



(12) 发明专利

(10) 授权公告号 CN 114168227 B

(45) 授权公告日 2024. 08. 06

(21) 申请号 202111508681.9

(22) 申请日 2021.12.10

(65) 同一申请的已公布的文献号

申请公布号 CN 114168227 A

(43) 申请公布日 2022.03.11

(73) 专利权人 掌阅科技股份有限公司

地址 100124 北京市朝阳区四惠大厦2029E

(72) 发明人 谢德良

(74) 专利代理机构 北京市浩天知识产权代理事

务所(普通合伙) 11276

专利代理师 张颖瑛

(51) Int. Cl.

G06F 9/445 (2018.01)

G06F 8/71 (2018.01)

(56) 对比文件

CN 107577483 A, 2018.01.12

CN 110244990 A, 2019.09.17

审查员 张永辉

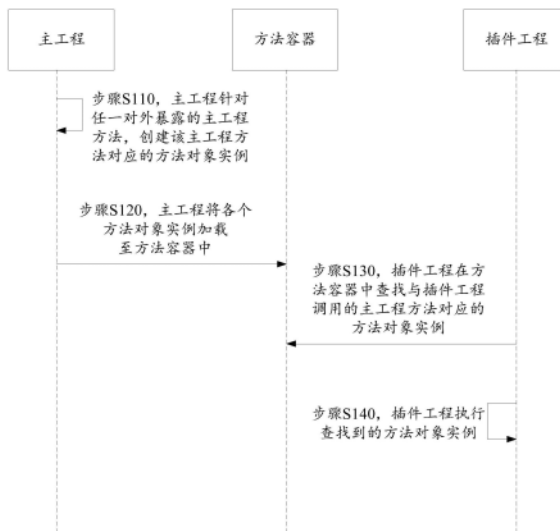
权利要求书3页 说明书10页 附图3页

(54) 发明名称

程序调用的处理方法、电子设备及计算机存储介质

(57) 摘要

本发明公开了一种程序调用的处理方法、电子设备及计算机存储介质。方法包括：主工程针对任一对外暴露的主工程方法，创建该主工程方法对应的方法对象实例；主工程将各个方法对象实例加载至方法容器中；插件工程在方法容器中查找与插件工程调用的主工程方法对应的方法对象实例，并执行查找到的方法对象实例。本方案中，插件工程在调用主工程方法时，无需识别主工程方法对应的主工程版本，即可实现主工程方法的调用。从而一方面提升了插件工程调用主工程方法的调用效率，另一方面避免了现有技术中出现的因主工程版本混乱而引起的调用错误的技术弊端。



1. 一种程序调用的处理方法,包括:

主工程针对任一对外暴露的主工程方法,创建该主工程方法对应的方法对象实例;其中,针对同时存在于多个主工程版本中对外暴露的主工程方法,仅创建最新的主工程版本中该主工程方法对应的方法对象实例;

主工程将各个方法对象实例加载至方法容器中;该方法容器为开辟的内存空间;

插件工程在所述方法容器中查找与所述插件工程调用的主工程方法对应的方法对象实例,并执行查找到的方法对象实例。

2. 根据权利要求1所述的方法,其中,所述主工程针对任一对外暴露的主工程方法,创建该主工程方法对应的方法对象实例进一步包括:

主工程针对任一对外暴露的主工程方法,确定该主工程方法隶属的方法类型;

根据该主工程方法隶属的方法类型,创建该主工程方法对应的方法对象实例。

3. 根据权利要求2所述的方法,其中,所述方法类型包括以下类型中的一种:

无参数无返回值类型、无参数有返回值类型、有参数无返回值类型以及有参数有返回值类型。

4. 根据权利要求1-3中任一项所述的方法,其中,所述主工程针对任一对外暴露的主工程方法,创建该主工程方法对应的方法对象实例进一步包括:

主工程针对任一对外暴露的主工程方法,获取该主工程方法的方法名;

创建该主工程方法对应的方法对象实例,以使所述方法对象实例的属性中包含该主工程方法的方法名。

5. 根据权利要求4所述的方法,其中,所述插件工程在所述方法容器中查找与所述插件工程调用的主工程方法对应的方法对象实例进一步包括:

插件工程确定调用的主工程方法的方法名;

插件工程依据调用的主工程方法的方法名,在所述方法容器中查找与所述插件工程调用的主工程方法对应的方法对象实例。

6. 根据权利要求1-3中任一项所述的方法,其中,所述主工程针对任一对外暴露的主工程方法,创建该主工程方法对应的方法对象实例进一步包括:

主工程初始化时,主工程针对于支持的至少一个主工程版本中的任一对外暴露的主工程方法,创建该主工程方法对应的方法对象实例。

7. 根据权利要求1-3中任一项所述的方法,其中,在所述主工程将各个方法对象实例加载至方法容器中之前,所述方法还包括:

主工程初始化时,加载所述主工程对应的方法容器。

8. 一种程序调用的处理方法,所述方法包括:

针对任一对外暴露的主工程方法,创建该主工程方法对应的方法对象实例;其中,针对同时存在于多个主工程版本中对外暴露的主工程方法,仅创建最新的主工程版本中该主工程方法对应的方法对象实例;

将各个方法对象实例加载至方法容器中,以供插件工程在所述方法容器中查找与所述插件工程调用的主工程方法对应的方法对象实例,并执行查找到的方法对象实例;该方法容器为开辟的内存空间。

9. 根据权利要求8所述的方法,其中,所述针对任一对外暴露的主工程方法,创建该主

工程方法对应的方法对象实例进一步包括：

针对任一对外暴露的主工程方法，确定该主工程方法隶属的方法类型；
根据该主工程方法隶属的方法类型，创建该主工程方法对应的方法对象实例。

10. 根据权利要求9所述的方法，其中，所述方法类型包括以下类型中的一种：

无参数无返回值类型、无参数有返回值类型、有参数无返回值类型以及有参数有返回值类型。

11. 根据权利要求8-10中任一项所述的方法，其中，所述针对任一对外暴露的主工程方法，创建该主工程方法对应的方法对象实例进一步包括：

针对任一对外暴露的主工程方法，获取该主工程方法的方法名；

创建该主工程方法对应的方法对象实例，以使所述方法对象实例的属性中包含该主工程方法的方法名。

12. 根据权利要求8-10中任一项所述的方法，其中，所述针对任一对外暴露的主工程方法，创建该主工程方法对应的方法对象实例进一步包括：

初始化时，针对于支持的至少一个主工程版本中的任一对外暴露的主工程方法，创建该主工程方法对应的方法对象实例。

13. 根据权利要求8-10中任一项所述的方法，其中，在所述将各个方法对象实例加载至方法容器中之前，所述方法还包括：

初始化时，加载所述主工程对应的方法容器。

14. 一种程序调用的处理方法，所述方法包括：

在方法容器中查找与调用的主工程方法对应的方法对象实例；其中，针对同时存在于多个主工程版本中对外暴露的主工程方法，仅创建最新的主工程版本中该主工程方法对应的方法对象实例；该方法容器为开辟的内存空间；

执行查找到的方法对象实例。

15. 根据权利要求14所述的方法，其中，所述在方法容器中查找与调用的主工程方法对应的方法对象实例进一步包括：

确定调用的主工程方法的方法名；

依据调用的主工程方法的方法名，在方法容器中查找与调用的主工程方法对应的方法对象实例。

16. 一种主工程客户端，所述主工程客户端包含至少一个可执行指令，所述可执行指令使处理器执行如下操作：

针对任一对外暴露的主工程方法，创建该主工程方法对应的方法对象实例；其中，针对同时存在于多个主工程版本中对外暴露的主工程方法，仅创建最新的主工程版本中该主工程方法对应的方法对象实例；

将各个方法对象实例加载至方法容器中，以供插件工程在所述方法容器中查找与所述插件工程调用的主工程方法对应的方法对象实例，并执行查找到的方法对象实例；该方法容器为开辟的内存空间。

17. 根据权利要求16所述的主工程客户端，其中，所述可执行指令使处理器执行如下操作：

针对任一对外暴露的主工程方法，确定该主工程方法隶属的方法类型；

根据该主工程方法隶属的方法类型,创建该主工程方法对应的方法对象实例。

18. 根据权利要求17所述的主工程客户端,其中,所述方法类型包括以下类型中的一种:

无参数无返回值类型、无参数有返回值类型、有参数无返回值类型以及有参数有返回值类型。

19. 根据权利要求16-18中任一项所述的主工程客户端,其中,所述可执行指令使处理器执行如下操作:

针对任一对外暴露的主工程方法,获取该主工程方法的方法名;

创建该主工程方法对应的方法对象实例,以使所述方法对象实例的属性中包含该主工程方法的方法名。

20. 根据权利要求16-18中任一项所述的主工程客户端,其中,所述可执行指令使处理器执行如下操作:

初始化时,针对于支持的至少一个主工程版本中的任一对外暴露的主工程方法,创建该主工程方法对应的方法对象实例。

21. 根据权利要求16-18中任一项所述的主工程客户端,其中,所述可执行指令使处理器执行如下操作:

初始化时,加载所述主工程对应的方法容器。

22. 一种插件工程客户端,所述插件工程客户端包含至少一个可执行指令,所述可执行指令使处理器执行如下操作:

在方法容器中查找与调用的主工程方法对应的方法对象实例;其中,针对同时存在于多个主工程版本中对外暴露的主工程方法,仅创建最新的主工程版本中该主工程方法对应的方法对象实例;该方法容器为开辟的内存空间;

执行查找到的方法对象实例。

23. 根据权利要求22所述的插件工程客户端,其中,所述可执行指令使处理器执行如下操作:

确定调用的主工程方法的方法名;

依据调用的主工程方法的方法名,在方法容器中查找与调用的主工程方法对应的方法对象实例。

24. 一种应用程序客户端,包括如权利要求16-21任一项所述的主工程客户端以及如权利要求22或23所述的插件工程客户端。

25. 一种电子设备,包括:处理器、存储器、通信接口和通信总线,所述处理器、所述存储器和所述通信接口通过所述通信总线完成相互间的通信;

所述存储器用于存放至少一可执行指令,所述可执行指令使所述处理器执行如权利要求8-15任一项所述的程序调用的处理方法对应的操作。

26. 一种计算机存储介质,所述存储介质中存储有至少一可执行指令,所述可执行指令使处理器执行如权利要求8-15任一项所述的程序调用的处理方法对应的操作。

程序调用的处理方法、电子设备及计算机存储介质

技术领域

[0001] 本发明涉及计算机技术领域,具体涉及一种程序调用的处理方法、电子设备及计算机存储介质。

背景技术

[0002] 随着科技及社会的不断发展,应用程序所能提供的功能越来越多。为了提高应用程序的开发及执行效率,许多应用程序采用了多工程模式。在多工程模式中,应用程序包含主工程及插件工程。其中,主工程用于提供应用程序的基础功能,插件工程用于提供某项或某几项附加功能。在许多应用程序中,插件工程通常需调用主工程中的主工程方法来实现插件功能。

[0003] 然而,发明人在实施过程中发现,现有技术中存在如下缺陷:由于不同版本的主工程中包含的主工程方法会存在差异,例如,版本1的主工程中包含有可供插件工程调用的主工程方法A,版本2的主工程中包含有可供插件工程调用的主工程方法B。现有技术中的插件工程在调用主工程方法时,需要预先确定出调用的主工程方法对应的主工程版本。然而,在主工程并行开发等情况下,极易出现主工程版本混乱的情况,从而导致确定出的调用主工程方法对应的主工程版本错误的情况,继而导致插件工程执行错误。

发明内容

[0004] 鉴于上述问题,提出了本发明以便提供一种克服上述问题或者至少部分地解决上述问题的程序调用的处理方法、电子设备及计算机存储介质。

[0005] 根据本发明的第一方面,提供了一种程序调用的处理方法,包括:

[0006] 主工程针对任一对外暴露的主工程方法,创建该主工程方法对应的方法对象实例;

[0007] 主工程将各个方法对象实例加载至方法容器中;

[0008] 插件工程在所述方法容器中查找与所述插件工程调用的主工程方法对应的方法对象实例,并执行查找到的方法对象实例。

[0009] 根据本发明的第二方面,提供了一种程序调用的处理方法,所述方法包括:

[0010] 针对任一对外暴露的主工程方法,创建该主工程方法对应的方法对象实例;

[0011] 将各个方法对象实例加载至方法容器中,以供插件工程在所述方法容器中查找与所述插件工程调用的主工程方法对应的方法对象实例,并执行查找到的方法对象实例。

[0012] 根据本发明的第三方面,提供了一种程序调用的处理方法,所述方法包括:

[0013] 在方法容器中查找与调用的主工程方法对应的方法对象实例;

[0014] 执行查找到的方法对象实例。

[0015] 根据本发明的第四方面,提供了一种主工程客户端,所述主工程客户端包含至少一个可执行指令,所述可执行指令使处理器执行如下操作:

[0016] 针对任一对外暴露的主工程方法,创建该主工程方法对应的方法对象实例;

[0017] 将各个方法对象实例加载至方法容器中,以供插件工程在所述方法容器中查找与
所述插件工程调用的主工程方法对应的方法对象实例,并执行查找到的方法对象实例。

[0018] 根据本发明的第五方面,提供了一种插件工程客户端,所述插件工程客户端包含
至少一个可执行指令,所述可执行指令使处理器执行如下操作:

[0019] 在方法容器中查找与调用的主工程方法对应的方法对象实例;

[0020] 执行查找到的方法对象实例。

[0021] 根据本发明的第六方面,提供了一种应用程序客户端,包括上述主工程客户端以
及上述插件工程客户端。

[0022] 根据本发明的第七方面,提供了一种电子设备,包括:处理器、存储器、通信接口和
通信总线,所述处理器、所述存储器和所述通信接口通过所述通信总线完成相互间的通信;

[0023] 所述存储器用于存放至少一可执行指令,所述可执行指令使所述处理器执行上述
程序调用的处理方法对应的操作。

[0024] 根据本发明的第八方面,提供了一种计算机存储介质,所述存储介质中存储有至
少一可执行指令,所述可执行指令使处理器执行如上述程序调用的处理方法对应的操作。

[0025] 本发明公开的程序调用的处理方法、电子设备及计算机存储介质中:主工程针对
任一对外暴露的主工程方法,创建该主工程方法对应的方法对象实例;主工程将各个方法
对象实例加载至方法容器中;插件工程在方法容器中查找与插件工程调用的主工程方法对
应的方法对象实例,并执行查找到的方法对象实例。本方案中,插件工程在调用主工程方法
时,无需识别主工程方法对应的主工程版本,即可实现主工程方法的调用。从而一方面提升
了插件工程调用主工程方法的调用效率,另一方面避免了现有技术中出现的因主工程版本
混乱而引起的调用错误的技术弊端。

[0026] 上述说明仅是本发明技术方案的概述,为了能够更清楚了解本发明的技术手段,
而可依照说明书的内容予以实施,并且为了让本发明的上述和其它目的、特征和优点能够
更明显易懂,以下特举本发明的具体实施方式。

附图说明

[0027] 通过阅读下文优选实施方式的详细描述,各种其他的优点和益处对于本领域普通
技术人员将变得清楚明了。附图仅用于示出优选实施方式的目的,而并不认为是对本发明
的限制。而且在整个附图中,用相同的参考符号表示相同的部件。在附图中:

[0028] 图1示出了本发明实施例一提供的一种程序调用的处理方法的流程示意图;

[0029] 图2示出了本发明实施例二提供的一种程序调用的处理方法的流程示意图;

[0030] 图3示出了本发明实施例三提供的一种程序调用的处理方法的流程示意图;

[0031] 图4示出了本发明实施例六提供的一种应用程序客户端的示意图;

[0032] 图5示出了本发明实施例七提供的一种电子设备的结构示意图。

具体实施方式

[0033] 下面将参照附图更详细地描述本发明的示例性实施例。虽然附图中显示了本发明
的示例性实施例,然而应当理解,可以以各种形式实现本发明而不应被这里阐述的实施例
所限制。相反,提供这些实施例是为了能够更透彻地理解本发明,并且能够将本发明的范围

完整的传达给本领域的技术人员。

[0034] 实施例一

[0035] 图1示出了本发明实施例一提供的一种程序调用的处理方法的流程示意图。本发明实施例所提供的程序调用的处理方法可以应用于任何多工程模式的应用程序中,本发明实施例提供的方法可以由应用程序所处的终端设备执行。

[0036] 如图1所示,该方法包括如下步骤:

[0037] 步骤S110,主工程针对任一对外暴露的主工程方法,创建该主工程方法对应的方法对象实例。

[0038] 主工程中包含对内的主工程方法以及对外暴露的主工程方法。对内的主工程方法无法供插件工程调用,而对外暴露的主工程方法可以供插件工程调用。

[0039] 本发明实施例针对于任一对外暴露的主工程方法,将该主工程方法抽象为方法对象,并通过方法对象的实例化生成该主工程方法对应的方法对象实例。其中,本发明实施例对具体实例化的方式不作限定。例如,可以使用new操作符、调用Class对象的newInstance()方法、调用clone()方法,通过ObjectInputStream的readObject()方法反序列化类等方式创建方法对象实例。

[0040] 在一种可选的实施方式中,为了提高插件工程调用主工程方法的执行结果的准确度,本发明实施例将对外暴露的主工程方法进行类别划分。其中,对外暴露的主工程方法的方法类型包括以下类型中的一种:无参数无返回值类型、无参数有返回值类型、有参数无返回值类型以及有参数有返回值类型。其中,插件工程不向无参数无返回值类型的主工程方法提供参数,并且无参数无返回值类型的主工程方法也不会向插件工程返回返回值,例如该无参数无返回值类型的主工程方法可以为书籍计数方法,通过该主工程方法执行书籍计数加1,并不返回计数结果;插件工程不向无参数有返回值类型的主工程方法提供参数,但无参数有返回值类型的主工程方法会向插件工程返回相应的返回值,例如该无参数有返回值类型的主工程方法可以为用户数统计方法,即通过该主工程方法执行用户计数加1,并返回计数结果;插件工程向有参数无返回值类型的主工程方法提供参数,但有参数无返回值类型的主工程方法不会向插件工程返回相应的返回值,例如该有参数无返回值类型的主工程方法可以为费用统计方法,即插件工程传递费用参数,该主工程方法通过传递的费用参数计算总费用,但不向插件工程返回总费用结果;插件工程向有参数有返回值类型的主工程方法提供参数,并且有参数有返回值类型的主工程方法会向插件工程返回相应的返回值,例如该有参数有返回值类型的主工程方法可以为书籍添加书架方法,即插件工程传递书籍参数,该主工程方法将传递的书籍参数对应的书籍添加至书架,并向插件工程返回添加成功的返回值。其中,本发明实施例所提供的方法类型并不仅限于上述四种,例如,还可以根据参数是否为多个和/或返回值是否为多个进行划分,继而得到的方法类型包括以下类型中的至少一种:无参数无返回值类型、无参数单返回值类型、无参数多返回值类型、单参数无返回值类型、多参数无返回值类型、单参数单返回值类型、单参数多返回值类型、多参数单返回值类型以及多参数单返回值类型等等。

[0041] 进一步地,主工程针对任一对外暴露的主工程方法,确定该主工程方法隶属的方法类型;继而根据该主工程方法隶属的方法类型,创建该主工程方法对应的方法对象实例,从而使该方法对象实例中携带该主工程方法的方法类型信息。

[0042] 在又一种可选的实施方式中,为了便于提升后续插件工程查找方法对象实例的查找效率及查找精度,主工程针对任一对外暴露的主工程方法,获取该主工程方法的方法名;并创建该主工程方法对应的方法对象实例,以使方法对象实例的属性中包含该主工程方法的方法名。

[0043] 在再一种可选的实施方式中,主工程在初始化时,主工程针对于支持的至少一个主工程版本中的任一对外暴露的主工程方法,创建该主工程方法对应的方法对象实例。从中可以看出,创建的方法对象实例对应的主工程方法可以对应于同一个主工程版本,也可以对应于不同的主工程版本等。并且,创建的方法对象实例对应的主工程方法均是当前主工程支持的版本中的方法,从而创建的方法对象实例能够正常执行。

[0044] 进一步可选的,同一对外暴露的主工程方法可以同时存在于多个主工程版本中,为节约资源以及进一步简化插件工程的方法调用过程,本发明实施例针对于同时存在于多个主工程版本中对外暴露的主工程方法,仅创建最新的主工程版本中该主工程方法对应的方法对象实例。

[0045] 步骤S120,主工程将各个方法对象实例加载至方法容器中。

[0046] 主工程将步骤S110创建的各个方法对象实例加载至方法容器中。该方法容器可以是开辟的内存空间等。在此应当理解的是,若直接将对外暴露的主工程方法加载至方法容器中,会导致因存储的主工程方法为字符串而引起的主工程方法无法执行的弊端。由此本发明实施例通过创建主工程方法对应的方法对象实例能够为主工程方法分配内存,从而后续能够使方法对象实例顺利执行。

[0047] 在一种可选的实施方式中,主工程初始化时,加载主工程对应的方法容器,从而再将各个方法对象实例加载至方法容器中。在主工程退出时释放该方法容器,从而实现方法容器的动态创建。采用该种方式,一方面满足插件工程调用主工程方法的需求,另一方面能够节约系统资源。

[0048] 在又一种可选的实施方式中,为避免插件工程调用错误,当有多个应用程序的主工程均处于运行状态时,分别加载各个主工程对应的方法容器,并为每个方法容器分配主工程标识。从而后续插件工程依据主工程标识确定出对应的方法容器,继而再在方法容器中查找对应的方法对象实例。

[0049] 步骤S130,插件工程在方法容器中查找与插件工程调用的主工程方法对应的方法对象实例。

[0050] 与现有技术不同的是,本发明实施例插件工程在调用主工程中的主工程方法时,无需查找调用的主工程方法对应的主工程版本,而是在主工程对应的方法容器中查找与插件工程调用的主工程方法对应的方法对象实例。

[0051] 在一种可选的实施方式中,插件工程具体是确定调用的主工程方法的方法名,继而插件工程依据调用的主工程方法的方法名,在方法容器中查找与插件工程调用的主工程方法对应的方法对象实例。其中,与插件工程调用的主工程方法对应的方法对象实例的方法名属性与调用的主工程方法的方法名一致。

[0052] 步骤S140,插件工程执行查找到的方法对象实例。

[0053] 若能够在方法容器中查找与插件工程调用的主工程方法对应的方法对象实例,则执行查找到的方法对象实例,从而实现主工程方法的调用。

[0054] 在一种可选的实施方式中,若无法在方法容器中查找到与插件工程调用的主工程方法对应的方法对象实例,则向插件工程反馈相应的调用失败的提示信息,以提示插件工程当前调用存在错误。进一步可选的,为提升故障解决效率,可以生成对应的故障工单,该工作工单中记录有主工程标识、插件工程标识、调用错误的主工程方法名等等。继而可以通过邮件、短信、系统内信息等方式将故障工单派发至插件工程或主工程对应的开发端或维护端,以供插件工程或主工程的开发者或维护者对故障进行及时修复。由于在方法容器中未查找与插件工程调用的主工程方法对应的方法对象实例时,并不会进一步执行该主工程方法的调用,从而避免了获得错误的调用结果;而且还可以避免现有技术中因执行了错误版本中主工程方法而导致的程序崩溃的弊端。

[0055] 由此可见,本发明实施例中,主工程创建对外暴露的主工程方法对应的方法对象实例,并在方法容器中加载创建的各个方法对象实例。插件工程在调用主工程方法时,无需识别主工程方法对应的主工程版本,仅需在方法容器中查找与插件工程调用的主工程方法对应的方法对象实例,并执行查找到的方法对象实例,即可实现主工程方法的调用。从而一方面提升了插件工程调用主工程方法的调用效率,另一方面避免了现有技术中出现的因主工程版本混乱而引起的调用错误的技术弊端。

[0056] 实施例二

[0057] 图2示出了本发明实施例二提供的一种程序调用的处理方法的流程示意图。本发明实施例所提供的程序调用的处理方法可以应用于任何多工程模式的应用程序中,本发明实施例提供的方法可以由主工程客户端执行。

[0058] 如图2所示,该方法包括如下步骤:

[0059] 步骤S210,针对任一对外暴露的主工程方法,创建该主工程方法对应的方法对象实例。

[0060] 步骤S220,将各个方法对象实例加载至方法容器中,以供插件工程在该方法容器中查找与插件工程调用的主工程方法对应的方法对象实例,并执行查找到的方法对象实例。

[0061] 在一种可选的实施方式中,所述针对任一对外暴露的主工程方法,创建该主工程方法对应的方法对象实例进一步包括:针对任一对外暴露的主工程方法,确定该主工程方法隶属的方法类型;根据该主工程方法隶属的方法类型,创建该主工程方法对应的方法对象实例。

[0062] 在一种可选的实施方式中,所述方法类型包括以下类型中的一种:

[0063] 无参数无返回值类型、无参数有返回值类型、有参数无返回值类型以及有参数有返回值类型。

[0064] 在一种可选的实施方式中,所述针对任一对外暴露的主工程方法,创建该主工程方法对应的方法对象实例进一步包括:

[0065] 针对任一对外暴露的主工程方法,获取该主工程方法的方法名;

[0066] 创建该主工程方法对应的方法对象实例,以使所述方法对象实例的属性中包含该主工程方法的方法名。

[0067] 在一种可选的实施方式中,所述针对任一对外暴露的主工程方法,创建该主工程方法对应的方法对象实例进一步包括:初始化时,针对于支持的至少一个主工程版本中的

任一对外暴露的主工程方法,创建该主工程方法对应的方法对象实例。

[0068] 在一种可选的实施方式中,所述将各个方法对象实例加载至方法容器中之前,所述方法还包括:初始化时,加载所述主工程对应的方法容器。

[0069] 本发明实施例的具体执行过程可参照实施例一中相应部分的描述,在此不作赘述。

[0070] 由此可见,本发明实施例创建对外暴露的主工程方法对应的方法对象实例,并在方法容器中加载创建的各个方法对象实例,从而能够使插件工程在调用主工程方法时,无需识别主工程方法对应的主工程版本便可以实现主工程方法的调用。从而一方面提升了插件工程调用主工程方法的调用效率,另一方面避免了现有技术中出现的因主工程版本混乱而引起的调用错误的技术弊端。

[0071] 实施例三

[0072] 图3示出了本发明实施例三提供的一种程序调用的处理方法的流程示意图。本发明实施例所提供的程序调用的处理方法可以应用于任何多工程模式的应用程序中,本发明实施例提供的方法可以由插件工程客户端执行。

[0073] 如图3所示,该方法包括如下步骤:

[0074] 步骤S310,在方法容器中查找与调用的主工程方法对应的方法对象实例。

[0075] 步骤S320,执行查找到的方法对象实例。

[0076] 在一种可选的实施方式中,所述在方法容器中查找与调用的主工程方法对应的方法对象实例进一步包括:确定调用的主工程方法的方法名;依据调用的主工程方法的方法名,在方法容器中查找与调用的主工程方法对应的方法对象实例。

[0077] 本发明实施例的具体执行过程可参照实施例一中相应部分的描述,在此不作赘述。

[0078] 由此可见,本发明实施例插件工程在调用主工程方法时,无需识别主工程方法对应的主工程版本便可以实现主工程方法的调用。从而一方面提升了插件工程调用主工程方法的调用效率,另一方面避免了现有技术中出现的因主工程版本混乱而引起的调用错误的技术弊端。

[0079] 实施例四

[0080] 本发明实施例四提供了一种主工程客户端,所述主工程客户端包含至少一个可执行指令,所述可执行指令使处理器执行如下操作:

[0081] 针对任一对外暴露的主工程方法,创建该主工程方法对应的方法对象实例;

[0082] 将各个方法对象实例加载至方法容器中,以供插件工程在所述方法容器中查找与所述插件工程调用的主工程方法对应的方法对象实例,并执行查找到的方法对象实例。

[0083] 在一种可选的实施方式中,所述可执行指令使处理器执行如下操作:

[0084] 针对任一对外暴露的主工程方法,确定该主工程方法隶属的方法类型;

[0085] 根据该主工程方法隶属的方法类型,创建该主工程方法对应的方法对象实例。

[0086] 在一种可选的实施方式中,所述方法类型包括以下类型中的一种:

[0087] 无参数无返回值类型、无参数有返回值类型、有参数无返回值类型以及有参数有返回值类型。

[0088] 在一种可选的实施方式中,所述可执行指令使处理器执行如下操作:

[0089] 针对任一对外暴露的主工程方法,获取该主工程方法的方法名;

[0090] 创建该主工程方法对应的方法对象实例,以使所述方法对象实例的属性中包含该主工程方法的方法名。

[0091] 在一种可选的实施方式中,所述可执行指令使处理器执行如下操作:

[0092] 初始化时,针对于支持的至少一个主工程版本中的任一对外暴露的主工程方法,创建该主工程方法对应的方法对象实例。

[0093] 在一种可选的实施方式中,所述可执行指令使处理器执行如下操作:

[0094] 所述将各个方法对象实例加载至方法容器中之前,初始化时,加载所述主工程对应的方法容器。

[0095] 由此可见,本发明实施例能够使插件工程在调用主工程方法时,无需识别主工程方法对应的主工程版本便可以实现主工程方法的调用。从而一方面提升了插件工程调用主工程方法的调用效率,另一方面避免了现有技术中出现的因主工程版本混乱而引起的调用错误的技术弊端。

[0096] 实施例五

[0097] 本发明实施例五提供了一种插件工程客户端,所述插件工程客户端包含至少一个可执行指令,所述可执行指令使处理器执行如下操作:

[0098] 在方法容器中查找与调用的主工程方法对应的方法对象实例;

[0099] 执行查找到的方法对象实例。

[0100] 在一种可选的实施方式中,所述可执行指令使处理器执行如下操作:

[0101] 确定调用的主工程方法的方法名;

[0102] 依据调用的主工程方法的方法名,在方法容器中查找与调用的主工程方法对应的方法对象实例。

[0103] 由此可见,本发明实施例在调用主工程方法时,无需识别主工程方法对应的主工程版本便可以实现主工程方法的调用。从而一方面提升了插件工程调用主工程方法的调用效率,另一方面避免了现有技术中出现的因主工程版本混乱而引起的调用错误的技术弊端。

[0104] 实施例六

[0105] 图4示出了本发明实施例六提供的一种应用程序客户端的示意图。如图4所示,应用程序客户端400包括上述主工程客户端410以及上述插件工程客户端420。

[0106] 实施例七

[0107] 图5示出了本发明实施例七提供的一种电子设备的结构示意图。本发明具体实施例并不对电子设备的具体实现做限定。

[0108] 如图5所示,该电子设备可以包括:处理器(processor)502、通信接口(Communications Interface)504、存储器(memory)506、以及通信总线508。

[0109] 其中:处理器502、通信接口504、以及存储器506通过通信总线508完成相互间的通信。通信接口504,用于与其它设备比如客户端或其它服务器等的网元通信。处理器502,用于执行程序510,具体可以执行上述用于程序调用的处理方法实施例中的相关步骤。

[0110] 具体地,程序510可以包括程序代码,该程序代码包括计算机操作指令。

[0111] 处理器502可能是中央处理器CPU,或者是特定集成电路ASIC(Application

Specific Integrated Circuit), 或者是被配置成实施本发明实施例的一个或多个集成电路。电子设备包括的一个或多个处理器, 可以是同一类型的处理器, 如一个或多个CPU; 也可以是不同类型的处理器, 如一个或多个CPU以及一个或多个ASIC。

[0112] 存储器506, 用于存放程序510。存储器506可能包含高速RAM存储器, 也可能还包括非易失性存储器(non-volatile memory), 例如至少一个磁盘存储器。

[0113] 程序510具体可以用于使得处理器502执行以下操作:

[0114] 针对任一对外暴露的主工程方法, 创建该主工程方法对应的方法对象实例;

[0115] 将各个方法对象实例加载至方法容器中, 以供插件工程在所述方法容器中查找与所述插件工程调用的主工程方法对应的方法对象实例, 并执行查找到的方法对象实例。

[0116] 在一种可选的实施方式中, 程序510具体可以用于使得处理器502执行以下操作:

[0117] 针对任一对外暴露的主工程方法, 确定该主工程方法隶属的方法类型;

[0118] 根据该主工程方法隶属的方法类型, 创建该主工程方法对应的方法对象实例。

[0119] 在一种可选的实施方式中, 所述方法类型包括以下类型中的一种:

[0120] 无参数无返回值类型、无参数有返回值类型、有参数无返回值类型以及有参数有返回值类型。

[0121] 在一种可选的实施方式中, 程序510具体可以用于使得处理器502执行以下操作:

[0122] 针对任一对外暴露的主工程方法, 获取该主工程方法的方法名;

[0123] 创建该主工程方法对应的方法对象实例, 以使所述方法对象实例的属性中包含该主工程方法的方法名。

[0124] 在一种可选的实施方式中, 程序510具体可以用于使得处理器502执行以下操作:

[0125] 初始化时, 针对于支持的至少一个主工程版本中的任一对外暴露的主工程方法, 创建该主工程方法对应的方法对象实例。

[0126] 在一种可选的实施方式中, 程序510具体可以用于使得处理器502执行以下操作:

[0127] 所述将各个方法对象实例加载至方法容器中之前, 初始化时, 加载所述主工程对应的方法容器。

[0128] 在一种可选的实施方式中, 程序510具体可以用于使得处理器502执行以下操作:

[0129] 在方法容器中查找与调用的主工程方法对应的方法对象实例; 执行查找到的方法对象实例。

[0130] 在一种可选的实施方式中, 程序510具体可以用于使得处理器502执行以下操作:

[0131] 确定调用的主工程方法的方法名;

[0132] 依据调用的主工程方法的方法名, 在方法容器中查找与调用的主工程方法对应的方法对象实例。

[0133] 由此可见, 本发明实施例中, 主工程创建对外暴露的主工程方法对应的方法对象实例, 并在方法容器中加载创建的各个方法对象实例。插件工程在调用主工程方法时, 无需识别主工程方法对应的主工程版本, 仅需在方法容器中查找与插件工程调用的主工程方法对应的方法对象实例, 并执行查找到的方法对象实例, 即可实现主工程方法的调用。从而一方面提升了插件工程调用主工程方法的调用效率, 另一方面避免了现有技术中出现的因主工程版本混乱而引起的调用错误的技术弊端。

[0134] 实施例八

[0135] 本发明实施例八提供了一种非易失性计算机存储介质,所述计算机存储介质存储有至少一可执行指令,该计算机可执行指令可执行上述任意方法实施例中的方法。

[0136] 可执行指令具体可以用于使得处理器执行以下操作:

[0137] 针对任一对外暴露的主工程方法,创建该主工程方法对应的方法对象实例;

[0138] 将各个方法对象实例加载至方法容器中,以供插件工程在所述方法容器中查找与所述插件工程调用的主工程方法对应的方法对象实例,并执行查找到的方法对象实例。

[0139] 在一种可选的实施方式中,可执行指令具体可以用于使得处理器执行以下操作:

[0140] 针对任一对外暴露的主工程方法,确定该主工程方法隶属的方法类型;

[0141] 根据该主工程方法隶属的方法类型,创建该主工程方法对应的方法对象实例。

[0142] 在一种可选的实施方式中,所述方法类型包括以下类型中的一种:

[0143] 无参数无返回值类型、无参数有返回值类型、有参数无返回值类型以及有参数有返回值类型。

[0144] 在一种可选的实施方式中,可执行指令具体可以用于使得处理器执行以下操作:

[0145] 针对任一对外暴露的主工程方法,获取该主工程方法的方法名;

[0146] 创建该主工程方法对应的方法对象实例,以使所述方法对象实例的属性中包含该主工程方法的方法名。

[0147] 在一种可选的实施方式中,可执行指令具体可以用于使得处理器执行以下操作:

[0148] 初始化时,针对于支持的至少一个主工程版本中的任一对外暴露的主工程方法,创建该主工程方法对应的方法对象实例。

[0149] 在一种可选的实施方式中,可执行指令具体可以用于使得处理器执行以下操作:

[0150] 所述将各个方法对象实例加载至方法容器中之前,初始化时,加载所述主工程对应的方法容器。

[0151] 在一种可选的实施方式中,可执行指令具体可以用于使得处理器执行以下操作:

[0152] 在方法容器中查找与调用的主工程方法对应的方法对象实例;执行查找到的方法对象实例。

[0153] 在一种可选的实施方式中,可执行指令具体可以用于使得处理器执行以下操作:

[0154] 确定调用的主工程方法的方法名;

[0155] 依据调用的主工程方法的方法名,在方法容器中查找与调用的主工程方法对应的方法对象实例。

[0156] 由此可见,本发明实施例中,主工程创建对外暴露的主工程方法对应的方法对象实例,并在方法容器中加载创建的各个方法对象实例。插件工程在调用主工程方法时,无需识别主工程方法对应的主工程版本,仅需在方法容器中查找与插件工程调用的主工程方法对应的方法对象实例,并执行查找到的方法对象实例,即可实现主工程方法的调用。从而一方面提升了插件工程调用主工程方法的调用效率,另一方面避免了现有技术中出现的因主工程版本混乱而引起的调用错误的技术弊端。

[0157] 在此提供的算法或显示不与任何特定计算机、虚拟系统或者其它设备固有相关。各种通用系统也可以与基于在此的示教一起使用。根据上面的描述,构造这类系统所要求的结构是显而易见的。此外,本发明实施例也不针对任何特定编程语言。应当明白,可以利用各种编程语言实现在此描述的本发明的内容,并且上面对特定语言所做的描述是为了披

露本发明的最佳实施方式。

[0158] 在此处所提供的说明书中,说明了大量具体细节。然而,能够理解,本发明的实施例可以在没有这些具体细节的情况下实践。在一些实例中,并未详细示出公知的方法、结构和技术,以便不模糊对本说明书的理解。

[0159] 类似地,应当理解,为了精简本发明并帮助理解各个发明方面中的一个或多个,在上面对本发明的示例性实施例的描述中,本发明实施例的各个特征有时被一起分组到单个实施例、图、或者对其的描述中。然而,并不应将该公开的方法解释成反映如下意图:即所要求保护的本发明要求比在每个权利要求中所明确记载的特征更多的特征。更确切地说,如下面的权利要求书所反映的那样,发明方面在于少于前面公开的单个实施例的所有特征。因此,遵循具体实施方式的权利要求书由此明确地并入该具体实施方式,其中每个权利要求本身都作为本发明的单独实施例。

[0160] 本领域那些技术人员可以理解,可以对实施例中的设备中的模块进行自适应性地改变并且把它们设置在与该实施例不同的一个或多个设备中。可以把实施例中的模块或单元或组件组合成一个模块或单元或组件,以及此外可以把它们分成多个子模块或子单元或子组件。除了这样的特征和/或过程或者单元中的至少一些是相互排斥之外,可以采用任何组合对本说明书(包括伴随的权利要求、摘要和附图)中公开的所有特征以及如此公开的任何方法或者设备的所有过程或单元进行组合。除非另外明确陈述,本说明书(包括伴随的权利要求、摘要和附图)中公开的每个特征可以由提供相同、等同或相似目的的替代特征来代替。

[0161] 此外,本领域的技术人员能够理解,尽管在此的一些实施例包括其它实施例中所包括的某些特征而不是其它特征,但是不同实施例的特征的组合意味着处于本发明的范围之内并且形成不同的实施例。例如,在下面的权利要求书中,所要求保护的实施例的任意之一都可以以任意的组合方式来使用。

[0162] 本发明的各个部件实施例可以以硬件实现,或者以在一个或者多个处理器上运行的软件模块实现,或者以它们的组合实现。本领域的技术人员应当理解,可以在实践中使用微处理器或者数字信号处理器(DSP)来实现根据本发明实施例的一些或者全部部件的一些或者全部功能。本发明还可以实现为用于执行这里所描述的方法的一部分或者全部的设备或者装置程序(例如,计算机程序和计算机程序产品)。这样的实现本发明的程序可以存储在计算机可读介质上,或者可以具有一个或者多个信号的形式。这样的信号可以从因特网网站上下载得到,或者在载体信号上提供,或者以任何其他形式提供。

[0163] 应该注意的是上述实施例对本发明进行说明而不是对本发明进行限制,并且本领域技术人员在不脱离所附权利要求的范围的情况下可设计出替换实施例。在权利要求中,不应将位于括号之间的任何参考符号构造成对权利要求的限制。单词“包含”不排除存在未列在权利要求中的元件或步骤。位于元件之前的单词“一”或“一个”不排除存在多个这样的元件。本发明可以借助于包括有若干不同元件的硬件以及借助于适当编程的计算机来实现。在列举了若干装置的单元权利要求中,这些装置中的若干个可以是通过同一个硬件项来具体体现。单词第一、第二、以及第三等的使用不表示任何顺序。可将这些单词解释为名称。上述实施例中的步骤,除有特殊说明外,不应理解为对执行顺序的限定。

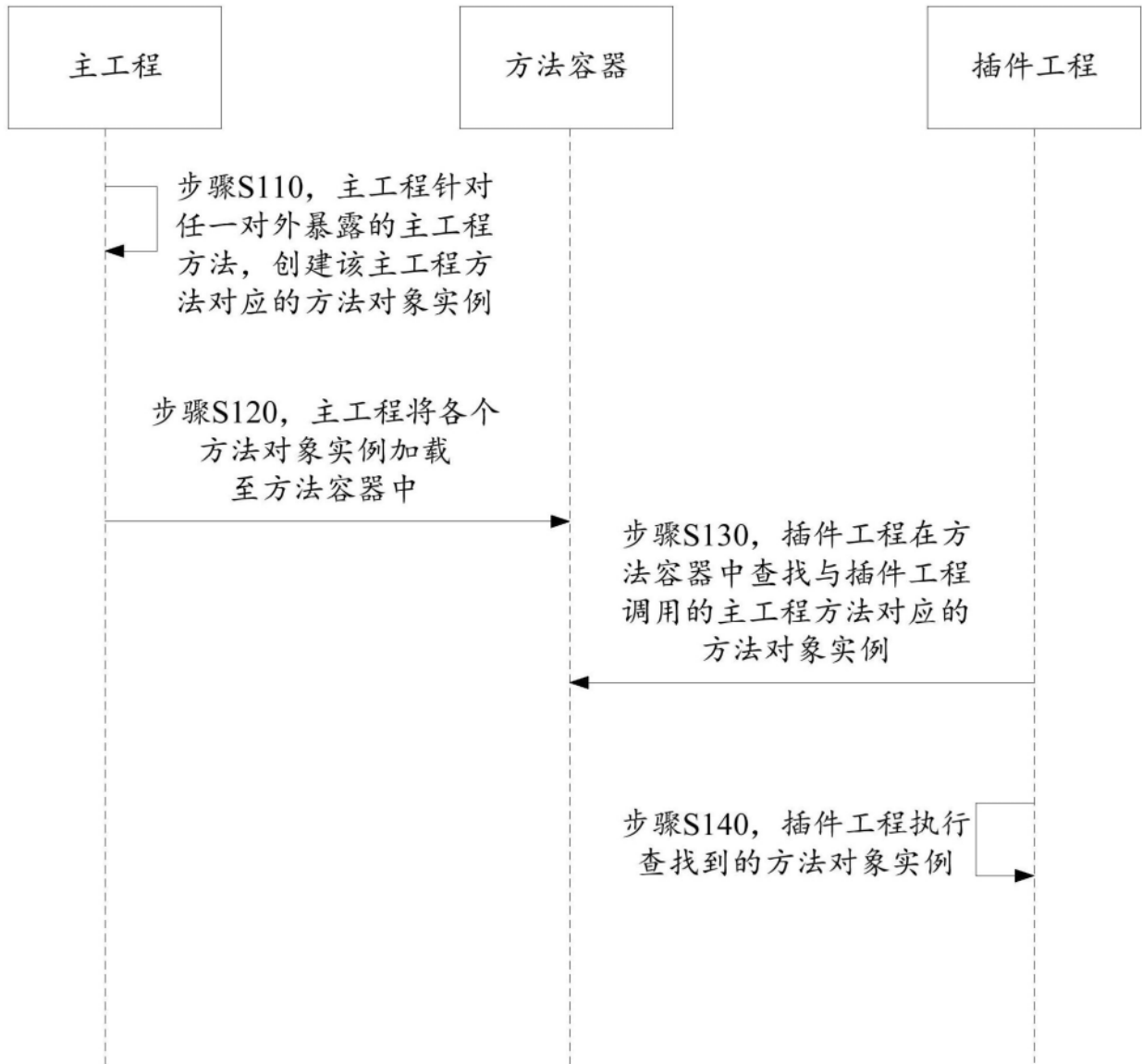


图1

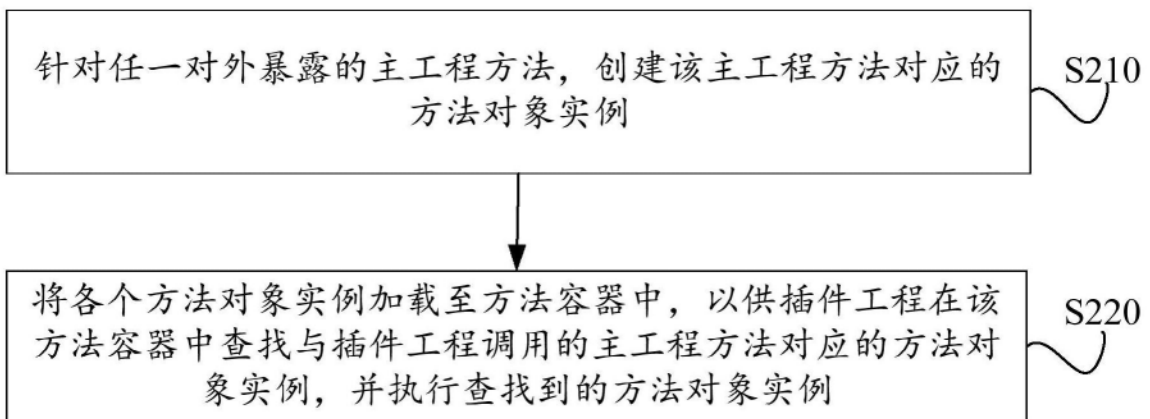


图2

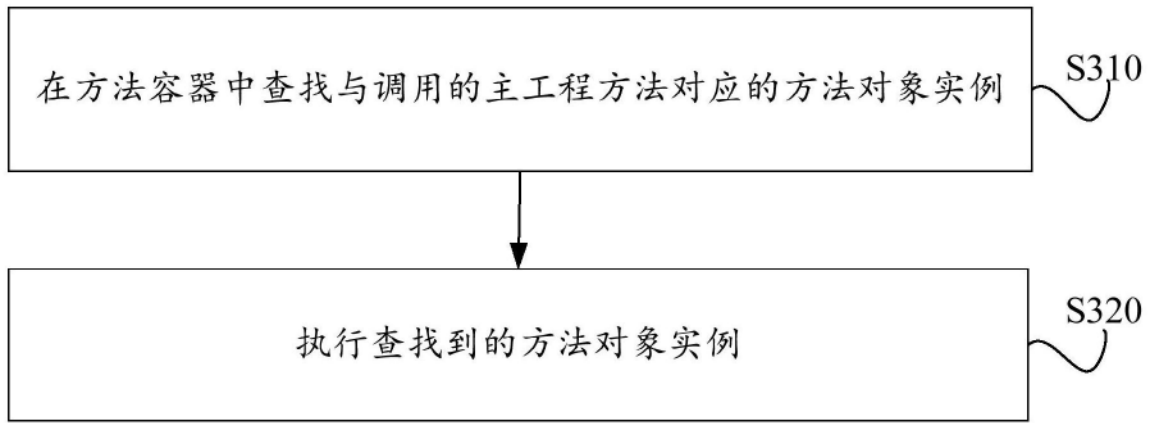


图3

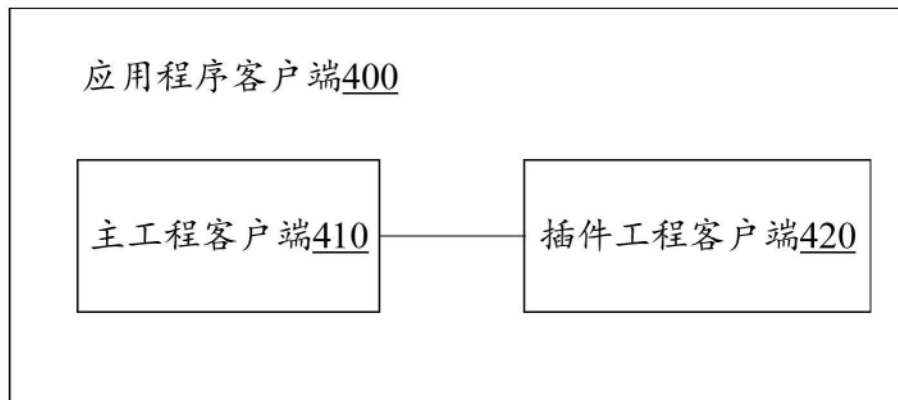


图4

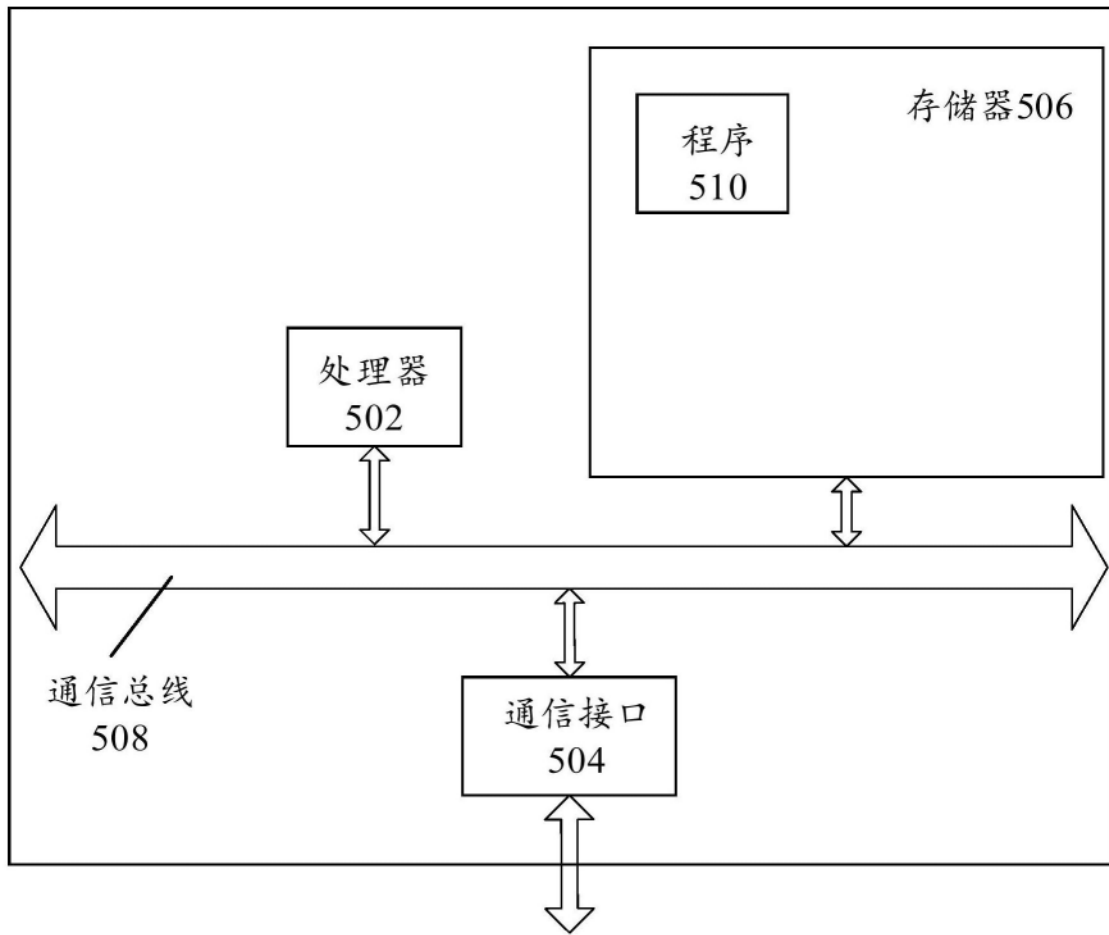


图5