了一种代码编写装置，如图2所示，该代码编写装置20可以包括：获取模块201、第一查找模块202、第一确定模块203以及第一插入模块204，其中，

[0078] 获取模块201，用于获取用户输入的信息。

[0079] 第一查找模块202，用于基于获取到的用户输入的信息，查找与用户输入的信息相匹配的第一代码信息。

[0080] 第一确定模块203，用于当检测到用户针对查找到的第一代码信息的选择操作，确

定目标第一代码信息。

[0081] 第一插入模块204,用于将目标第一代码信息或者包含目标第一代码信息的第二代码信息插入至对应的位置,以实现代码编写。

[0082] 本申请实施例的另一种可能实现方式,代码编写装置20还可以包括第二查找模块,其中,

[0083] 第二查找模块,用于基于查找到的与用户输入的信息相匹配的第一代码信息,查找包含第一代码信息的第二代码信息。

[0084] 本申请实施例的另一种可能实现方式,代码编写装置20还可以包括第三查找模块,其中,

[0085] 第三查找模块,用于查找包含目标第一代码信息的第二代码信息。

[0086] 本申请实施例的另一种可能实现方式,若查找到与用户输入的信息相匹配的第一代码信息为至少两个,代码编写装置20还可以包括第二确定模块和显示模块,其中,

[0087] 第二确定模块,用于按照命中次数,确定查找到的各个第一代码信息分别对应的排序信息。

[0088] 显示模块,用于按照查找到的各个第一代码信息分别对应的排序信息,显示各个第一代码信息。

[0089] 其中,命中次数为用户的选择操作选择各个第一代码信息的次数。

[0090] 本申请实施例的另一种可能实现方式,第一插入模块204包括第一插入单元和第二插入单元中的任一项,其中,

[0091] 第一插入单元,用于若存在包含目标第一代码信息的第二代码信息,则将包含目标第一代码信息的第二代码信息插入至对应的位置。

[0092] 第二插入单元,用于若不存在包含目标第一代码信息的第二代码信息,则将目标第一代码信息插入至对应的位置。

[0093] 本申请实施例的另一种可能实现方式,第一插入模块204还包括第三插入单元,其中,

[0094] 第三插入单元,用于通过预设方式,将目标第一代码信息或者包含目标第一代码信息的第二代码信息插入至对应的位置,以实现代码编写;

[0095] 其中,预设方式包括以下至少一项:

[0096] 将目标第一代码信息中除用户输入的信息之外的内容或者包含目标第一代码信息的第二代码信息中除用户输入的信息之外的内容插入至用户输入的信息之后。

[0097] 将目标第一代码信息或者包含目标第一代码信息的第二代码信息替换用户输入的信息。

[0098] 本申请实施例的另一种可能实现方式,代码编写装置20还包括更新模块,其中,

[0099] 更新模块,用于更新目标第一代码信息的命中次数。

[0100] 本申请实施例的另一种可能实现方式,代码编写装置20还包括存储创建模块,其中,

[0101] 存储创建模块,用于存储用户输入的多个第二代码信息以及各个第二代码信息分别对应的第一代码信息,并创建各个第一代码信息与各个第二代码信息之间的对应关系。

[0102] 本申请实施例的另一种可能实现方式,若查找到的与用户输入的信息相匹配第一

代码信息包含至少一个,且至少一个第一代码信息中存在至少一个满足预设条件的第一代码信息,该满足预设条件的第一代码信息存在第二代码信息,代码编写装置20还包括第三确定模块和第二插入模块,其中,

[0103] 第三确定模块,用于确定每个第二代码信息对应的功能信息,每个第二代码信息对应的功能信息用于表征每个第二代码信息所实现的功能。

[0104] 第二插入模块,用于当检测到用户针对功能信息的选择操作,将选择操作选择的的功能信息对应的第二代码信息插入至对应的位置。

[0105] 对于本申请实施例,第一查找模块202、第二查找模块以及第三查收模块可以为同一个查找模块,也可以任两个为相同的查找模块,也可以为三个不同的查找模块;第一确定模块203、第二确定模块以及第三确定模块可以为同一个确定模块,也可以任两个为相同的确定模块,也可以为三个不同的确定模块;第一插入模块204和第二插入模块可以为同一个插入模块,也可以为两个不同的插入模块;第一插入单元、第二插入单元以及第三插入单元可以为同一个插入单元,也可以任两个为相同的插入单元,也可以为三个不同的插入单元,在本申请实施例中不做限定。

[0106] 本申请实施例的代码编写装置20可以执行本申请方法实施例的代码插入方法对应的操作,其实现原理类似,在此不再赘述。

[0107] 本申请实施例提供了一种代码编写装置,与现有技术相比,本申请实施例通过获取用户输入的信息,基于获取到的用户输入的信息,查找与用户输入的信息相匹配的第一代码信息,当检测到用户针对查找到的第一代码信息的选择操作,确定目标第一代码信息,将目标第一代码信息或者包含目标第一代码信息的第二代码信息插入至对应的位置,以实现代码编写,一方面减少了手动代码编写耗费的时间和人力成本;另一方面降低复制粘贴编程人员之前已编写的代码的可能性限制,从而减少复制粘贴的耗费时间并降低了复制粘贴的复杂度,提升了用户的体验。

[0108] 上述从虚拟模块或者虚拟单元的角度介绍本申请的代码编写装置,下面从实体装置的角度介绍本申请的电子设备。

[0109] 本申请实施例提供了一种电子设备,如图3所示,图3所示的电子设备4000包括:处理器4001和存储器4003。其中,处理器4001和存储器4003相连,如通过总线4002相连。可选地,电子设备4000还可以包括收发器4004。需要说明的是,实际应用中收发器4004不限于一个,该电子设备4000的结构并不构成对本申请实施例的限定。

[0110] 处理器4001可以是CPU,通用处理器,DSP,ASIC,FPGA或者其他可编程逻辑器件、晶体管逻辑器件、硬件部件或者其任意组合。其可以实现或执行结合本申请公开内容所描述的各种示例性的逻辑方框,模块和电路。处理器4001也可以是实现计算功能的组合,例如包含一个或多个微处理器组合,DSP和微处理器的组合等。

[0111] 总线4002可包括一通路,在上述组件之间传送信息。总线4002可以是PCI总线或EISA总线等。总线4002可以分为地址总线、数据总线、控制总线等。为便于表示,图3中仅用一条粗线表示,但并不表示仅有一根总线或一种类型的总线。

[0112] 存储器4003可以是ROM或可存储静态信息和指令的其他类型的静态存储设备,RAM或者可存储信息和指令的其他类型的动态存储设备,也可以是EEPROM、CD-ROM或其他光盘存储、光碟存储(包括压缩光碟、激光碟、光碟、数字通用光碟、蓝光光碟等)、磁盘存储介质

或者其他磁存储设备、或者能够用于携带或存储具有指令或数据结构形式的期望的程序代码并能够由计算机存取的任何其他介质,但不限于此。

[0113] 存储器4003用于存储执行本申请方案的应用程序代码,并由处理器4001来控制执行。处理器4001用于执行存储器4003中存储的应用程序代码,以实现前述任一方法实施例所示的内容。

[0114] 本申请实施例提供了一种电子设备,本申请实施例中的电子设备包括:存储器和处理器;至少一个程序,存储于所述存储器中,用于被所述处理器执行时,与现有技术相比可实现:本申请实施例通过获取用户输入的信息,基于获取到的用户输入的信息,查找与用户输入的信息相匹配的第一代码信息,当检测到用户针对查找到的第一代码信息的选择操作,确定目标第一代码信息,将目标第一代码信息或者包含目标第一代码信息的第二代码信息插入至对应的位置,以实现代码编写,一方面减少了手动代码编写耗费的时间和人力成本;另一方面降低复制粘贴编程人员之前已编写的代码的可能性限制,从而减少复制粘贴的耗费时间并降低了复制粘贴的复杂度,提升了用户的体验。

[0115] 上述从实体装置的角度介绍本申请的电子设备,下面从存储介质的角度介绍本申请的计算机可读存储介质。

[0116] 本申请实施例提供了一种计算机可读存储介质,该计算机可读存储介质上存储有计算机程序,当其在计算机上运行时,使得计算机可以执行前述方法实施例中相应内容。与现有技术相比,本申请实施例通过获取用户输入的信息,基于获取到的用户输入的信息,查找与用户输入的信息相匹配的第一代码信息,当检测到用户针对查找到的第一代码信息的选择操作,确定目标第一代码信息,将目标第一代码信息或者包含目标第一代码信息的第二代码信息插入至对应的位置,以实现代码编写,一方面减少了手动代码编写耗费的时间和人力成本;另一方面降低复制粘贴编程人员之前已编写的代码的可能性限制,从而减少复制粘贴的耗费时间并降低了复制粘贴的复杂度,提升了用户的体验。

[0117] 应该理解的是,虽然附图的流程图中的各个步骤按照箭头的指示依次显示,但是这些步骤并不是必然按照箭头指示的顺序依次执行。除非本文中有明确的说明,这些步骤的执行并没有严格的顺序限制,其可以以其他的顺序执行。而且,附图的流程图中的至少一部分步骤可以包括多个子步骤或者多个阶段,这些子步骤或者阶段并不必然是在同一时刻执行完成,而是可以在不同的时刻执行,其执行顺序也不必然是依次进行,而是可以与其他步骤或者其他步骤的子步骤或者阶段的至少一部分轮流或者交替地执行。

[0118] 以上所述仅是本申请的部分实施方式,应当指出,对于本技术领域的普通技术人员来说,在不脱离本申请原理的前提下,还可以做出若干改进和润饰,这些改进和润饰也应视为本申请的保护范围。

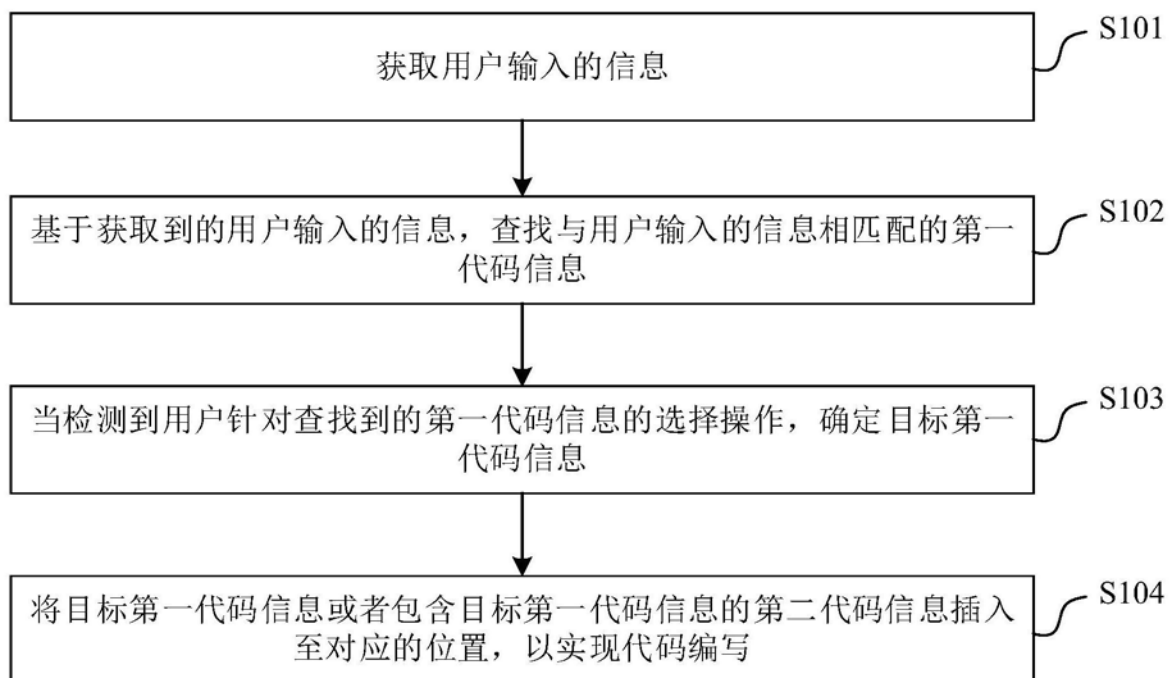


图1

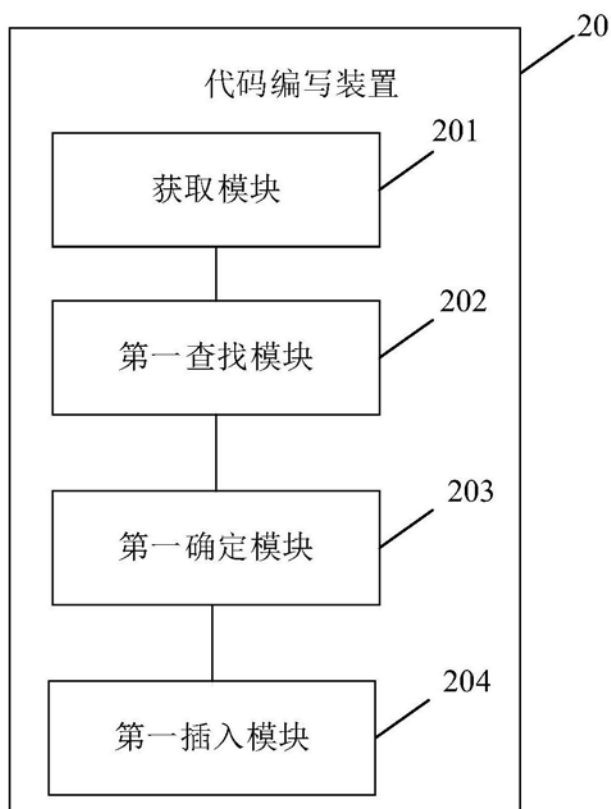


图2

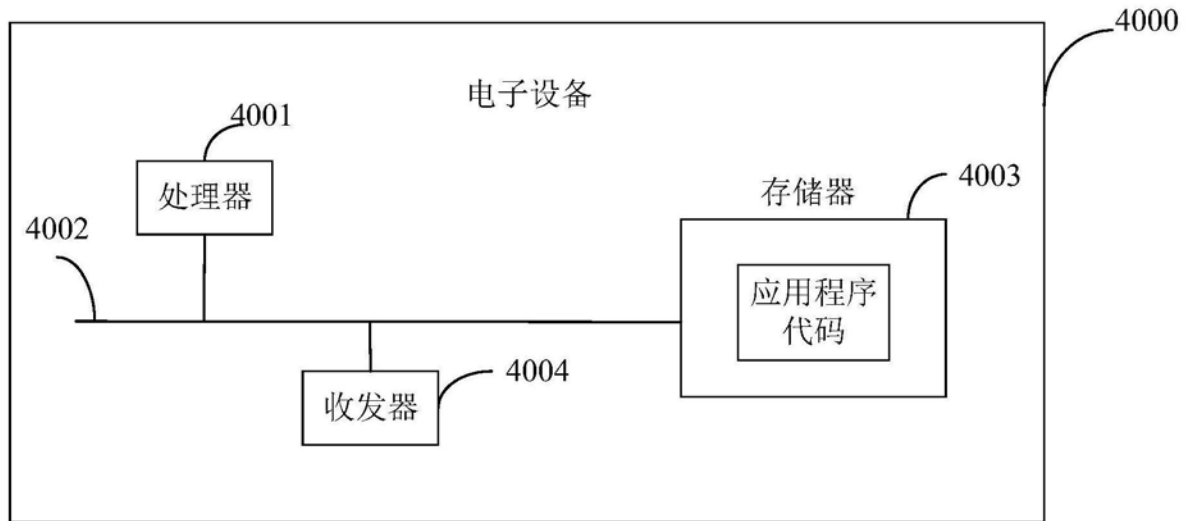


图3

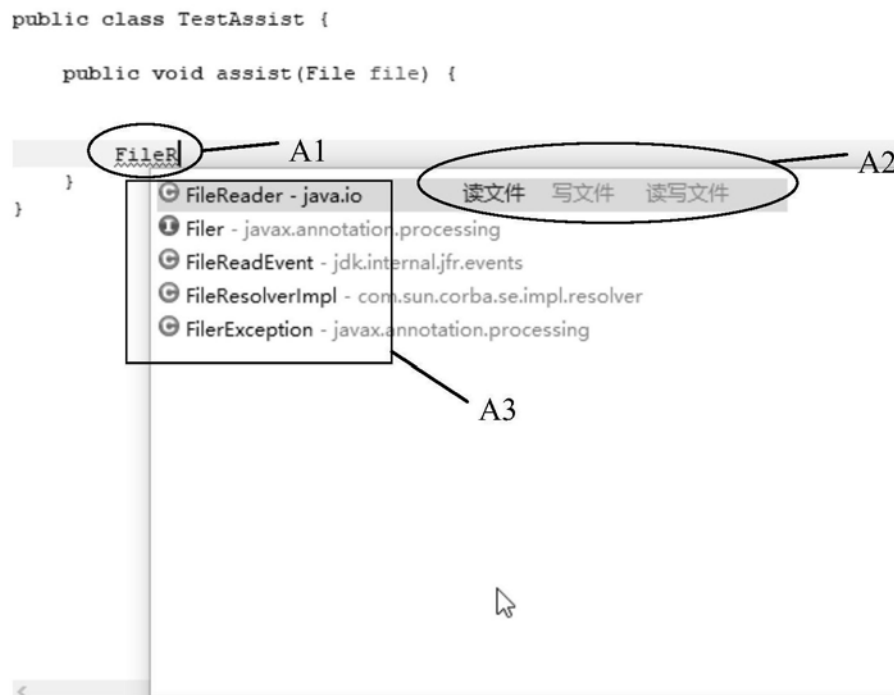


图4

```
public class TestAssist {  
    public void assist(File file) {  
  
        FileReader fileReader = null;  
        BufferedReader bufferedReader = null;  
        String line = null;  
        StringBuffer fileContentBuffer = new StringBuffer();  
        try {  
            fileReader = new FileReader(file);  
            bufferedReader = new BufferedReader(fileReader);  
            while ((line = bufferedReader.readLine()) != null) {  
                fileContentBuffer.append(line);  
                fileContentBuffer.append("\n");  
            }  
        } catch (Exception e) {  
            e.printStackTrace();  
        } finally {  
            try {  
                if (fileReader != null) {  
                    fileReader.close();  
                }  
                if (bufferedReader != null) {  
                    bufferedReader.close();  
                }  
            } catch (Exception e) {  
                e.printStackTrace();  
            }  
        }  
    }  
}
```

B

图5

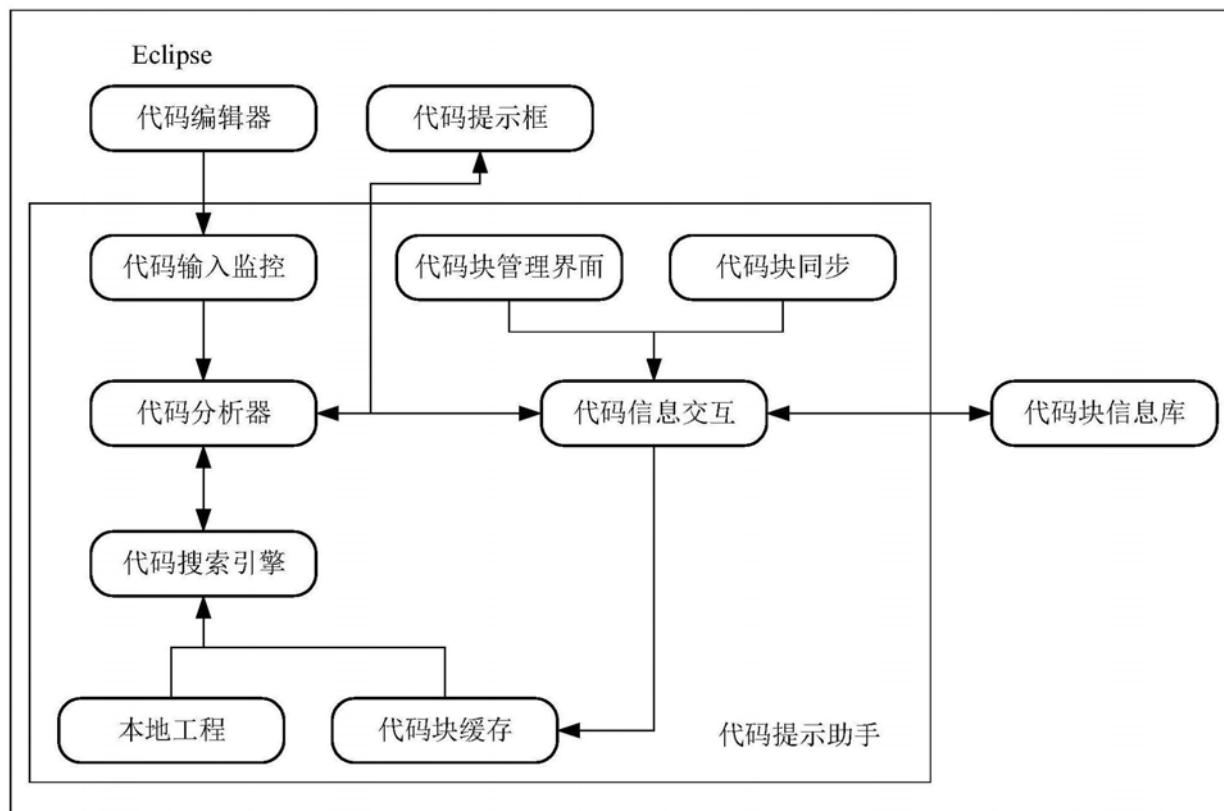


图6