



(12) 发明专利

(10) 授权公告号 CN 111562909 B

(45) 授权公告日 2024. 04. 02

(21) 申请号 202010375199.1

(22) 申请日 2020.05.06

(65) 同一申请的已公布的文献号
申请公布号 CN 111562909 A

(43) 申请公布日 2020.08.21

(73) 专利权人 北京金堤科技有限公司
地址 100086 北京市海淀区知春路65号中
国卫星通信大厦B座23层

(72) 发明人 周强 柳超

(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 刘云贵

(51) Int. Cl.

G06F 8/30 (2018.01)

G06F 9/445 (2018.01)

(56) 对比文件

CN 101206571 A, 2008.06.25

CN 109561104 A, 2019.04.02

CN 110569282 A, 2019.12.13

CN 110795147 A, 2020.02.14

GB 0329246 D0, 2004.01.21

US 2009248887 A1, 2009.10.01

US 2012077546 A1, 2012.03.29

剧忻;苗放.基于MINA开发高性能网络应用程序——以实现XMPP协议Openfire3.3.3为例.重庆工学院学报(自然科学版).2008,(10),全文.

审查员 李思彤

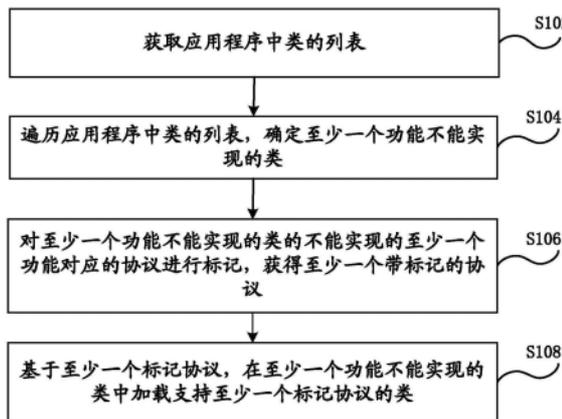
权利要求书2页 说明书11页 附图3页

(54) 发明名称

基于类的应用实现方法和装置、电子设备和存储介质

(57) 摘要

本公开的实施例公开了一种基于类的应用实现方法和装置、电子设备和存储介质,其中,方法包括:获取应用程序中类的列表;遍历所述应用程序中类的列表,确定至少一个功能不能实现的类;对所述至少一个功能不能实现的类的不能实现的至少一个功能对应的协议进行标记,获得所述至少一个带标记的协议;基于至少一个带标记的协议,在至少一个功能不能实现的类中加载支持至少一个带标记的协议的类。由此,本公开实施例利用加载支持带标记的协议的类的方式实现了类的多继承,避免了代码冗余,方便了后期协议功能的改动。



1. 一种基于类的应用实现方法,其特征在于,包括:
 - 获取应用程序中类的列表;
 - 遍历所述应用程序中类的列表,确定至少一个功能不能实现的类;
 - 对所述至少一个功能不能实现的类的不能实现的至少一个功能对应的协议进行标记,获得所述至少一个带标记的协议;
 - 基于所述至少一个带标记的协议,确定所述至少一个带标记的协议对应的代码;
 - 基于所述至少一个带标记的协议对应的代码,生成支持所述至少一个带标记协议的类;
 - 在所述至少一个功能不能实现的类中加载支持所述至少一个带标记的协议的类。
2. 根据权利要求1所述的方法,其特征在于,所述类的列表包括:各类需要实现的至少一个功能、各类支持的协议、以及各协议对应的功能。
3. 根据权利要求1所述的方法,其特征在于,所述基于所述至少一个带标记的协议,确定所述至少一个带标记的协议对应的代码,包括:
 - 基于所述至少一个带标记的协议,获取所述至少一个带标记的协议支持的功能;
 - 基于所述至少一个带标记的协议支持的功能,确定所述至少一个带标记的协议支持的至少一个子功能;
 - 基于所述至少一个带标记的协议支持的至少一个子功能,确定所述至少一个带标记的协议对应的代码。
4. 根据权利要求3所述的方法,其特征在于,所述基于所述至少一个带标记的协议支持的功能,确定所述至少一个带标记的协议支持的至少一个子功能和所述基于所述至少一个带标记的协议支持的至少一个子功能,确定所述至少一个带标记的协议对应的代码之间,包括:
 - 基于所述至少一个带标记的协议支持的至少一个子功能,确定各子功能的属性;其中所述各子功能的属性包括:必须实现的功能和可能实现的功能;
 - 基于所述各子功能的属性,确定所述各子功能对应的必须实现方法或可能实现的方法。
5. 根据权利要求3或4所述的方法,其特征在于,所述基于所述至少一个带标记的协议支持的至少一个子功能,确定所述至少一个带标记的协议对应的代码,包括:
 - 基于所述至少一个带标记的协议支持的至少一个子功能,确定各子功能对应的实例方法和/或类方法;
 - 基于所述各子功能对应的实例方法和/或类方法,确定所述至少一个带标记的协议对应的代码。
6. 一种基于类的应用实现装置,其特征在于,包括:
 - 获取模块,用于获取应用程序中类的列表;
 - 确定模块,用于遍历所述应用程序中类的列表,确定至少一个功能不能实现的类;
 - 获得模块,用于对所述至少一个功能不能实现的类的不能实现的至少一个功能对应的协议进行标记,获得所述至少一个带标记的协议;
 - 加载模块,用于基于所述至少一个带标记的协议,在所述至少一个功能不能实现的类中加载支持所述至少一个带标记的协议的类;

其中,所述加载模块,包括:

确定单元,用于基于所述至少一个带标记的协议,确定所述至少一个带标记的协议对应的代码;

生成单元,用于基于所述至少一个带标记的协议对应的代码,生成支持所述至少一个带标记的协议的类;

加载单元,用于在所述至少一个功能不能实现的类中加载支持所述至少一个带标记的协议的类。

7.根据权利要求6所述的装置,其特征在于,所述确定单元包括:

获取子单元,用于基于所述至少一个带标记的协议,获取所述至少一个带标记的协议支持的功能;

第一确定子单元,用于基于所述至少一个标记协议支持的功能,确定所述至少一个标记协议支持的多个子功能;

第四确定子单元,用于基于所述至少一个带标记的协议支持的多个子功能,确定所述至少一个带标记的协议对应的代码。

8.根据权利要求7所述的装置,其特征在于,所述第一确定子单元和所述第四确定子单元之间,包括:

第二确定子单元,用于基于所述至少一个带标记的协议支持的至少一个子功能,确定各子功能的属性;其中所述各子功能的属性包括:必须实现的功能和可能实现的功能;

第三确定子单元,用于基于所述各子功能的属性,确定所述各子功能对应的必须实现方法或可能实现的方法。

9.根据权利要求7或8所述的装置,其特征在于,所述第四确定子单元,具体用于:

基于所述至少一个带标记的协议支持的至少一个子功能,确定各子功能对应的实例方法和/或类方法;

基于所述各子功能对应的实例方法和/或类方法,确定所述至少一个带标记的协议对应的代码。

10.一种计算机可读存储介质,其特征在于,所述存储介质存储有计算机程序,所述计算机程序用于执行上述权利要求1-5任一所述的基于类的应用实现方法。

11.一种电子设备,其特征在于,所述电子设备包括:

处理器;

用于存储所述处理器可执行指令的存储器;

所述处理器,用于从所述存储器中读取所述可执行指令,并执行所述指令以实现上述权利要求1-5任一所述的基于类的应用实现方法。

基于类的应用实现方法和装置、电子设备和存储介质

技术领域

[0001] 本公开涉及应用开发技术,尤其是一种基于类的应用实现方法和装置、电子设备和存储介质。

背景技术

[0002] 在程序开发过程中用到大部分高级语言都存在着类的概念,类可以对功能进行封装、拆分,使代码变得格式化,模块化,且使用起来更安全。如果想要拥有某个类的功能,就需要继承自这个类,创建子类来实现。如果需要拥有多个类的功能,就需要通过多继承,同时继承多个类来实现。

[0003] 在实现本公开的过程中,发明人发现:有些程序语言在用于IOS操作系统时,会出现不支持类的多继承,例如,Objective-C语言等,这些程序语言很容易造成代码冗余,例如:类A需要有多个协议对应的功能,就需要声明其支持这些协议,并对每个协议对应的功能进行编码同理,类B需要支持这些功能的时候,也需要进行编码,以此类推,不仅会造成代码冗余,而且也会给后期协议功能的改动带来很多不便。

发明内容

[0004] 为了解决上述技术问题,提出了本公开。本公开的实施例提供了一种基于类的应用实现方法和装置、电子设备和存储介质。

[0005] 根据本公开实施例的一个方面,提供了一种基于类的应用实现方法,包括:

[0006] 获取应用程序中类的列表;

[0007] 遍历所述应用程序中类的列表,确定至少一个功能不能实现的类;

[0008] 对所述至少一个功能不能实现的类的不能实现的至少一个功能对应的协议进行标记,获得所述至少一个带标记的协议;

[0009] 基于所述至少一个带标记的协议,在所述至少一个功能不能实现的类中加载支持所述至少一个带标记的协议的类。

[0010] 可选地,在本公开上述各方法实施例中,所述类的列表包括:各类需要实现的至少一个功能、各类支持的协议、以及各协议对应的功能。

[0011] 可选地,在本公开上述各方法实施例中,所述基于所述至少一个带标记的协议,在所述至少一个功能不能实现的类中加载支持所述至少一个带标记的协议的类,包括:

[0012] 基于所述至少一个带标记的协议,确定所述至少一个带标记的协议对应的代码;

[0013] 基于所述至少一个带标记的协议对应的代码,生成支持所述至少一个带标记协议的类;

[0014] 在所述至少一个功能不能实现的类中加载支持所述至少一个带标记的协议的类。

[0015] 可选地,在本公开上述各方法实施例中,所述基于所述至少一个带标记的协议,确定所述至少一个带标记的协议对应的代码,包括:

[0016] 基于所述至少一个带标记的协议,获取所述至少一个带标记的协议支持的功能;

[0017] 基于所述至少一个带标记的协议支持的功能,确定所述至少一个带标记的协议支持的至少一个子功能;

[0018] 基于所述至少一个带标记的协议支持的至少一个子功能,确定所述至少一个带标记的协议对应的代码。

[0019] 可选地,在本公开上述各方法实施例中,所述基于所述至少一个带标记的协议支持的功能,确定所述至少一个带标记的协议支持的至少一个子功能和所述基于所述至少一个带标记的协议支持的至少一个子功能,确定所述至少一个带标记的协议对应的代码之间,包括:

[0020] 基于所述至少一个带标记的协议支持的至少一个子功能,确定各子功能的属性;其中所述各子功能的属性包括:必须实现的功能和可能实现的功能;

[0021] 基于所述各子功能的属性,确定所述各子功能对应的必须实现方法或可能实现的方法。

[0022] 可选地,在本公开上述各方法实施例中,所述基于所述至少一个带标记的协议支持的至少一个子功能,确定所述至少一个带标记的协议对应的代码,包括:

[0023] 基于所述至少一个带标记的协议支持的至少一个子功能,确定各子功能对应的实例方法和/或类方法;

[0024] 基于所述各子功能对应的实例方法和/或类方法,确定所述至少一个带标记的协议对应的代码。

[0025] 根据本公开实施例的另一个方面,提供了一种基于类的应用实现装置,包括:

[0026] 获取模块,用于获取应用程序中类的列表;确定模块,用于遍历所述应用程序中类的列表,确定至少一个功能不能实现的类;

[0027] 获得模块,用于对所述至少一个功能不能实现的类的不能实现的至少一个功能对应的协议进行标记,获得所述至少一个带标记的协议;

[0028] 加载模块,用于基于所述至少一个带标记的协议,在所述至少一个功能不能实现的类中加载支持所述至少一个带标记的协议的类。

[0029] 可选地,在本公开上述各装置实施例中,所述类的列表包括:各类需要实现的至少一个功能、各类支持的协议、以及各协议对应的功能。

[0030] 可选地,在本公开上述各装置实施例中,所述加载模块,包括:

[0031] 确定单元,用于基于所述至少一个带标记的协议,确定所述至少一个带标记的协议对应的代码;

[0032] 生成单元,用于基于所述至少一个带标记的协议对应的代码,生成支持所述至少一个带标记的协议的类;

[0033] 加载单元,用于在所述至少一个功能不能实现的类中加载支持所述至少一个带标记的协议的类。

[0034] 可选地,在本公开上述各装置实施例中,所述确定单元包括:

[0035] 获取子单元,用于基于所述至少一个带标记的协议,获取所述至少一个带标记的协议支持的功能;

[0036] 第一确定子单元,用于基于所述至少一个标记协议支持的功能,确定所述至少一个标记协议支持的多个子功能;

[0037] 第四确定子单元,用于基于所述至少一个带标记的协议支持的多个子功能,确定所述至少一个带标记的协议对应的代码。

[0038] 可选地,在本公开上述各装置实施例中,所述第一确定子单元和所述第四确定子单元之间,包括:

[0039] 第二确定子单元,用于基于所述至少一个带标记的协议支持的至少一个子功能,确定各子功能的属性;其中所述各子功能的属性包括:必须实现的功能和可能实现的功能;

[0040] 第三确定子单元,用于基于所述各子功能的属性,确定所述各子功能对应的必须实现方法或可能实现的方法。

[0041] 可选地,在本公开上述各装置实施例中,所述第四确定子单元,具体用于:

[0042] 基于所述至少一个带标记的协议支持的至少一个子功能,确定各子功能对应的实例方法和/或类方法;

[0043] 基于所述各子功能对应的实例方法和/或类方法,确定所述至少一个带标记的协议对应的代码。

[0044] 根据本公开实施例的又一个方面,提供了一种计算机可读存储介质,该存储介质存储有计算机程序,所述计算机程序用于执行本公开上述任一实施例所述的基于类的应用实现方法。

[0045] 根据本公开实施例的又一个方面,提供了一种电子设备,该电子设备包括:处理器;用于存储所述处理器可执行指令的存储器;所述处理器,用于从所述存储器中读取所述可执行指令,并执行所述指令以实现上述任一实施例所述的基于类的应用实现方法。

[0046] 本公开上述实施例提供的基于类的应用实现方法和装置、电子设备和存储介质,获取应用程序中类的列表;遍历所述应用程序中类的列表,确定至少一个功能不能实现的类;对所述至少一个功能不能实现的类的不能实现的至少一个功能对应的协议进行标记,获得所述至少一个带标记的协议;基于至少一个带标记的协议,在至少一个功能不能实现的类中加载支持至少一个带标记的协议的类。由此,本公开实施例利用加载支持带标记的协议的类的方式实现了类的多继承,避免了代码冗余,方便了后期协议功能的改动。

[0047] 下面通过附图和实施例,对本公开的技术方案做进一步的详细描述。

附图说明

[0048] 通过结合附图对本公开实施例进行更详细的描述,本公开的上述以及其他目的、特征和优势将变得更加明显。附图用来提供对本公开实施例的进一步理解,并且构成说明书的一部分,与本公开实施例一起用于解释本公开,并不构成对本公开的限制。在附图中,相同的参考标号通常代表相同部件或步骤。

[0049] 图1是本公开一示例性实施例提供的基于类的应用实现方法的流程示意图。

[0050] 图2是本公开另一示例性实施例提供的基于类的应用实现方法的流程示意图。

[0051] 图3是本公开又一示例性实施例提供的基于类的应用实现方法的流程示意图。

[0052] 图4是本公开再一示例性实施例提供的基于类的应用实现方法的流程示意图。

[0053] 图5是本公开另一示例性实施例提供的基于类的应用实现方法的流程示意图。

[0054] 图6是本公开一示例性实施例提供的基于类的应用实现装置的结构示意图。

[0055] 图7是本公开一示例性实施例提供的电子设备的结构图。

具体实施方式

[0056] 下面,将参考附图详细地描述根据本公开的示例实施例。显然,所描述的实施例仅仅是本公开的一部分实施例,而不是本公开的全部实施例,应理解,本公开不受这里描述的示例实施例的限制。

[0057] 应注意到:除非另外具体说明,否则在这些实施例中阐述的部件和步骤的相对布置、数字表达式和数值不限制本公开的范围。

[0058] 本领域技术人员可以理解,本公开实施例中的“第一”、“第二”等术语仅用于区别不同步骤、设备或模块等,既不代表任何特定技术含义,也不表示它们之间的必然逻辑顺序。

[0059] 还应理解,在本公开实施例中,“多个”可以指两个或两个以上,“至少一个”可以指一个、两个或两个以上。

[0060] 还应理解,对于本公开实施例中提及的任一部件、数据或结构,在没有明确限定或者在前后文给出相反启示的情况下,一般可以理解为一个或多个。

[0061] 另外,本公开中术语“和/或”,仅仅是一种描述关联对象的关联关系,表示可以存在三种关系,例如,A和/或B,可以表示:单独存在A,同时存在A和B,单独存在B这三种情况。另外,本公开中字符“/”,一般表示前后关联对象是一种“或”的关系。

[0062] 还应理解,本公开对各个实施例的描述着重强调各个实施例之间的不同之处,其相同或相似之处可以相互参考,为了简洁,不再一一赘述。

[0063] 同时,应当明白,为了便于描述,附图中所示出的各个部分的尺寸并不是按照实际的比例关系绘制的。

[0064] 以下对至少一个示例性实施例的描述实际上仅仅是说明性的,决不作为对本公开及其应用或使用的任何限制。

[0065] 对于相关领域普通技术人员已知的技术、方法和设备可能不作详细讨论,但在适当情况下,所述技术、方法和设备应当被视为说明书的一部分。

[0066] 应注意到:相似的标号和字母在下面的附图中表示类似项,因此,一旦某一项在一个附图中被定义,则在随后的附图中不需要对其进行进一步讨论。

[0067] 本公开实施例可以应用于终端设备、计算机系统、服务器等电子设备,其可与众多其它通用或专用计算系统环境或配置一起操作。适于与终端设备、计算机系统、服务器等电子设备一起使用的众所周知的终端设备、计算系统、环境和/或配置的例子包括但不限于:个人计算机系统、服务器计算机系统、瘦客户机、厚客户机、手持或膝上设备、基于微处理器的系统、机顶盒、可编程消费电子产品、网络个人电脑、小型计算机系统大型计算机系统和包括上述任何系统的分布式云计算技术环境,等等。

[0068] 终端设备、计算机系统、服务器等电子设备可以在由计算机系统执行的计算机系统可执行指令(诸如程序模块)的一般语境下描述。通常,程序模块可以包括例程、程序、目标程序、组件、逻辑、数据结构等等,它们执行特定的任务或者实现特定的抽象数据类型。计算机系统/服务器可以在分布式云计算环境中实施,分布式云计算环境中,任务是由通过通信网络链接的远程处理设备执行的。在分布式云计算环境中,程序模块可以位于包括存储设备的本地或远程计算系统存储介质上。

[0069] 图1是本公开一示例性实施例提供的基于类的应用实现方法流程图。本实施例可

应用在电子设备上,如图1所示,该基于类的应用实现方法包括如下步骤:

[0070] S102,获取应用程序中类的列表。

[0071] 在本公开的实施例中类的列表可以包括:各类需要实现的至少一个功能、各类支持的协议、以及各协议对应的功能。

[0072] S104,遍历应用程序中类的列表,确定至少一个功能不能实现的类。

[0073] 例如,遍历应用程序中类的列表,确定其中功能不能实现的类有:类A、类B、以及类C。

[0074] S106,对至少一个功能不能实现的类的不能实现的至少一个功能对应的协议进行标记,获得至少一个带标记的协议。

[0075] 例如,对类A中不能实现的功能,即非继承功能Y和Z对应的协议进行标记,获得两个带标记的协议。

[0076] S108,基于至少一个带标记的协议,在至少一个功能不能实现的类中加载支持至少一个带标记的协议的类。

[0077] 在一个具体的例子中,类A需要实现功能X、Y、Z,其中,继承功能X对应协议MyProtocol、对非继承功能Y和Z对应的协议进行标记,获得两个带标记的协议Protocol Y'和Procotol Z',利用Objective-C的运行机制,在主函数(main函数)执行时,加载两个带标记的协议Protocol Y'和Procotol Z'对应的类,具体执行程序如下所示:

[0078] @interface MyClass

[0079] <MyProtocol>//只需加入该行代码,引入协议声明即可

[0080] <Protocol H',Procotol K' ...>//如需多继承,则加入多个协议即可

[0081] ...

[0082] @end

[0083] 本公开上述实施例提供的基于类的应用实现方法,本公开上述实施例提供的获取应用程序中类的列表;遍历应用程序中类的列表,确定至少一个功能不能实现的类;对至少一个功能不能实现的类的不能实现的至少一个功能对应的协议进行标记,获得至少一个带标记的协议;基于至少一个带标记的协议,在至少一个功能不能实现的类中加载支持至少一个带标记的协议的类。由此,本公开实施例利用加载支持带标记的协议的类的方式实现了类的多继承,避免了代码冗余,方便了后期协议功能的改动。

[0084] 如图2所示,在上述图1所示实施例的基础上,其中一些可选的实施方式中,步骤S108具体可以包括:

[0085] S201,基于至少一个带标记的协议,确定至少一个带标记的协议对应的代码。

[0086] S202,基于至少一个带标记的协议对应的代码,生成支持至少一个带标记的协议的类。

[0087] 其中,可以将协议名和计数器对应的数目进行拼接,来确定各类的类名,例如,可以利用宏定义把协议名称和计数器的计数次数(_COUNTER_)进行拼接,以生成类名(如:MyProtocol_DynamicClass_1):

[0088] @protocol MyProtocol;

[0089] @interface MyProtocol_DynamicClass_1:NSObject<MyProtocol>

[0090] @end

[0091] S203,在至少一个功能不能实现的类中加载支持至少一个带标记的协议的类。

[0092] 例如,如果类B需要多个协议的功能,只需要标记支持这些协议即可,然后将对应承载这些协议功能的类都加载到类B中。

[0093] 本公开实施例可以在应用程序运行前,直接加载带标记的协议对应的类,避免了代码冗余导致的程序运行崩溃。

[0094] 如图3所示,在上述图2所示实施例的基础上,其中一些可选的实施方式中,步骤S201具体可以包括:

[0095] S301,基于至少一个带标记的协议,获取至少一个带标记的协议支持的功能。

[0096] 例如,当类A标记其支持协议H',则需要实现协议对应的功能Y,当其他模块使用类A的时候就知道包括协议H'对应的功能Y,从而可以使用功能Y。

[0097] S302,基于至少一个带标记的协议支持的功能,确定至少一个带标记的协议支持的至少一个子功能。

[0098] S303,基于至少一个带标记的协议支持的至少一个子功能,确定至少一个带标记的协议对应的代码。

[0099] 在一个可选的示例中,列出一个带标记的协议支持的功能所支持的全部子功能,获取各子功能对应的子代码,确定至少一个带标记的协议对应的代码。

[0100] 如图4所示,在上述图3所示实施例的基础上,其中一些可选的实施方式中,步骤S302和S303之间具体可以包括:

[0101] S401,基于至少一个带标记的协议支持的至少一个子功能,确定各子功能的属性。

[0102] 其中,各子功能的属性包括:必须实现的功能和可能实现的功能。

[0103] S402,基于各子功能的属性,确定各子功能对应的必须实现方法或可能实现的方法。

[0104] 例如,在以下示例的程序中,在实例(void)foo之前加@required,表示必须实现的方法;在实例(void)bar之前加@optional可选择实现的方法:

```
[0105] @protocol MyProtocol
```

```
[0106] @required
```

```
[0107] -(void)foo;
```

```
[0108] @optional
```

```
[0109] -(void)bar;
```

```
[0110] @end
```

[0111] 在一个如下所示的具体例子中,类MyProtocol_DynamicClass_1中包括必须实现的功能对应的必须实现的方法(void)foo和可选择实现的功能对应的可选择实现的方法(void)bar:

```
    @implementation MyProtocol_DynamicClass-1
    -(void) foo
    {
        ...
    }
    -(void) bar
    {
        ...
    }
    @end
```

[0114] 本公开上述实施例可以实现动态构造一个类以承载协议的声明支持的功能,从而使得协议功能对应的类能够简洁、高效的接入父类中。

[0115] 如图5所示,在上述图4所示实施例的基础上,其中一些可选的实施方式中,步骤S303具体可以包括:

[0116] S501,基于至少一个带标记的协议支持的至少一个子功能,确定各子功能对应的实例方法和/或类方法。

[0117] 其中,类方法用于表示类对应的功能,例如:信息交互类对应的信息交互功能;实例方法用于表示实例对应的功能,例如:即时消息对应的文字输入功能、语音输入功能、以及图片输入功能等。

[0118] S502,基于各子功能对应的实例方法和/或类方法,确定至少一个带标记的协议对应的代码。

[0119] 在如下所示的程序中,利用Objective-C语言本身的特性和ios的动态链接器系统(dyld)的加载时序,在带标记的协议的类中加载(load)实例方法和/或类方法,以在至少一个功能不能实现的类进行用加载时获取到对应的实例方法和/或类方法并保存:

```
+ (void) load
```

```
{
```

```
...
```

1. 获取类的实例方法列表。

[0120]

2. 获取类的类方法列表。

3. 把该协议和协议的方法实现形成映射关系，并保存下来。

```
...
```

```
}
```

[0121] 本公开实施例通过上述基于实例方法和类方法确定的协议和协议方法之间的映射关系，可以准确快速地实现类的多继承，提高运行速率。

[0122] 本公开实施例提供的任一种基于类的应用实现方法可以由任意适当的具有数据处理能力的设备执行，包括但不限于：终端设备和服务器等。或者，本公开实施例提供的任一种基于类的应用实现方法可以由处理器执行，如处理器通过调用存储器存储的相应指令来执行本公开实施例提及的任一种基于类的应用实现方法。下文不再赘述。

[0123] 本公开实施例提供的任一种基于类的应用实现方法可以由任意适当的具有数据处理能力的设备执行，包括但不限于：终端设备和服务器等。或者，本公开实施例提供的任一种基于类的应用实现方法可以由处理器执行，如处理器通过调用存储器存储的相应指令来执行本公开实施例提及的任一种基于类的应用实现方法。下文不再赘述。

[0124] 图6是本公开一示例性实施例提供的基于类的应用实现装置的结构示意图。该装置可以设置于终端设备、服务器等电子设备中，执行本公开上述任一实施例的基于类的应用实现方法。如图6所示，该基于类的应用实现装置包括：

[0125] 获取模块61，用于获取应用程序中类的列表；

[0126] 确定模块62，用于遍历所述应用程序中类的列表，确定至少一个功能不能实现的类；

[0127] 获得模块63，用于对所述至少一个功能不能实现的类的不能实现的至少一个功能对应的协议进行标记，获得所述至少一个带标记的协议；

[0128] 加载模块64，用于基于所述至少一个带标记的协议，在所述至少一个功能不能实现的类中加载支持所述至少一个带标记的协议的类。

[0129] 本公开上述实施例提供的基于类的应用实现装置，获取应用程序中类的列表；遍历所述应用程序中类的列表，确定至少一个功能不能实现的类；对所述至少一个功能不能实现的类的不能实现的至少一个功能对应的协议进行标记，获得所述至少一个带标记的协议；基于至少一个带标记的协议，在至少一个功能不能实现的类中加载支持至少一个带标记的协议的类。由此，本公开实施例利用加载支持带标记的协议的类的方式实现了类的多继承，避免了代码冗余，方便了后期协议功能的改动。

[0130] 在其中一些实施方式中，所述类的列表包括：各类需要实现的至少一个功能、各类

支持的协议、以及各协议对应的功能。

[0131] 在其中一些实施方式中,所述加载模块64,包括:

[0132] 确定单元,用于基于所述至少一个带标记的协议,确定所述至少一个带标记的协议对应的代码;

[0133] 生成单元,用于基于所述至少一个带标记的协议对应的代码,生成支持所述至少一个带标记的协议的类;

[0134] 加载单元,用于在所述至少一个功能不能实现的类中加载支持所述至少一个带标记的协议的类。

[0135] 在其中一些实施方式中,所述确定单元包括:

[0136] 获取子单元,用于基于所述至少一个带标记的协议,获取所述至少一个带标记的协议支持的功能;

[0137] 第一确定子单元,用于基于所述至少一个标记协议支持的功能,确定所述至少一个标记协议支持的多个子功能;

[0138] 第四确定子单元,用于基于所述至少一个带标记的协议支持的多个子功能,确定所述至少一个带标记的协议对应的代码。

[0139] 在其中一些实施方式中,所述第一确定子单元和所述第四确定子单元之间,包括:

[0140] 第二确定子单元,用于基于所述至少一个带标记的协议支持的至少一个子功能,确定各子功能的属性;其中所述各子功能的属性包括:必须实现的功能和可能实现的功能;

[0141] 第三确定子单元,用于基于所述各子功能的属性,确定所述各子功能对应的必须实现方法或可能实现的方法。

[0142] 在其中一些实施方式中,所述第四确定子单元,具体用于:

[0143] 基于所述至少一个带标记的协议支持的至少一个子功能,确定各子功能对应的实例方法和/或类方法;

[0144] 基于所述各子功能对应的实例方法和/或类方法,确定所述至少一个带标记的协议对应的代码。

[0145] 另外,本公开实施例还提供了一种电子设备,该电子设备包括:处理器;用于存储所述处理器可执行指令的存储器;

[0146] 所述处理器,用于从所述存储器中读取所述可执行指令,并执行所述指令以实现本公开上述任一实施例所述的基于类的应用实现方法。

[0147] 下面,参考图7来描述根据本公开实施例的电子设备。该电子设备可以是第一设备和第二设备中的任一个或两者、或与它们独立的单机设备,该单机设备可以与第一设备和第二设备进行通信,以从它们接收所采集到的输入信号。图7图示了根据本公开实施例的电子设备的框图。如图7所示,电子设备包括一个或多个处理器71和存储器72。

[0148] 处理器71可以是中央处理单元(CPU)或者具有数据处理能力和/或指令执行能力的其他形式的处理单元,并且可以控制电子设备中的其他组件以执行期望的功能。

[0149] 存储器72可以包括一个或多个计算机程序产品,所述计算机程序产品可以包括各种形式的计算机可读存储介质,例如易失性存储器和/或非易失性存储器。所述易失性存储器例如可以包括随机存取存储器(RAM)和/或高速缓冲存储器(cache)等。所述非易失性存储器例如可以包括只读存储器(ROM)、硬盘、闪存等。在所述计算机可读存储介质上可以存

储一个或多个计算机程序指令,处理器71可以运行所述程序指令,以实现上文所述的本公开的各个实施例的软件程序的基于类的应用实现方法以及/或者其他期望的功能。在一个示例中,电子设备还可以包括:输入装置73和输出装置74,这些组件通过总线系统和/或其他形式的连接机构(未示出)互连。

[0150] 此外,该输入设备73还可以包括例如键盘、鼠标等等。

[0151] 该输出装置74可以向外部输出各种信息。该输出设备74可以包括例如显示器、扬声器、打印机、以及通信网络及其所连接的远程输出设备等等。

[0152] 当然,为了简化,图7中仅示出了该电子设备中与本公开有关的组件中的一些,省略了诸如总线、输入/输出接口等等的组件。除此之外,根据具体应用情况,电子设备还可以包括任何其他适当的组件。

[0153] 除了上述方法和设备以外,本公开的实施例还可以是计算机程序产品,其包括计算机程序指令,所述计算机程序指令在被处理器运行时使得所述处理器执行本说明书上述各种实施例的软件程序的基于类的应用实现方法中的步骤。

[0154] 所述计算机程序产品可以以一种或多种程序设计语言的任意组合来编写用于执行本公开实施例操作的程序代码,所述程序设计语言包括面向对象的程序设计语言,诸如Java、C++等,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设备上执行、部分地在用户设备上执行、作为一个独立的软件包执行、部分在用户计算设备上部分在远程计算设备上执行、或者完全在远程计算设备或服务器上执行。

[0155] 此外,本公开的实施例还可以是计算机可读存储介质,其上存储有计算机程序指令,所述计算机程序指令在被处理器运行时使得所述处理器执行本说明书上述各种实施例的软件程序的基于类的应用实现方法中的步骤。

[0156] 所述计算机可读存储介质可以采用一个或多个可读介质的任意组合。可读介质可以是可读信号介质或者可读存储介质。可读存储介质例如可以包括但不限于电、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。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。

[0157] 以上结合具体实施例描述了本公开的基本原理,但是,需要指出的是,在本公开中提及的优点、优势、效果等仅是示例而非限制,不能认为这些优点、优势、效果等是本公开的各个实施例必须具备的。另外,上述公开的具体细节仅是为了示例的作用和便于理解的作用,而非限制,上述细节并不限制本公开为必须采用上述具体的细节来实现。

[0158] 本说明书中各个实施例均采用递进的方式描述,每个实施例重点说明的都是与其它实施例的不同之处,各个实施例之间相同或相似的部分相互参见即可。对于系统实施例而言,由于其与方法实施例基本对应,所以描述的比较简单,相关之处参见方法实施例的部分说明即可。

[0159] 本公开中涉及的器件、装置、设备、系统的方框图仅作为例示性的例子并且不意图要求或暗示必须按照方框图示出的方式进行连接、布置、配置。如本领域技术人员将认识到的,可以按任意方式连接、布置、配置这些器件、装置、设备、系统。诸如“包括”、“包含”、“具

有”等等的词语是开放性词汇,指“包括但不限于”,且可与其互换使用。这里所使用的词汇“或”和“和”指词汇“和/或”,且可与其互换使用,除非上下文明确指示不是如此。这里所使用的词汇“诸如”指词组“诸如但不限于”,且可与其互换使用。

[0160] 可能以许多方式来实现本公开的方法和装置。例如,可通过软件、硬件、固件或者软件、硬件、固件的任何组合来实现本公开的方法和装置。用于所述方法的步骤的上述顺序仅是为了进行说明,本公开的方法的步骤不限于以上具体描述的顺序,除非以其它方式特别说明。此外,在一些实施例中,还可将本公开实施为记录在记录介质中的程序,这些程序包括用于实现根据本公开的方法的机器可读指令。因而,本公开还覆盖存储用于执行根据本公开的方法的程序的记录介质。

[0161] 还需要指出的是,在本公开的装置、设备和方法中,各部件或各步骤是可以分解和/或重新组合的。这些分解和/或重新组合应视为本公开的等效方案。

[0162] 提供所公开的方面的以上描述以使本领域的任何技术人员能够做出或者使用本公开。对这些方面的各种修改对于本领域技术人员而言是非常显而易见的,并且在此定义的一般原理可以应用于其他方面而不脱离本公开的范围。因此,本公开不意图被限制到在此示出的方面,而是按照与在此公开的原理和新颖的特征一致的最宽范围。

[0163] 为了例示和描述的目的已经给出了以上描述。此外,此描述不意图将本公开的实施例限制到在此公开的形式。尽管以上已经讨论了多个示例方面和实施例,但是本领域技术人员将认识到其某些变型、修改、改变、添加和子组合。

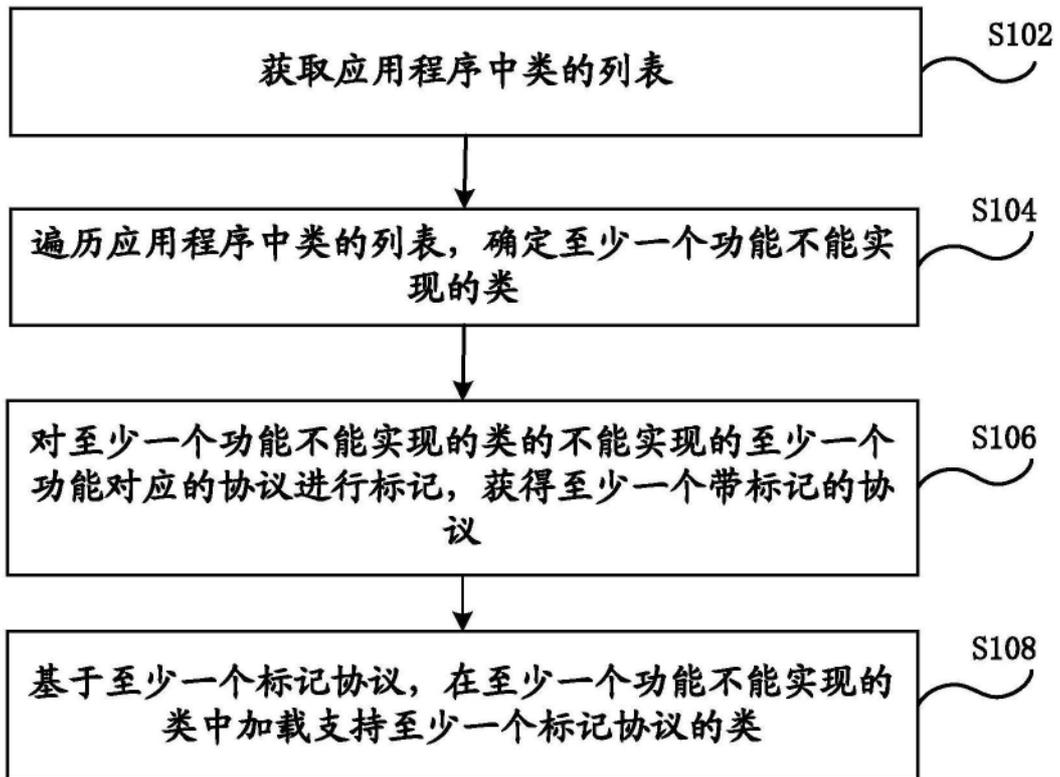


图1

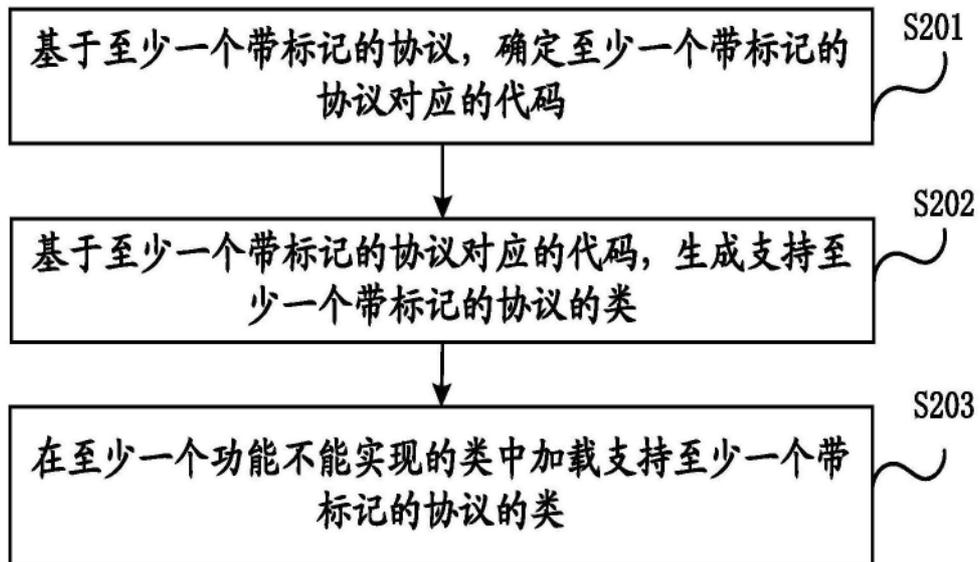
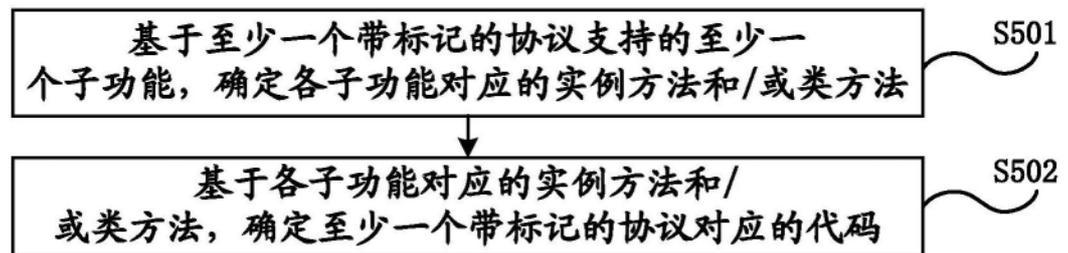
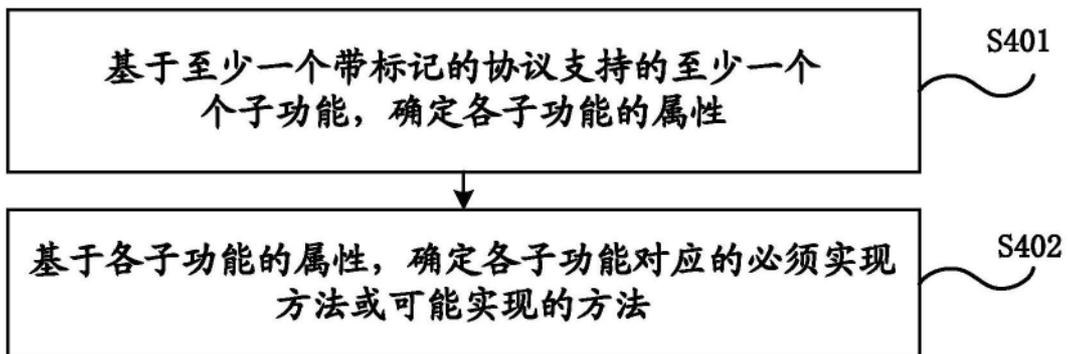
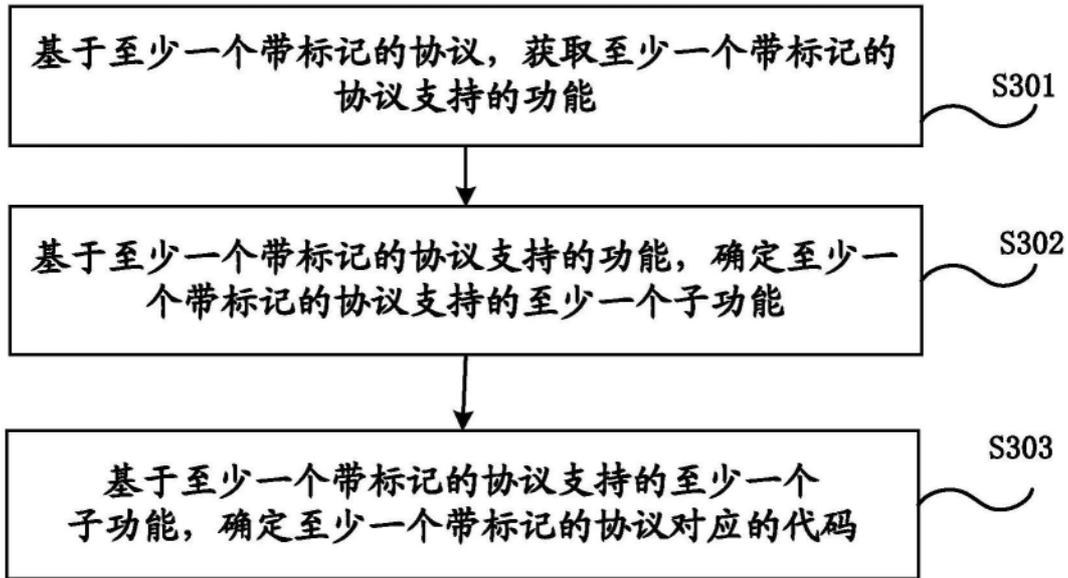


图2



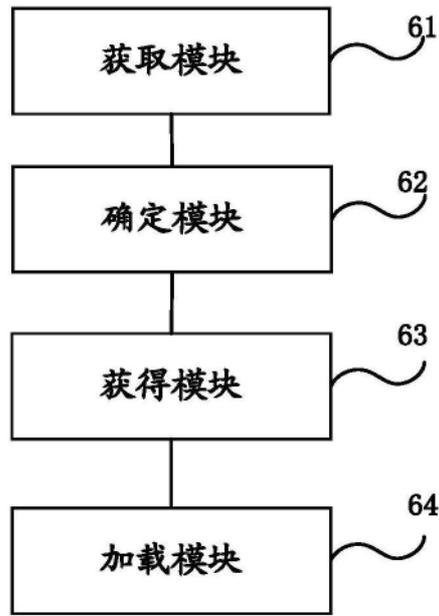


图6

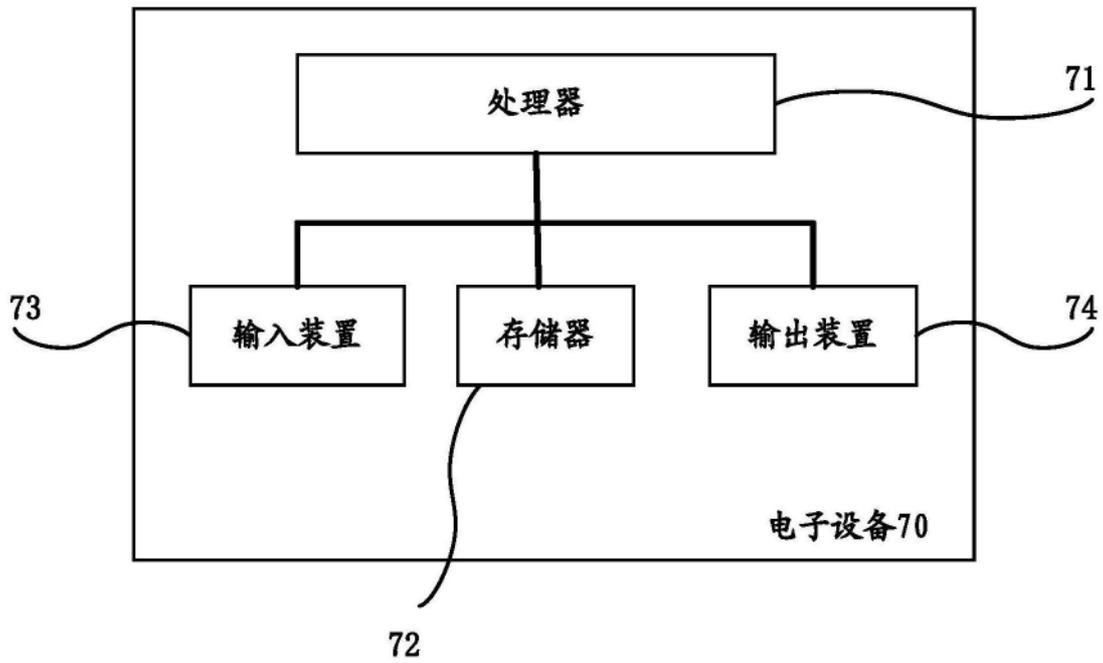


图7