



(12) 发明专利

(10) 授权公告号 CN 108427731 B

(45) 授权公告日 2021.08.13

(21) 申请号 201810161015.4

G06F 8/36 (2018.01)

(22) 申请日 2018.02.26

审查员 徐捷

(65) 同一申请的已公布的文献号

申请公布号 CN 108427731 A

(43) 申请公布日 2018.08.21

(73) 专利权人 平安普惠企业管理有限公司

地址 518000 广东省深圳市前海深港合作区前湾一路1号A栋201室(入驻深圳市前海商务秘书有限公司)

(72) 发明人 张彩凤

(74) 专利代理机构 深圳中一专利商标事务所

44237

代理人 官建红

(51) Int. Cl.

G06F 16/957 (2019.01)

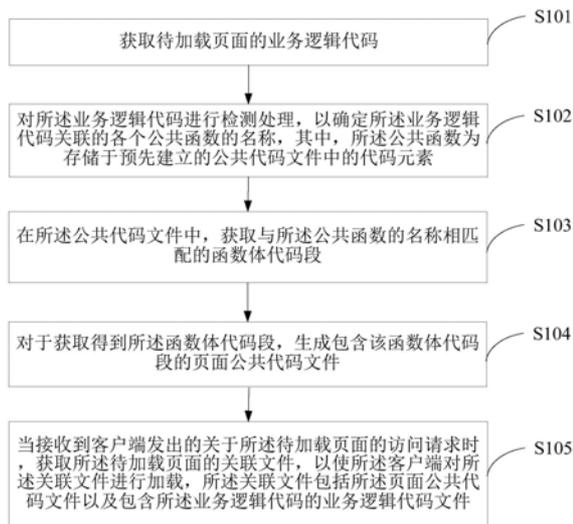
权利要求书2页 说明书9页 附图5页

(54) 发明名称

页面代码的处理方法、装置、终端设备及介质

(57) 摘要

本发明适用于互联网技术领域,提供了一种页面代码的处理方法、装置、终端设备及介质,包括:获取待加载页面的业务逻辑代码;确定业务逻辑代码关联的各个公共函数的名称;在公共代码文件中,获取与公共函数的名称相匹配的函数体代码段;对于获取得到函数体代码段,生成包含该函数体代码段的页面公共代码文件;当接收到客户端发出的关于待加载页面的访问请求时,获取待加载页面的关联文件,以使客户端加载页面公共代码文件以及业务逻辑代码文件。本发明保证了用户在通过客户端或浏览器来访问某一网页页面时,能够仅加载该网页页面对应的页面公共代码文件,而无需再加载公共代码文件中所包含的所有公共代码,故降低了页面加载过程中的代码冗余程度。



1. 一种页面代码的处理方法,其特征在于,包括:

获取待加载页面的业务逻辑代码;

对所述业务逻辑代码进行检测处理,以确定所述业务逻辑代码关联的各个公共函数的名称,其中,所述公共函数为存储于预先建立的公共代码文件中的代码元素;所述公共代码文件包含多个公共函数;

在所述公共代码文件中,获取与所述公共函数的名称相匹配的函数体代码段;

对于获取得到所述函数体代码段,重新生成包含该函数体代码段的页面公共代码文件;

当接收到客户端发出的关于所述待加载页面的访问请求时,获取所述待加载页面的关联文件,以使所述客户端对所述关联文件进行加载,所述关联文件包括所述页面公共代码文件以及包含所述业务逻辑代码的业务逻辑代码文件。

2. 如权利要求1所述的页面代码的处理方法,其特征在于,所述对所述业务逻辑代码进行检测处理,以确定所述业务逻辑代码关联的各个公共函数的名称,包括:

获取预设的公共函数标识符,所述公共函数标识符用于标识所述公共函数的起始位置;

对所述业务逻辑代码进行检测处理,其中,若在所述业务逻辑代码中检测到所述公共函数标识符,则将后邻于所述公共函数标识符的代码元素名确定为所述业务逻辑代码关联的所述公共函数的名称。

3. 如权利要求1所述的页面代码的处理方法,其特征在于,还包括:

获取用户预先输入的代码段,将所述代码段存储于所述公共代码文件中;

创建与所述代码段对应的代码模块;

在开发平台界面中,展示每一所述代码模块对应的控件,所述控件的标识符为所述用户输入的代码标识;

当检测到基于所述控件的选取指令时,将所述代码段插入至所述开发平台界面的代码编辑区域。

4. 如权利要求3所述的页面代码的处理方法,其特征在于,在所述创建与所述代码段对应的代码模块之前,还包括:

获取用户输入的所述代码标识;

判断所述代码标识与所述公共代码文件中任一所述公共函数的名称是否相同;

若所述代码标识与所述公共代码文件中任一所述公共函数的名称相同,则拒绝创建与所述代码段对应的代码模块。

5. 如权利要求3所述的页面代码的处理方法,其特征在于,在所述创建与所述代码段对应的代码模块之后,还包括:

展示所述代码模块对应的网络链接地址;

在所述代码编辑区域的地址链接标签中,若检测到所述用户键入所述网络链接地址,则将所述代码模块对应的所述代码段插入至所述代码编辑区域。

6. 一种终端设备,包括存储器以及处理器,所述存储器存储有可在所述处理器上运行的计算机程序,其特征在于,所述处理器执行所述计算机程序时实现如下步骤:

获取待加载页面的业务逻辑代码;

对所述业务逻辑代码进行检测处理,以确定所述业务逻辑代码关联的各个公共函数的名称,其中,所述公共函数为存储于预先建立的公共代码文件中的代码元素;所述公共代码文件包含多个公共函数;

在所述公共代码文件中,获取与所述公共函数的名称相匹配的函数体代码段;

对于获取得到所述函数体代码段,重新生成包含该函数体代码段的页面公共代码文件;

当接收到客户端发出的关于所述待加载页面的访问请求时,获取所述待加载页面的关联文件,以使所述客户端对所述关联文件进行加载,所述关联文件包括所述页面公共代码文件以及包含所述业务逻辑代码的业务逻辑代码文件。

7.如权利要求6所述的终端设备,其特征在于,所述对所述业务逻辑代码进行检测处理,以确定所述业务逻辑代码关联的各个公共函数的名称的步骤,具体包括:

获取预设的公共函数标识符,所述公共函数标识符用于标识所述公共函数的起始位置;

对所述业务逻辑代码进行检测处理,其中,若在所述业务逻辑代码中检测到所述公共函数标识符,则将后邻于所述公共函数标识符的代码元素名确定为所述业务逻辑代码关联的所述公共函数的名称。

8.如权利要求6所述的终端设备,其特征在于,所述处理器执行所述计算机程序时,还实现如下步骤:

获取用户预先输入的代码段,将所述代码段存储于所述公共代码文件中;

创建与所述代码段对应的代码模块;

在开发平台界面中,展示每一所述代码模块对应的控件,所述控件的标识符为所述用户输入的代码标识;

当检测到基于所述控件的选取指令时,将所述代码段插入至所述开发平台界面的代码编辑区域。

9.如权利要求8所述的终端设备,其特征在于,所述处理器执行所述计算机程序时,还实现如下步骤:

获取用户输入的所述代码标识;

判断所述代码标识与所述公共代码文件中任一所述公共函数的名称是否相同;

若所述代码标识与所述公共代码文件中任一所述公共函数的名称相同,则拒绝创建与所述代码段对应的代码模块。

10.一种计算机可读存储介质,所述计算机可读存储介质存储有计算机程序,其特征在于,所述计算机程序被处理器执行时实现如权利要求1至5任一项所述方法的步骤。

## 页面代码的处理方法、装置、终端设备及介质

### 技术领域

[0001] 本发明属于互联网技术领域,尤其涉及一种页面代码的处理方法、装置、终端设备及介质。

### 背景技术

[0002] 在软件开发项目中,开发人员经常会使用到一些相同的功能模块,因此,为了减少代码重复编写的情况出现,项目组通常都会预先建立一个公共代码文件,并将常用的函数、类代码存储于该公共代码文件中,由此使得开发人员所编写的各个网页页面可以直接从该公共代码文件中,快速调用所需的功能模块。

[0003] 对于某一网页页面来说,其通常包括业务逻辑代码以及公共代码两部分。网页页面所需调用的公共代码,往往只是公共代码文件中的一小部分代码,并且,不同网页页面实际需要从公共代码文件中加载的一部分公共代码也不相同。然而,在浏览器加载一网页页面时,除了加载该网页页面的业务逻辑代码之外,为了调用业务逻辑代码所关联的公共代码,浏览器还得同时加载后台所预先建立的整个公共代码文件。可见,浏览器实际加载的公共代码文件中包含了许多网页页面无须调用的公共代码,因此,现有技术中存在页面代码加载冗余的问题。

### 发明内容

[0004] 有鉴于此,本发明实施例提供了一种页面代码的处理方法、装置、终端设备及介质,以解决现有技术中所存在的页面代码加载冗余的问题。

[0005] 本发明实施例的第一方面提供了一种页面代码的处理方法,包括:

[0006] 获取待加载页面的业务逻辑代码;

[0007] 对所述业务逻辑代码进行检测处理,以确定所述业务逻辑代码关联的各个公共函数的名称,其中,所述公共函数为存储于预先建立的公共代码文件中的代码元素;

[0008] 在所述公共代码文件中,获取与所述公共函数的名称相匹配的函数体代码段;

[0009] 对于获取得到所述函数体代码段,生成包含该函数体代码段的页面公共代码文件;

[0010] 当接收到客户端发出的关于所述待加载页面的访问请求时,获取所述待加载页面的关联文件,以使所述客户端对所述关联文件进行加载,所述关联文件包括所述页面公共代码文件以及包含所述业务逻辑代码的业务逻辑代码文件。

[0011] 本发明实施例的第二方面提供了一种页面代码的处理装置,包括:

[0012] 第一获取单元,用于获取待加载页面的业务逻辑代码;

[0013] 检测单元,用于对所述业务逻辑代码进行检测处理,以确定所述业务逻辑代码关联的各个公共函数的名称,其中,所述公共函数为存储于预先建立的公共代码文件中的代码元素;

[0014] 第二获取单元,用于在所述公共代码文件中,获取与所述公共函数的名称相匹配

的函数体代码段；

[0015] 生成单元,用于对于获取得到所述函数体代码段,生成包含该函数体代码段的页面公共代码文件；

[0016] 加载单元,用于当接收到客户端发出的关于所述待加载页面的访问请求时,获取所述待加载页面的关联文件,以使所述客户端对所述关联文件进行加载,所述关联文件包括所述页面公共代码文件以及包含所述业务逻辑代码的业务逻辑代码文件。

[0017] 本发明实施例的第三方面提供了一种终端设备,包括存储器以及处理器,所述存储器存储有可在所述处理器上运行的计算机程序,所述处理器执行所述计算机程序时实现如第一方面所述的页面代码的处理方法的步骤。

[0018] 本发明实施例的第四方面提供了一种计算机可读存储介质,所述计算机可读存储介质存储有计算机程序,所述计算机程序被处理器执行时实现如第一方面所述的页面代码的处理方法的步骤。

[0019] 本发明实施例中,通过对页面的业务逻辑代码进行检测,以识别出其所包含的公共函数,并从公共代码文件中提取出该公共函数所匹配的函数体代码段,使其存储于重新生成的一个页面公共代码文件中,保证了用户在通过客户端或浏览器来访问某一网页页面时,能够仅加载该网页页面对应的页面公共代码文件,而无需再加载公共代码文件中所包含的所有公共代码,因此,降低了页面加载过程中的代码冗余程度,使得客户端所加载的公共代码就是当前页面所需调用的代码;并且,由于公共代码的加载数据量大大减小,因而也提高了网页的加载速度。

## 附图说明

[0020] 为了更清楚地说明本发明实施例中的技术方案,下面将对实施例或现有技术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发明的一些实施例,对于本领域普通技术人员来讲,在不付出创造性劳动性的前提下,还可以根据这些附图获得其他的附图。

[0021] 图1是本发明实施例提供的页面代码的处理方法的实现流程图；

[0022] 图2是本发明实施例提供的页面代码的处理方法S102的具体实现流程图；

[0023] 图3是本发明另一实施例提供的页面代码的处理方法的实现流程图；

[0024] 图4是本发明又一实施例提供的页面代码的处理方法的实现流程图；

[0025] 图5是本发明再一实施例提供的页面代码的处理方法的实现流程图；

[0026] 图6是本发明实施例提供的页面代码的处理装置的结构框图；

[0027] 图7是本发明实施例提供的终端设备的示意图。

## 具体实施方式

[0028] 以下描述中,为了说明而不是为了限定,提出了诸如特定系统结构、技术之类的具体细节,以便透彻理解本发明实施例。然而,本领域的技术人员应当清楚,在没有这些具体细节的其它实施例中也可以实现本发明。在其它情况中,省略对众所周知的系统、装置、电路以及方法的详细说明,以免不必要的细节妨碍本发明的描述。

[0029] 为了说明本发明所述的技术方案,下面通过具体实施例来进行说明。

[0030] 图1示出了本发明实施例提供的页面代码的处理方法的实现流程,该方法流程包括步骤S101至S105。各步骤的具体实现原理如下:

[0031] S101:获取待加载页面的业务逻辑代码。

[0032] 不同的项目需要开发不同的功能,不同的功能需要有不同的实现,实现这些核心功能的代码即为业务逻辑代码。待加载页面的业务逻辑代码由开发人员预先生成且存储于开发平台的信息库中。

[0033] 本发明实施例中,客户端可基于访问地址来访问页面,故将该访问地址所请求的页面称为上述待加载页面。其中,客户端包括但不限于终端设备所运行的浏览器以及安装于终端设备上的各类移动应用等。可见,待加载页面可以是浏览器所需展示的网页页面,也可以是移动应用中的应用界面。

[0034] 本发明实施例中,根据接收的访问地址,从信息库中读取待加载页面的业务逻辑代码。

[0035] S102:对所述业务逻辑代码进行检测处理,以确定所述业务逻辑代码关联的各个公共函数的名称,其中,所述公共函数为存储于预先建立的公共代码文件中的代码元素。

[0036] 为了减少代码重复编写的情况出现,项目组通常都会预先建立一个公共代码文件,并将常用的函数、类代码存储于该公共代码文件中,由此使得开发人员所编写的各个页面可以直接从该公共代码文件中,快速调用所需的功能模块。因此,除了在上述信息库中存储待加载页面的业务逻辑代码之外,信息库中还存储有该公共代码文件。每一公共代码文件与多个页面的业务逻辑相关。例如,页面A与页面B均为域名abc.com下的网页页面,但具有不同的存储路径,则为了方便管理,将页面A以及页面B所关联的公共函数存储至同一公共代码文件中。

[0037] 本发明实施例中,对上述S101所获取得到的待加载页面的业务逻辑代码进行检测处理,以从该业务逻辑代码中,识别出其所包含且存储于公共代码文件中的各个公共函数的名称。

[0038] 具体地,所述对所述业务逻辑代码进行检测处理,以确定所述业务逻辑代码关联的各个公共函数的名称,包括:

[0039] 获取公共代码文件中各个公共函数的名称;对于其中每一公共函数的名称,将其作为参数传入业务逻辑代码中所预设的function\_exists()函数;运行该业务逻辑代码,以获取function\_exists()函数的输出值;若输出值为真值,则确定业务逻辑代码中存在该公共函数的名称;若输出值为假值,则确定业务逻辑代码中不存在该公共函数的名称。

[0040] 作为本发明的一个实施例,图2示出了本发明实施例提供的页面代码的处理方法S102的另一具体实现流程,详述如下:

[0041] S1021:获取预设的公共函数标识符,所述公共函数标识符用于标识所述公共函数的起始位置。

[0042] S1022:对所述业务逻辑代码进行检测处理,其中,若在所述业务逻辑代码中检测到所述公共函数标识符,则将后邻于所述公共函数标识符的代码元素名确定为所述业务逻辑代码关联的所述公共函数的名称。

[0043] 在业务逻辑代码中,以预设的标识符来定义其所需引用的一个公共函数,将该标识符称为公共函数标识符。根据预先制订的业务逻辑代码的编写协议,获取与该编写协议

匹配的上述公共函数标识符。上述公共函数标识符例如可以是“\$”、“@”以及由一个或多个字符所组合得到的其他标识符等。

[0044] 依次遍历业务逻辑代码中的每一语句,以确定上述公共函数标识符是否存在于该语句中的任一位置。若检测到语句中的任一位置存在该公共函数标识符,则确定该位置所对应的语句为业务逻辑代码调用公共函数的语句。此时,对该语句进行分析处理,以获取该语句所包含的各个代码元素,上述代码元素包括但不限于函数名、变量名、类名、链接以及文件名等。例如,通过对语句进行分词处理,可将空格符所分隔得到的每一字符串输出为一个代码元素。

[0045] 在识别出语句所包含的各个代码元素后,获取后邻于公共函数标识符的代码元素,则将该代码元素确定为业务逻辑代码所需关联调用的一个公共函数的名称。

[0046] S103:在所述公共代码文件中,获取与所述公共函数的名称相匹配的函数体代码段。

[0047] 本发明实施例中,公共代码文件包含多个公共函数,每一公共函数包含公共函数的名称及其对应的函数体代码段。其中,函数体代码段为实现公共函数功能的所有代码所组成的整体。

[0048] 根据确定出的业务逻辑代码所关联的公共函数的名称,在公共代码文件中查找与该名称对应的字符串,并读取位于该字符串之后的一段代码。具体地,由于函数体代码段通常以预设的标识符来作为其起止标志,因此,将该字符串之后的第一个起止标志之间所包含的代码段确定为与该公共函数的名称所对应的函数体代码段。

[0049] 示例性地,上述起止标志例如可以是“{”以及“}”。

[0050] S104:对于获取得到所述函数体代码段,生成包含该函数体代码段的页面公共代码文件。

[0051] 本发明实施例中,创建与待加载页面对应的一个页面公共代码文件,将上述获取得到的函数体代码段及其对应的公共函数的名称写入该页面公共代码文件中。

[0052] 优选地,将待加载页面所关联的样式代码以及图片也写入该页面公共代码文件,以使客户端在加载该页面公共代码文件时,能够从中直接调取页面的显示样式以及页面所需展示的图片,因而提高了页面的加载效率以及渲染效率。

[0053] 本发明实施例中,页面公共代码文件为JavaScript文件。

[0054] S105:当接收到客户端发出的关于所述待加载页面的访问请求时,获取所述待加载页面的关联文件,以使所述客户端对所述关联文件进行加载,所述关联文件包括所述页面公共代码文件以及包含所述业务逻辑代码的业务逻辑代码文件。

[0055] 每一待加载页面所对应的页面公共代码文件可预先生成,也可实时生成。即,在接收到客户端发出的关于一待加载页面的访问请求时,执行上述S101至S104,或者,在接收到客户端发出的关于一待加载页面的访问请求之前,就先执行上述S101至S104。

[0056] 本发明实施例中,当接收到客户端发出的关于待加载页面的访问请求时,从信息库中读取该待加载页面的关联文件,并将关联文件输出为访问请求的响应信息,以将其返回至客户端。其中,待加载页面的关联文件包括该待加载页面的页面公共代码文件以及业务逻辑代码文件。

[0057] 客户端接收到待加载页面的关联文件后,通过相应的解析组件,如浏览器或者移

动应用,对该关联文件进行解析,使得客户端在运行业务逻辑代码的过程中,能够迅速调用页面公共代码文件中的函数体代码段,从而实现当前页面的加载。

[0058] 本发明实施例中,通过对页面的业务逻辑代码进行检测,以识别出其所包含的公共函数,并从公共代码文件中提取出该公共函数所匹配的函数体代码段,使其存储于重新生成的一个页面公共代码文件中,保证了用户在通过客户端或浏览器来访问某一网页页面时,能够仅加载该网页页面对应的页面公共代码文件,而无需再加载公共代码文件中所包含的所有公共代码,因此,降低了页面加载过程中的代码冗余程度,使得客户端所加载的公共代码就是当前页面所需调用的代码;并且,由于公共代码的加载数据量大大减小,因而也提高了网页的加载速度。

[0059] 作为本发明的另一个实施例,图3示出了本发明另一实施例提供的页面代码的处理方法的实现流程,详述如下:

[0060] S106:获取用户预先输入的代码段,将所述代码段存储于所述公共代码文件中。

[0061] 本发明实施例中,为了便于用户开发待加载页面的业务逻辑代码以及公共代码,可将用户所预先输入的代码段作为一公共函数来键入至公共代码文件。

[0062] 具体地,当接收到用户输入的公共函数创建指令时,在开发平台界面生成代码编辑区域,并请求用户在代码编辑区域输入自定义的代码语句。当检测到公共函数保存按键的触发事件时,将代码编辑区域中的各代码语句输出为一公共函数的函数体代码段,并将该函数体代码段添加至信息库所存储的公共代码文件的末端,或者,将该函数体代码段添加至实时生成的一个新的公共代码文件中。

[0063] S107:创建与所述代码段对应的代码模块。

[0064] S108:在开发平台界面中,展示每一所述代码模块对应的控件,所述控件的标识符为所述用户输入的代码标识。

[0065] 本发明实施例中,为了便于对各函数体代码段进行管理,将代码编辑区域中所接收到的代码段创建为一代码模块,并生成代码模块对应的一个控件。将该控件展示于开发平台界面,并将控件以及代码模块的对应关系存储于后台数据表中。并且,开发平台界面所展示的控件的标识符为用户输入的代码标识。

[0066] 其中,上述获取用户输入的代码标识,其可通过以下两种方式来实现:方式一,在开发平台界面弹出提示窗口,以提示用户在该窗口中输入当前所创建的代码模块所对应的代码标识,包括但不限于由一类或多类字符所组成的字符串;方式二,基于上述预设的公共函数标识符,对代码编辑区域的代码段进行检测,以获取检测得到的后邻于公共函数标识符的代码元素名,并将该代码元素名确定为该用户所输入的代码标识。

[0067] 示例性地,与代码模块对应的上述控件为按键,即以按键的形式,在开发平台界面的选项栏中展示一个新的按键,并将该按键的名称设置为获取到的代码标识。

[0068] S109:当检测到基于所述控件的选取指令时,将所述代码段插入至所述开发平台界面的代码编辑区域。

[0069] 开发平台界面的各个控件能够基于用户的点击操作而被选取。当检测到基于某一控件的选取指令时,根据预存储的控件以及代码模块的对应关系,获取与该控件对应的代码模块。在读取该代码模块所表征一代码段后,将该代码段显示至开发平台界面的代码编辑区域。

[0070] 优选地,在将该代码段显示至开发平台界面的代码编辑区域时,具体是将该代码段插入至光标所指向的空白区域,或者,将代码段及其对应的公共函数的名称共同插入至代码编辑区域的顶端。

[0071] 优选地,在检测到用户登录开发平台时,根据用户的登录账号来确定出该用户的代码模块选取权限。若检测到用户所选取的控件不属于其代码模块选取权限的范围之内,则停止将该控件对应的代码段插入至代码编辑区域,由此避免了用户自定义创建的代码模块被其他开发项目组所非法调用,因而提高了页面代码的管理安全。

[0072] 本发明实施例中,通过在开发平台界面展示每一代码模块所对应的控件,并在检测到基于控件的选取指令时,将相应的代码段插入至代码编辑区域,使得用户能够在编辑页面中的业务逻辑代码时,快速地引入所需调用的公共函数的函数体代码段,因而提高了开发效率;由于用户可以自定义代码模块所对应的代码段,并将代码模块存储展示为一个控件,故扩展了开发平台所存储的公共函数的数量,提高了开发操作的灵活性。

[0073] 作为本发明的又一个实施例,图4示出了本发明又一实施例提供的页面代码的处理方法的实现流程。如图4所示,在上述S107之前,还包括:

[0074] S110:获取用户输入的所述代码标识。

[0075] S111:判断所述代码标识与所述公共代码文件中任一所述公共函数的名称是否相同。

[0076] S112:拒绝创建与所述代码段对应的代码模块。

[0077] 本发明实施例中,将用户输入的代码标识识别为用户所需创建的一个公共函数的名称,为了避免公共函数的名称与公共代码文件中已存在的公共函数的名称发生冲突,在将代码段存储至公共代码文件以及创建代码段所对应的代码模块之前,判断该代码标识是否与公共代码文件中的任一公共函数的名称相同。若否,则执行步骤S107;若是,则执行步骤S112,拒绝创建与所述代码段对应的代码模块,并提示用户该代码标识已存在,需要重新输入代码段所对应的代码标识,此后,返回执行上述步骤S110;直至代码标识与公共代码文件中每一公共函数的名称均不相同,才创建与代码段对应的代码模块。

[0078] 本发明实施例中未提及的步骤实现原理与上述实施例中步骤实现原理相同,因此不再一一赘述。

[0079] 本发明实施例中,通过在接收到用户输入的代码段以及代码标识后,预先判断代码标识是否与公共代码文件中任一公共函数的名称相同,在判断结果为是时,拒绝创建与代码段对应的代码模块,避免了开发平台界面会出现相同代码标识的控件,保证了代码模块以及公共函数名称的唯一性,从而提高了页面代码开发的准确率,避免了后续基于相同代码标识的控件来读取并插入代码段时出现代码段导入错误的情况发生。

[0080] 作为本发明的再一实施例,图5示出了本发明再一实施例提供的页面代码的处理方法的实现流程,如图5所示,在上述S107之后,还包括:

[0081] S113:展示所述代码模块对应的网络链接地址。

[0082] 本发明实施例中,将接收到的每一代码段存储至新建的一个公共代码文件中,并将公共代码文件保存至信息库的不同存储路径之下。信息库的各个存储路径能够对外提供访问,因此,基于代码段的存储路径,生成该代码段对应的网络链接地址,并将该网络链接地址发布于开发平台界面中。

[0083] 优选地,将网络链接地址与代码段所对应的控件绑定,以在检测到控件选取指令时,将该控件所对应的网络链接地址输入剪贴板中,以使用户在代码编辑区域中执行粘贴操作时,能够将剪贴板中的网络链接地址插入至代码编辑区域。

[0084] S114:在所述代码编辑区域的地址链接标签中,若检测到所述用户键入所述网络链接地址,则将所述代码模块对应的所述代码段插入至所述代码编辑区域。

[0085] 本发明实施例中,代码编辑区域还展示有预设的多个功能菜单标签,例如,地址链接标签、保存标签以及权限设置标签等。若在地址链接标签中检测到用户输入了任一网络链接地址,则对该网络链接地址进行解析处理,以在信息库中判断该网络链接地址是否存在对应的代码模块。

[0086] 同理,若在代码编辑区域中检测到用户基于剪贴板所复制的网络链接地址来执行粘贴操作时,在信息库中判断该网络链接地址是否存在对应的代码模块。

[0087] 若信息库中存在该网络链接地址对应的代码模块,则读取该代码模块对应的代码段,并将该代码段插入至当前的代码编辑区域;若信息库中不存在该网络链接地址对应的代码模块,则将该网络链接地址作为实时生成的一个地址标签中的属性值后,插入至代码编辑区域。

[0088] 本发明实施例中,通过展示代码模块所对应的网络链接地址,使得用户可以灵活地在不同平台的代码编辑区域直接键入所需插入的代码模块的网络链接地址,使得最终在加载编写完成的待加载页面时,能够直接调用与该网络链接地址匹配的代码段,因而也提高了页面代码的开发效率,避免了开发人员需要重复输入相同的代码段的问题出现。由于不同的待加载页面可以共同调用相同网络链接地址的代码段,故也降低了所需加载的公共代码文件的冗余程度。

[0089] 应理解,上述实施例中各步骤的序号的大小并不意味着执行顺序的先后,各过程的执行顺序应以其功能和内在逻辑确定,而不对本发明实施例的实施过程构成任何限定。

[0090] 对应于本发明实施例所提供的页面代码的处理方法,图6示出了本发明实施例提供的页面代码的处理装置的结构框图。为了便于说明,仅示出了与本实施例相关的部分。

[0091] 参照图6,该装置包括:

[0092] 第一获取单元61,用于获取待加载页面的业务逻辑代码。

[0093] 检测单元62,用于对所述业务逻辑代码进行检测处理,以确定所述业务逻辑代码关联的各个公共函数的名称,其中,所述公共函数为存储于预先建立的公共代码文件中的代码元素。

[0094] 第二获取单元63,用于在所述公共代码文件中,获取与所述公共函数的名称相匹配的函数体代码段。

[0095] 生成单元64,用于对于获取得到所述函数体代码段,生成包含该函数体代码段的页面公共代码文件。

[0096] 加载单元65,用于当接收到客户端发出的关于所述待加载页面的访问请求时,获取所述待加载页面的关联文件,以使所述客户端对所述关联文件进行加载,所述关联文件包括所述页面公共代码文件以及包含所述业务逻辑代码的业务逻辑代码文件。

[0097] 可选地,所述检测单元62包括:

[0098] 获取子单元,用于获取预设的公共函数标识符,所述公共函数标识符用于标识所述公共函数的起始位置。

[0099] 检测子单元,用于对所述业务逻辑代码进行检测处理,其中,若在所述业务逻辑代码中检测到所述公共函数标识符,则将后邻于所述公共函数标识符的代码元素名确定为所述业务逻辑代码关联的所述公共函数的名称。

[0100] 可选地,所述页面代码的处理装置还包括:

[0101] 第三获取单元,用于获取用户预先输入的代码段,将所述代码段存储于所述公共代码文件中。

[0102] 创建单元,用于创建与所述代码段对应的代码模块。

[0103] 展示单元,用于在开发平台界面中,展示每一所述代码模块对应的控件,所述控件的标识符为所述用户输入的代码标识。

[0104] 插入单元,用于当检测到基于所述控件的选取指令时,将所述代码段插入至所述开发平台界面的代码编辑区域。

[0105] 可选地,所述页面代码的处理装置还包括:

[0106] 第四获取单元,用于获取用户输入的所述代码标识。

[0107] 判断单元,用于判断所述代码标识与所述公共代码文件中任一所述公共函数的名称是否相同。

[0108] 拒绝单元,用于若所述代码标识与所述公共代码文件中任一所述公共函数的名称相同,则拒绝创建与所述代码段对应的代码模块。

[0109] 可选地,所述页面代码的处理装置还包括:

[0110] 地址展示单元,用于展示所述代码模块对应的网络链接地址。

[0111] 第二插入单元,用于在所述代码编辑区域的地址链接标签中,若检测到所述用户键入所述网络链接地址,则将所述代码模块对应的所述代码段插入至所述代码编辑区域。

[0112] 本发明实施例中,通过对页面的业务逻辑代码进行检测,以识别出其所包含的公共函数,并从公共代码文件中提取出该公共函数所匹配的函数体代码段,使其存储于重新生成的一个页面公共代码文件中,保证了用户在通过客户端或浏览器来访问某一网页页面时,能够仅加载该网页页面对应的页面公共代码文件,而无需再加载公共代码文件中所包含的所有公共代码,因此,降低了页面加载过程中的代码冗余程度,使得客户端所加载的公共代码就是当前页面所需调用的代码;并且,由于公共代码的加载数据量大大减小,因而也提高了网页的加载速度。

[0113] 图7是本发明一实施例提供的终端设备的示意图。如图7所示,该实施例的终端设备7包括:处理器70以及存储器71,所述存储器71中存储有可在所述处理器70上运行的计算机程序72,例如页面代码的处理程序。所述处理器70执行所述计算机程序72时实现上述各个页面代码的处理方法实施例中的步骤,例如图1所示的步骤101至105。或者,所述处理器70执行所述计算机程序72时实现上述各装置实施例中各模块/单元的功能,例如图6所示单元61至65的功能。

[0114] 示例性的,所述计算机程序72可以被分割成一个或多个模块/单元,所述一个或者多个模块/单元被存储在所述存储器71中,并由所述处理器70执行,以完成本发明。所述一个或多个模块/单元可以是能够完成特定功能的一系列计算机程序指令段,该指令段用于

描述所述计算机程序72在所述终端设备7中的执行过程。

[0115] 所述终端设备7可以是桌上型计算机、笔记本、掌上电脑及云端服务器等计算设备。所述终端设备可包括,但不限于,处理器70、存储器71。本领域技术人员可以理解,图7仅仅是终端设备7的示例,并不构成对终端设备7的限定,可以包括比图示更多或更少的部件,或者组合某些部件,或者不同的部件,例如所述终端设备还可以包括输入输出设备、网络接入设备、总线等。

[0116] 所称处理器70可以是中央处理单元(Central Processing Unit,CPU),还可以是其他通用处理器、数字信号处理器(Digital Signal Processor,DSP)、专用集成电路(Application Specific Integrated Circuit,ASIC)、现成可编程门阵列(Field-Programmable Gate Array,FPGA)或者其他可编程逻辑器件、分立门或者晶体管逻辑器件、分立硬件组件等。通用处理器可以是微处理器或者该处理器也可以是任何常规的处理器等。

[0117] 所述存储器71可以是所述终端设备7的内部存储单元,例如终端设备7的硬盘或内存。所述存储器71也可以是所述终端设备7的外部存储设备,例如所述终端设备7上配备的插接式硬盘,智能存储卡(Smart Media Card,SMC),安全数字(Secure Digital,SD)卡,闪存卡(Flash Card)等。进一步地,所述存储器71还可以既包括所述终端设备7的内部存储单元也包括外部存储设备。所述存储器71用于存储所述计算机程序以及所述终端设备所需的其他程序和数据。所述存储器71还可以用于暂时地存储已经输出或者将要输出的数据。

[0118] 另外,在本申请各个实施例中的各功能单元可以集成在一个处理单元中,也可以是各个单元单独物理存在,也可以两个或两个以上单元集成在一个单元中。上述集成的单元既可以采用硬件的形式实现,也可以采用软件功能单元的形式实现。

[0119] 所述集成的单元如果以软件功能单元的形式实现并作为独立的产品销售或使用,可以存储在一个计算机可读取存储介质中。基于这样的理解,本申请的技术方案本质上或者说对现有技术做出贡献的部分或者该技术方案的全部或部分可以以软件产品的形式体现出来,该计算机软件产品存储在一个存储介质中,包括若干指令用以使得一台计算机设备(可以是个人计算机,服务器,或者网络设备)执行本申请各个实施例所述方法的全部或部分步骤。而前述的存储介质包括:U盘、移动硬盘、只读存储器(Read-Only Memory,ROM)、随机存取存储器(Random Access Memory,RAM)、磁碟或者光盘等各种可以存储程序代码的介质。

[0120] 以上所述,以上实施例仅用以说明本申请的技术方案,而非对其限制;尽管参照前述实施例对本申请进行了详细的说明,本领域的普通技术人员应当理解:其依然可以对前述各实施例所记载的技术方案进行修改,或者对其中部分技术特征进行等同替换;而这些修改或者替换,并不使相应技术方案的本质脱离本申请各实施例技术方案的精神和范围。

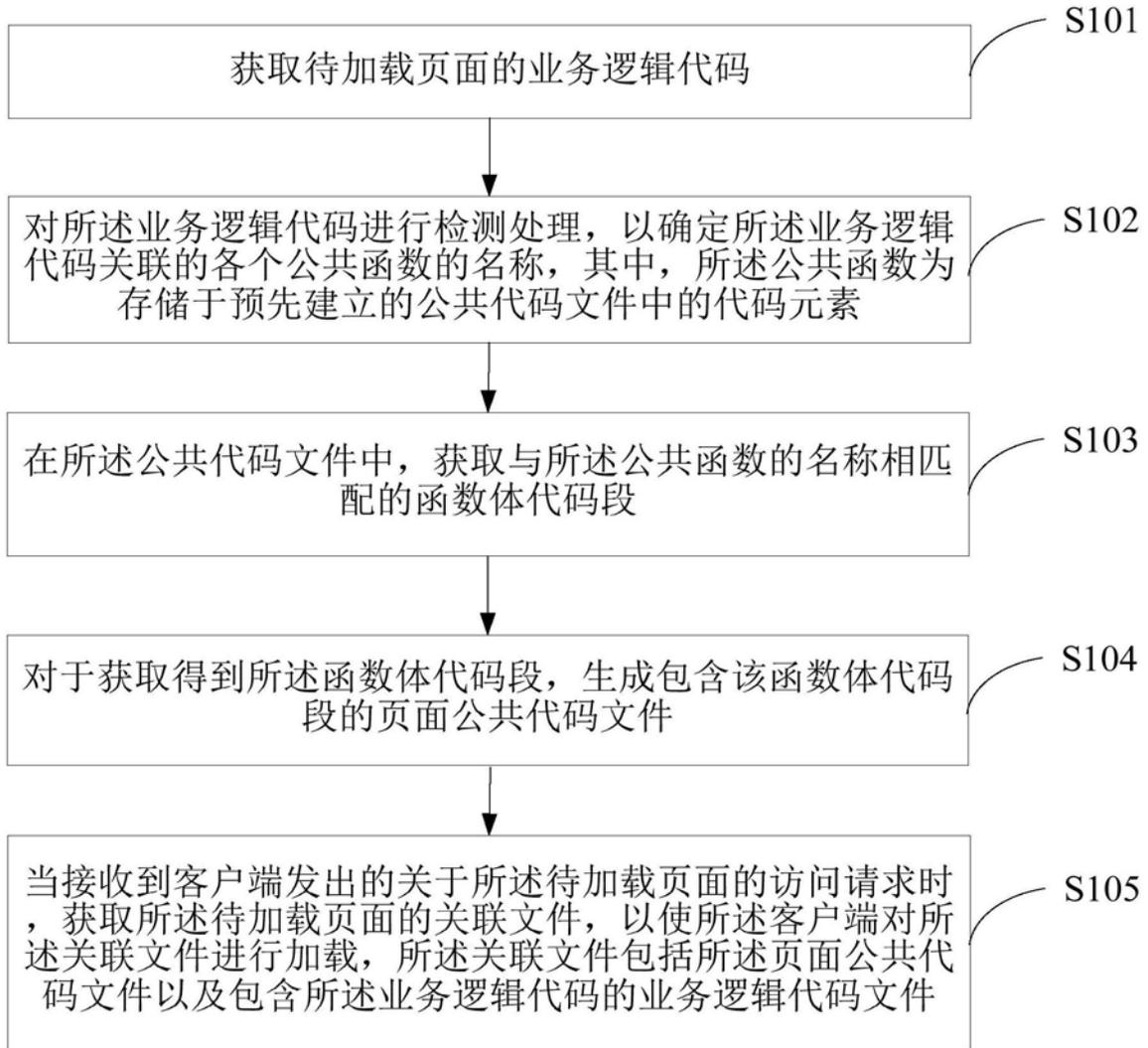


图1

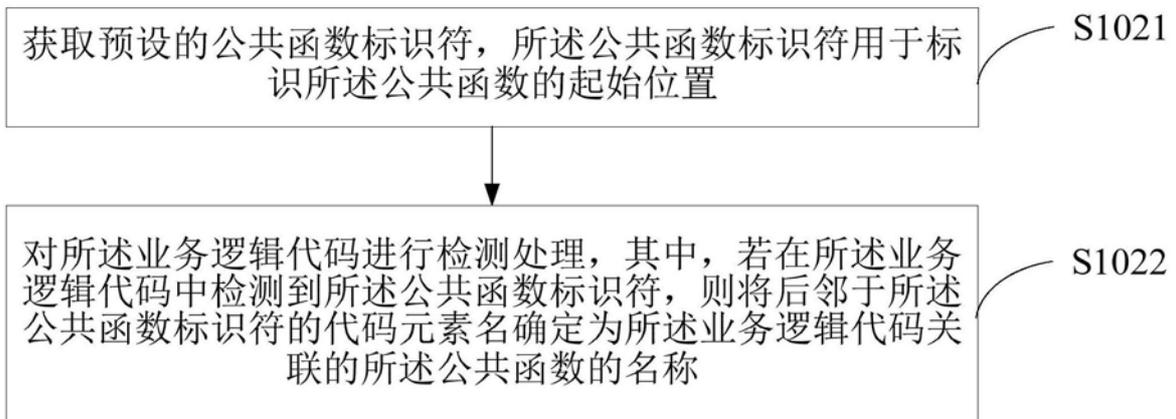


图2

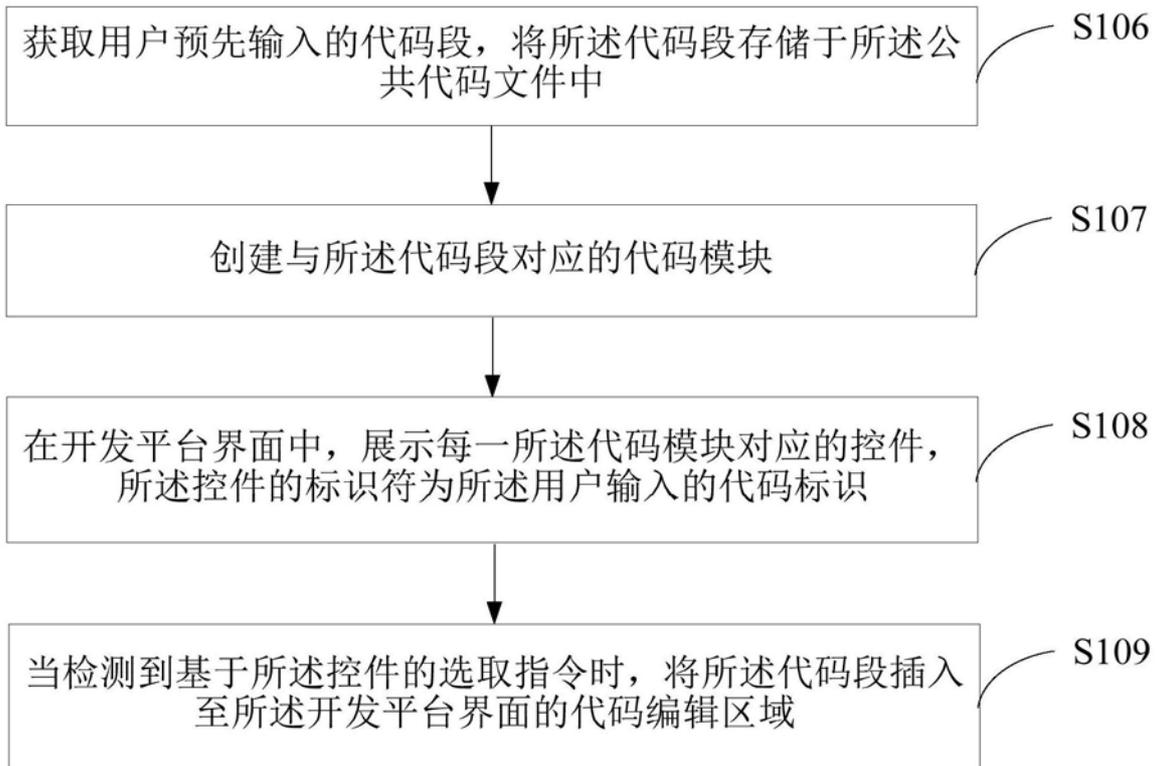


图3

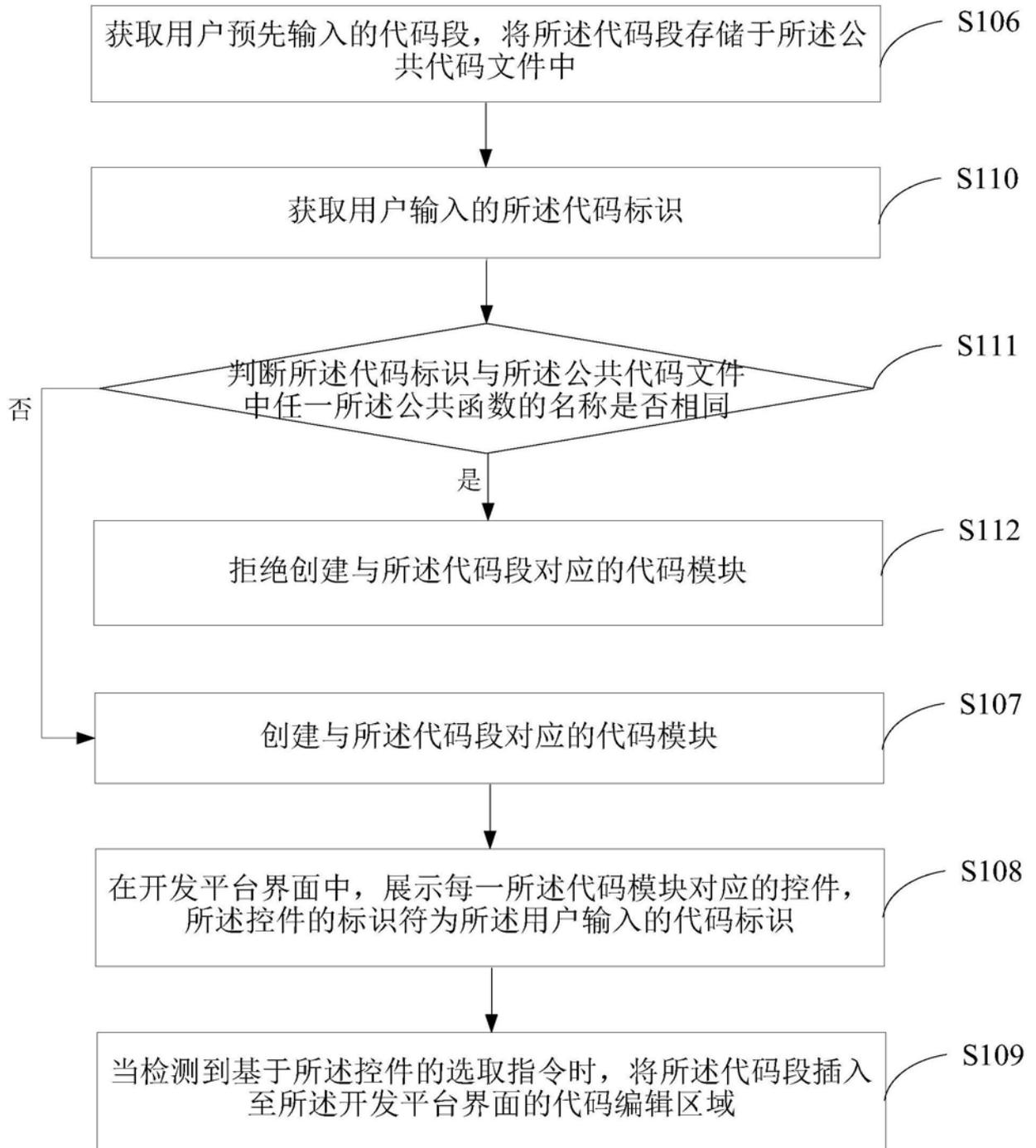


图4

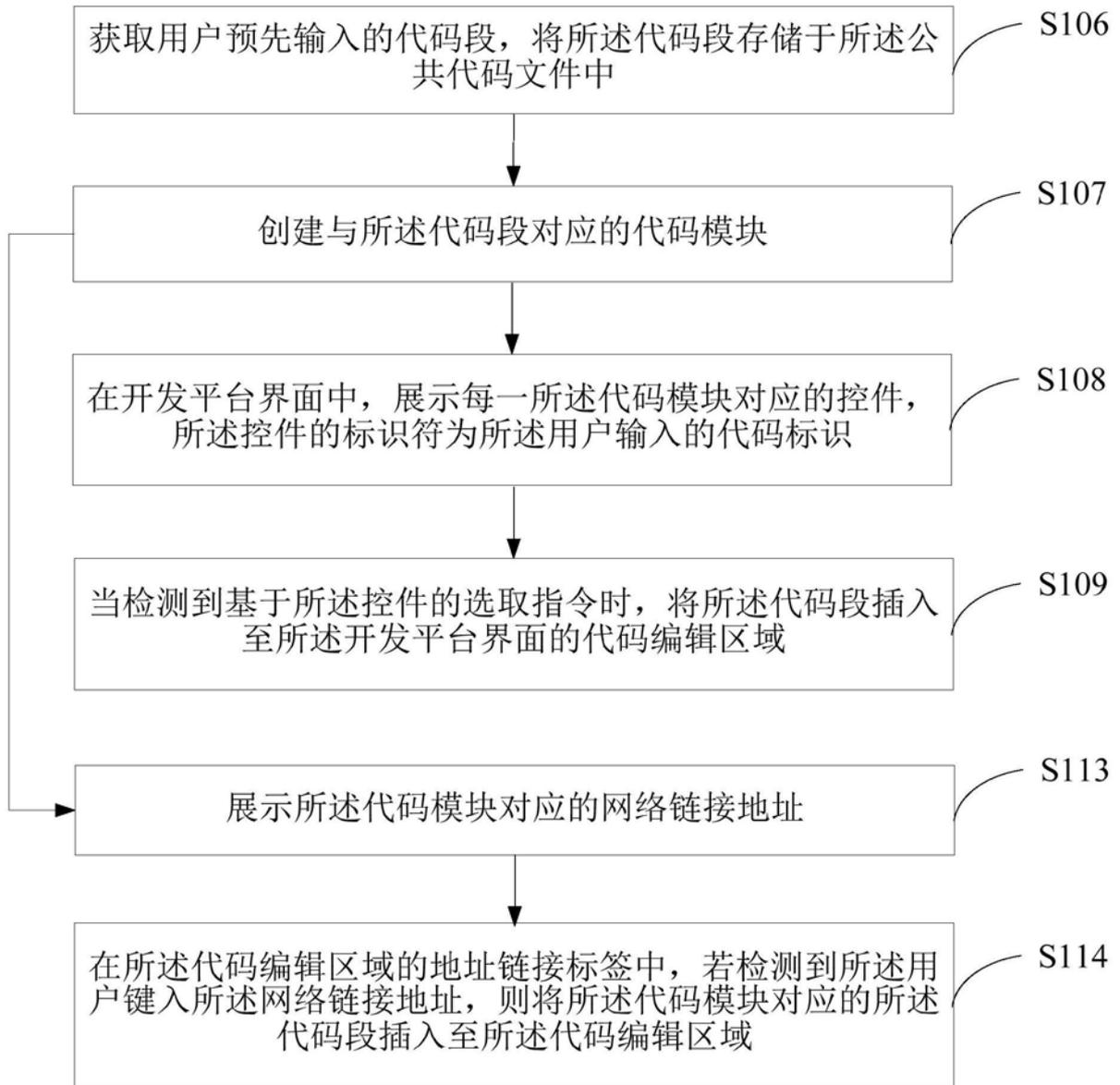


图5

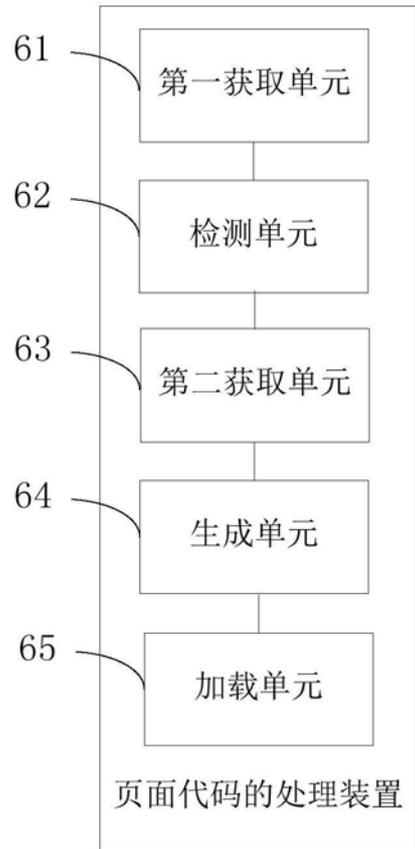


图6

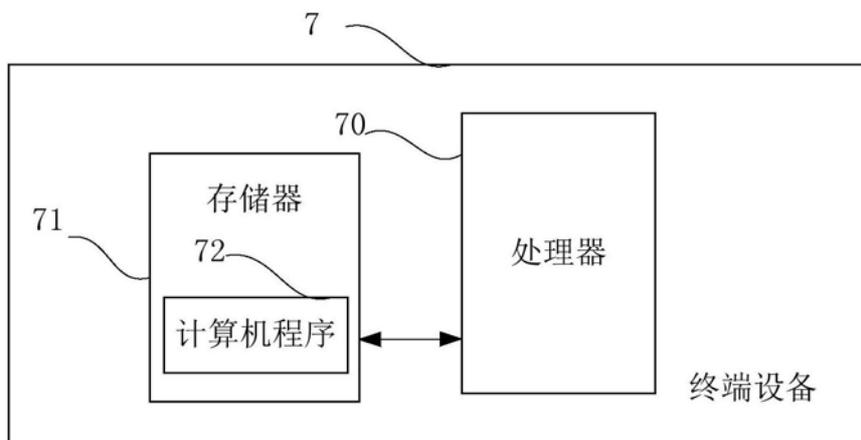


图7