(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 112835563 A
(43)申请公布日 2021.05.25

(21)申请号 202110101960.7

(22)申请日 2021.01.26

(71)申请人 广州欢网科技有限责任公司
地址 511400 广东省广州市番禺区小谷围街中二横路22号A513-A514

(72)发明人 宁炳剑

(74)专利代理机构 北京细软智谷知识产权代理有限责任公司 11471
代理人 牛晴

(51)Int.Cl.
G06F 8/30(2018.01)
G06F 8/36(2018.01)

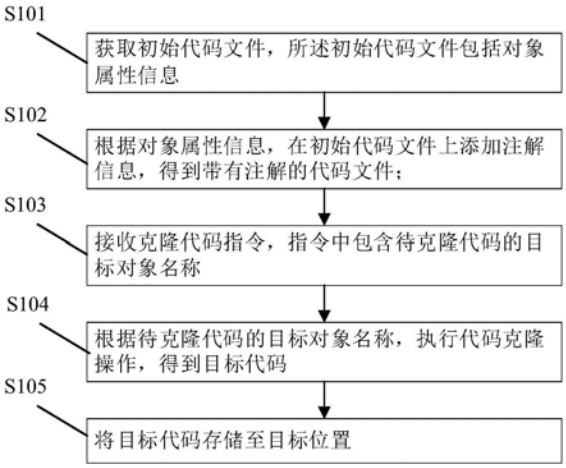
权利要求书1页 说明书8页 附图4页

(54)发明名称

一种代码克隆方法、装置、可读存储介质以及计算机

(57)摘要

本申请属于计算机技术领域，具体涉及一种代码克隆方法、装置可读存储介质以及计算机，通过获取初始代码文件，初始代码文件包括对象属性信息；并根据对象属性信息，在初始代码文件上添加注解信息，得到带有注解的代码文件；接收克隆代码指令，指令中包含待克隆代码的目标对象名称；根据待克隆代码的目标对象名称，执行代码克隆操作，得到目标代码；并将目标代码存储至目标位置。通过在编写代码时定义拷贝的规则，来减少传统开发使用get，set方法设置的繁琐，而且在减少繁琐性的同时也不影响对象克隆的性能，大大减轻开发的负担，提高了代码克隆的效率。



CN 112835563 A

1.一种代码克隆方法,其特征在于,包括:

获取初始代码文件,所述初始代码文件包括对象属性信息;

根据所述对象属性信息,在所述初始代码文件上添加注解信息,得到带有注解的代码文件;

接收克隆代码指令,所述指令中包含待克隆代码的目标对象名称;

根据所述待克隆代码的目标对象名称,执行代码克隆操作,得到目标代码;

将所述目标代码存储至目标位置。

2.根据权利要求1所述的代码克隆方法,其特征在于,所述获取初始代码文件,包括:获取一个或多个初始代码文件;或者根据待克隆代码的目标对象名称选取初始代码文件。

3.根据权利要求1所述的代码克隆方法,其特征在于,所述获取初始代码文件,所述初始代码文件包括对象属性信息,还包括:

对所述初始代码文件进行代码分析,得到所述代码文件的初始对象属性信息;

根据所述初始对象属性信息确定初始对象名称,并获取待克隆代码的目标对象名称。

4.根据权利要求1所述的代码克隆方法,其特征在于,根据所述待克隆代码的目标对象名称,执行代码克隆操作,包括:

根据所述待克隆代码的目标对象名称,在所述带有注解代码文件中检测所述待克隆代码的目标对象名称;

若检测的初始对象名称与待克隆代码的目标对象名称相符,则执行克隆操作。

5.根据权利要求1所述的代码克隆方法,其特征在于,所述对象属性信息与注解信息一一对应。

6.根据权利要求1所述的代码克隆方法,其特征在于,还包括:

预先编写脚本,所述脚本包括get,set拷贝对象属性信息的语句;

将所述脚本存储在预先设置的脚本存储目标位置。

7.一种代码克隆装置,其特征在于,包括:

获取模块,用于获取初始代码文件,所述初始代码文件包括对象属性信息;

添加注解模块,用于根据所述对象属性信息,在所述初始代码文件上添加注解信息,得到带有注解的代码文件;

接收模块,用于接收克隆代码指令,所述指令中包含待克隆代码的目标对象名称;

克隆模块,用于根据所述待克隆代码的目标对象名称,执行代码克隆操作,得到目标代码;

存储模块,用于将所述目标代码存储至目标位置。

8.根据权利要求7所述的克隆装置,其特征在于,还包括:

注解处理器,用于根据待克隆代码的目标对象名称,在带有注解代码文件中检测待克隆代码的目标对象名称。

9.一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如权利要求1至权利要求6任一项权利要求所述的方法。

10.一种计算机,包括存储器、处理器及存储在存储器上并可在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程序时实现如权利要求1至权利要求6任一项权利要求所述的方法。

一种代码克隆方法、装置、可读存储介质以及计算机

技术领域

[0001] 本申请属于计算机技术领域，具体涉及一种代码克隆方法、装置可读存储介质以及计算机。

背景技术

[0002] Java是一门面向对象编程语言，不仅吸收了C++语言的各种优点，还摒弃了C++里难以理解的多继承、指针等概念，因此Java语言具有功能强大和简单易用两个特征。Java语言作为静态面向对象编程语言的代表，极好地实现了面向对象理论，允许程序员以优雅的思维方式进行复杂的编程。在Java代码开发过程中通常存在大量相同的对像，因此，在制作该相同的对象时，通常是制作一个初始对象，然后基于该初始对象进行复制，进而得到大量相同的对像。

[0003] 现有技术中，每次克隆对象之间的属性的时候，都是调用get,set方法，这种方法在属性比较少的时候，是没有什么问题的，但是属性如果特别多，假设100个或者更多，那就要写很多get,set代码，现有技术中的这种方式存在工作量大，容易出错的问题。

[0004] 还有一种拷贝对象的方式就是通过反射的方式拷贝对象之间的属性，这种方式虽然简洁，不用写更多的get,set方法，但是性能低效，一般不建议使用。

[0005] 因此，目前需要本领域技术人员迫切解决的一个技术问题就是：如何实现更多个性化的属性拷贝，并且这种拷贝不影响性能，以减轻开发的负担，提高代码克隆的效率。

发明内容

[0006] 为至少在一定程度上克服相关技术中，解决现有技术中存在的问题，本申请提供以下技术方案，

[0007] 第一方面，一种代码克隆方法，包括：

[0008] 获取初始代码文件，所述初始代码文件包括对象属性信息；

[0009] 根据所述对象属性信息，在所述初始代码文件上添加注解信息，得到带有注解的代码文件；

[0010] 接收克隆代码指令，所述指令中包含待克隆代码的目标对象名称；

[0011] 根据所述待克隆代码的目标对象名称，执行代码克隆操作，得到目标代码；

[0012] 将所述目标代码存储至目标位置。

[0013] 进一步地，所述获取初始代码文件，包括：获取一个或多个初始代码文件；或者根据待克隆代码的目标对象名称选取初始代码文件。

[0014] 进一步地，所述获取初始代码文件，所述初始代码文件包括对象属性信息，还包括：

[0015] 对所述初始代码文件进行代码分析，得到所述代码文件的初始对象属性信息；

[0016] 根据所述初始对象属性信息确定初始对象名称，并获取待克隆代码的目标对象名称。

[0017] 进一步地，根据所述待克隆代码的目标对象名称，执行代码克隆操作，包括：

[0018] 根据所述待克隆代码的目标对象名称，在所述带有注解代码文件中检测所述待克隆代码的目标对象名称；

[0019] 若检测的初始对象名称与待克隆代码的目标对象名称相符，则执行克隆操作。

[0020] 进一步地，所述对象属性信息与注解信息一一对应。

[0021] 进一步地，还包括：

[0022] 预先编写脚本，所述脚本包括get，set拷贝对象属性信息的语句；

[0023] 将所述脚本存储在预先设置的脚本存储目标位置。

[0024] 第二方面，一种代码克隆装置，包括：

[0025] 获取模块，用于获取初始代码文件，所述初始代码文件包括对象属性信息；

[0026] 添加注解模块，用于根据所述对象属性信息，在所述初始代码文件上添加注解信息，得到带有注解的代码文件；

[0027] 接收模块，用于接收克隆代码指令，所述指令中包含待克隆代码的目标对象名称；

[0028] 克隆模块，用于根据所述待克隆代码的目标对象名称，执行代码克隆操作，得到目标代码；

[0029] 存储模块，用于将所述目标代码存储至目标位置。

[0030] 进一步地，还包括：

[0031] 注解处理器，用于根据待克隆代码的目标对象名称，在带有注解代码文件中检测待克隆代码的目标对象名称。

[0032] 第三方面，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如上任一项权利要求所述的方法。

[0033] 第四方面，一种计算机，包括存储器、处理器及存储在存储器上并可在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程序时实现如上任一项权利要求所述的方法。

[0034] 本发明的有益效果是：

[0035] 本发明所提供的一种代码克隆方法、装置可读存储介质以及计算机，通过获取初始代码文件，初始代码文件包括对象属性信息；并根据对象属性信息，在初始代码文件上添加注解信息，得到带有注解的代码文件；接收克隆代码指令，指令中包含待克隆代码的目标对象名称；根据待克隆代码的目标对象名称，执行代码克隆操作，得到目标代码；并将目标代码存储至目标位置。基于这种模式，通过在编写代码时定义拷贝的规则，来减少传统开发使用get，set方法设置的繁琐，而且在减少繁琐性的同时也不影响对象克隆的性能，大大减轻开发的负担，提高了代码克隆的效率。

[0036] 应当理解的是，以上的一般描述和后文的细节描述仅是示例性和解释性的，并不能限制本申请。

附图说明

[0037] 此处的附图被并入说明书中并构成本说明书的一部分，示出了符合本申请的实施例，并与说明书一起用于解释本申请的原理。

[0038] 图1为本申请一个实施例中一种代码克隆方法步骤图。

[0039] 图2为本申请另一个实施例中一种代码克隆方法步骤图

[0040] 图3为本申请另一个实施例中一种代码克隆方法步骤图。

[0041] 图4为本申请另一个实施例中一种代码克隆装置结构框图。

[0042] 图中：

[0043] 1-获取模块;2-添加注解模块;3-接收模块;4-克隆模块;5-存储模块。

具体实施方式

[0044] 为使本申请的目的、技术方案和优点更加清楚，下面将对本申请的技术方案进行详细的描述。显然，所描述的实施例仅仅是本申请一部分实施例，而不是全部的实施例。基于本申请中的实施例，本领域普通技术人员在没有做出创造性劳动的前提下所得到的所有其它实施方式，都属于本申请所保护的范围。

[0045] 为实现以上目的，本发明采用如下技术方案：

[0046] 如图1所示，为本申请一个实施例中一种代码克隆方法步骤图，

[0047] 包括：

[0048] 步骤S101、获取初始代码文件，所述初始代码文件包括对象属性信息；

[0049] 在本公开的一些示例性实施方式中，初始代码文件件可以是现有的代码功能模块中包含的各类资源文件，初始代码可以是功能模块中包含的代码段。

[0050] 步骤S102、根据对象属性信息，在初始代码文件上添加注解信息，得到带有注解的代码文件；

[0051] 从JDK5开始，Java增加对元数据的支持，也就是注解，注解与注释是有一定区别的，可以把注解理解为代码里的特殊标记，这些标记可以在编译，类加载，运行时被读取，并执行相应的处理。通过注解开发人员可以在不改变原有代码和逻辑的情况下在源代码中嵌入补充信息。注解，可以看作是对一个类/方法的一个扩展的模版，每个类/方法按照注解类中的规则，来为类/方法注解不同的参数，在用到的地方可以得到不同的类/方法中注解的各种参数与值。

[0052] 步骤S103、接收克隆代码指令，指令中包含待克隆代码的目标对象名称；

[0053] 步骤S104、根据待克隆代码的目标对象名称，执行代码克隆操作，得到目标代码；

[0054] 步骤S105、将目标代码存储至目标位置。

[0055] 需要说明的是，在现有技术中的java编程中，拷贝对象属性有主要有两种方式，第一种是通过get,set方法进行拷贝，这种方式的大致过程如下:

[0056] //模拟待拷贝的对象

[0057] PersonVO pv＝new PersonVO();

[0058] pv.setName("test");

[0059] //拷贝的目标对象

[0060] PersonBO pb＝new PersonBO();

[0061] //克隆对象的属性

[0062] pb.setName(pv.getName());

[0063] 每次克隆对象之间的属性的时候，都是调用get,set方法，这种方式在属性比较少的时候，是没有什么问题的，但是属性如果特别多，假设100个或者更多，那就要写很多get,

set代码,非常繁琐且容易出错。

[0064] 还有一种拷贝对象的方式就是通过反射的方式拷贝对象之间的属性,这种方式虽然简洁,不用写更多的get,set方法,但是性能低效,一般不建议使用。

[0065] 可以理解为,本发明所提供的一种代码克隆方法,通过获取初始代码文件,初始代码文件包括对象属性信息;并根据对象属性信息,在初始代码文件上添加注解信息,得到带有注解的代码文件;接收克隆代码指令,指令中包含待克隆代码的目标对象名称;根据待克隆代码的目标对象名称,执行代码克隆操作,得到目标代码;并将目标代码存储至目标位置。基于这种模式,通过在编写代码时定义拷贝的规则,来减少传统开发使用get,set方法设置的繁琐,而且在减少繁琐性的同时也不影响对象克隆的性能,大大减轻开发的负担,提高了代码克隆的效率。

[0066] 在一个实施例中,获取初始代码文件,包括:获取一个或多个初始代码文件;或者根据待克隆代码的目标对象名称选取初始代码文件。

[0067] 作为上述实施例的进一步改进,如图2所示,为本申请另一个实施例中一种代码克隆方法步骤图,

[0068] 步骤201、获取初始代码文件,初始代码文件包括对象属性信息;

[0069] 步骤202、对初始代码文件进行代码分析,得到代码文件的初始对象属性信息;

[0070] 步骤203、根据初始对象属性信息确定初始对象名称,并获取待克隆代码的目标对象名称。

[0071] 步骤S204、根据对象属性信息,在初始代码文件上添加注解信息,得到带有注解的代码文件;

[0072] 步骤S205、接收克隆代码指令,指令中包含待克隆代码的目标对象名称;

[0073] 步骤S206、根据待克隆代码的目标对象名称,执行代码克隆操作,得到目标代码;

[0074] 步骤S207、将目标代码存储至目标位置。

[0075] 可以理解为,本发明所提供的一种代码克隆方法,通过获取初始代码文件,初始代码文件包括对象属性信息;对初始代码文件进行代码分析,得到代码文件的初始对象属性信息;根据初始对象属性信息确定初始对象名称,并获取待克隆代码的目标对象名称。并根据对象属性信息,在初始代码文件上添加注解信息,得到带有注解的代码文件;接收克隆代码指令,指令中包含待克隆代码的目标对象名称;根据待克隆代码的目标对象名称,执行代码克隆操作,得到目标代码;并将目标代码存储至目标位置。基于这种模式,通过在编写代码时定义拷贝的规则,来减少传统开发使用get,set方法设置的繁琐,而且在减少繁琐性的同时也不影响对象克隆的性能,大大减轻开发的负担,提高了代码克隆的效率。

[0076] 作为上述实施例的进一步改进,如图3所示,为本申请另一个实施例中一种代码克隆方法步骤图,

[0077] 步骤301、获取初始代码文件,初始代码文件包括对象属性信息;

[0078] 步骤302、对初始代码文件进行代码分析,得到代码文件的初始对象属性信息;

[0079] 步骤303、根据初始对象属性信息确定初始对象名称,并获取待克隆代码的目标对象名称。

[0080] 步骤S304、根据对象属性信息,在初始代码文件上添加注解信息,得到带有注解的代码文件;

[0081] 步骤S305、接收克隆代码指令，指令中包含待克隆代码的目标对象名称；

[0082] 步骤S306、根据待克隆代码的目标对象名称，在带有注解代码文件中检测待克隆代码的目标对象名称；

[0083] 步骤S307、若检测的初始对象名称与待克隆代码的目标对象名称相符，则执行克隆操作。

[0084] 步骤S308、将目标代码存储至目标位置。

[0085] 可以理解为，本发明所提供的一种代码克隆方法，通过获取初始代码文件，初始代码文件包括对象属性信息；对初始代码文件进行代码分析，得到代码文件的初始对象属性信息；根据初始对象属性信息确定初始对象名称，并获取待克隆代码的目标对象名称。并根据对象属性信息，在初始代码文件上添加注解信息，得到带有注解的代码文件；接收克隆代码指令，指令中包含待克隆代码的目标对象名称；根据待克隆代码的目标对象名称，在带有注解代码文件中检测待克隆代码的目标对象名称；若检测的初始对象名称与待克隆代码的目标对象名称相符，则执行克隆操作，得到目标代码；并将目标代码存储至目标位置。基于这种模式，通过在编写代码时定义拷贝的规则，来减少传统开发使用get，set方法设置的繁琐，而且在减少繁琐性的同时也不影响对象克隆的性能，大大减轻开发的负担，提高了代码克隆的效率。

[0086] 一些实施例中，对象属性信息与注解信息一一对应。

[0087] 一些实施例中，还包括，预先编写脚本，脚本包括get，set拷贝对象属性信息的语句；

[0088] 将脚本存储在预先设置的脚本存储目标位置。

[0089] 作为上述实施例的进一步改进，如图4所示，为本申请一个实施例中一种代码克隆装置结构框图，

[0090] 包括：

[0091] 获取模块1，用于获取初始代码文件，初始代码文件包括对象属性信息；

[0092] 添加注解模块2，用于根据对象属性信息，在初始代码文件上添加注解信息，得到带有注解的代码文件；

[0093] 接收模块3，用于接收克隆代码指令，指令中包含待克隆代码的目标对象名称；

[0094] 克隆模块4，用于根据待克隆代码的目标对象名称，执行代码克隆操作，得到目标代码；

[0095] 存储模块5，用于将目标代码存储至目标位置。

[0096] 可以理解为，本发明所提供的一种代码克隆装置，通过获取模块获取初始代码文件，初始代码文件包括对象属性信息；通过添加注解模块，根据对象属性信息，在初始代码文件上添加注解信息，得到带有注解的代码文件；通过接收模块，接收克隆代码指令，指令中包含待克隆代码的目标对象名称；通过克隆模块根据待克隆代码的目标对象名称，执行代码克隆操作，得到目标代码；通过存储模块将目标代码存储至目标位置。基于这种模式，通过在编写代码时定义拷贝的规则，来减少传统开发使用get，set方法设置的繁琐，而且在减少繁琐性的同时也不影响对象克隆的性能，大大减轻开发的负担，提高了代码克隆的效率。

[0097] 作为上述装置的进一步改进，在一个实施例中，还包括：

[0098] 注解处理器,用于根据待克隆代码的目标对象名称,在带有注解代码文件中检测待克隆代码的目标对象名称。

[0099] 其中,注解处理器(Annotation Processor)是java的一个工具,不管是运行时注解还是编译时注解,都会通过处理器在编译时进行扫描和处理注解。

[0100] Java中有默认的注解处理器,使用者也可以自定义注解处理器,注册后使用注解处理器处理注解,最终达到注解本身起到的效果。注解处理器将标记了注解的类,变量等作为输入内容,经过注解处理器处理,生成想要生成的java代码。所以处理器可以理解为就是一个生成代码的工具,只是是通过注解的规则生成。生成后的代码,可以看作是同一般代码,最终被编译。

[0101] 一些实施例中,本申请还提供了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如上的方法。

[0102] 一些实施例中,本申请还提供了一种计算机,包括存储器、处理器及存储在存储器上并可在处理器上运行的计算机程序,其特征在于,处理器执行程序时实现如上的方法。

[0103] 为便于理解,本发明还提供了一个利用本方法实现代码克隆的例子,如下所示:

[0104] 例如:现在有两个类A和B他们都有age,name两个属性

```
A{
  int age
  string name
}
```

[0105]

```
B {
  int age
  string name
}
```

[0106] 假设现在A有个对象是这样的,年龄是15岁,名字叫张三;伪代码类似下面这样:

[0107] A a1=new A()

[0108] a1.age=15

[0109] a1.name='张三'

[0110] 现在要把A对应的a1对象克隆成B的b1对象,也就是要通过a1自动转化生成B类对象的属性;

[0111] 原来的方法可能是这样:

[0112] B b1=new B()

[0113] b1.age=a1.age

[0114] b1.name=a1.name

[0115] 通过本申请的所提供的代码克隆方法,现在可以通过配置注解加上调用转换操作的方法

[0116] @Mapper

[0117] interface BMapper{

[0118] BMapper getMapper(BMapper.class)

[0119] B toB(A a1)

[0120] }

[0121] B b1＝BMapper.getMapper().toB(a1)

[0122] 就可以自动完成前面的转换，在BMapper加上个Mapper注解之后，会自动在编译期间生成如下的类，伪代码如下：

```
BMapperImpl {
  B toB(A a1 ){
     B b1 = new B()
[0123]  b1.age = a1.age
     b1.name = a1.name
  }
}
```

[0124] 由此可见，基于这种模式，就可以通过注解加自动生成拷贝对象实现Mapper的方式实现对象属性之间的拷贝，更进一步，也可以实现更多个性化的属性拷贝，并且这种拷贝不影响性能。

[0125] 可以理解的是，上述各实施例中相同或相似部分可以相互参考，在一些实施例中未详细说明的内容可以参见其他实施例中相同或相似的内容。

[0126] 需要说明的是，在本申请的描述中，术语“第一”、“第二”等仅用于描述目的，而不能理解为指示或暗示相对重要性。此外，在本申请的描述中，除非另有说明，“多个”的含义是指至少两个。

[0127] 流程图中或在此以其他方式描述的任何过程或方法描述可以被理解为，表示包括一个或更多个用于实现特定逻辑功能或过程的步骤的可执行指令的代码的模块、片段或部分，并且本申请的优选实施方式的范围包括另外的实现，其中可以不按所示出或讨论的顺序，包括根据所涉及的功能按基本同时的方式或按相反的顺序，来执行功能，这应被本申请的实施例所属技术领域的技术人员所理解。

[0128] 应当理解，本申请的各部分可以用硬件、软件、固件或它们的组合来实现。在上述实施方式中，多个步骤或方法可以用存储在存储器中且由合适的指令执行系统执行的软件或固件来实现。例如，如果用硬件来实现，和在另一实施方式中一样，可用本领域公知的下列技术中的任一项或他们的组合来实现：具有用于对数据信号实现逻辑功能的逻辑门电路的离散逻辑电路，具有合适的组合逻辑门电路的专用集成电路，可编程门阵列(PGA)，现场可编程门阵列(FPGA)等。

[0129] 本技术领域的普通技术人员可以理解实现上述实施例方法携带的全部或部分步骤是可以通过程序来指令相关的硬件完成，所述的程序可以存储于一种计算机可读存储介

质中，该程序在执行时，包括方法实施例的步骤之一或其组合。

[0130] 此外，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模块中，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模块中。上述集成的模块既可以采用硬件的形式实现，也可以采用软件功能组件的形式实现。所述集成的模块如果以软件功能组件的形式实现并作为独立的产品销售或使用时，也可以存储在一个计算机可读取存储介质中。

[0131] 上述提到的存储介质可以是只读存储器，磁盘或光盘等。

[0132] 在本说明书的描述中，参考术语“一个实施例”、“一些实施例”、“示例”、“具体示例”、或“一些示例”等的描述意指结合该实施例或示例描述的具体特征、结构、材料或者特点包含于本申请的至少一个实施例或示例中。在本说明书中，对上述术语的示意性表述不一定指的是相同的实施例或示例。而且，描述的具体特征、结构、材料或者特点可以在任何的一个或多个实施例或示例中以合适的方式结合。

[0133] 尽管上面已经示出和描述了本申请的实施例，可以理解的是，上述实施例是示例性的，不能理解为对本申请的限制，本领域的普通技术人员在本申请的范围内可以对上述实施例进行变化、修改、替换和变型。

[0134] 需要说明的是，本发明不局限于上述最佳实施方式，本领域技术人员在本发明的启示下都可得出其他各种形式的产品，但不论在其形状或结构上作任何变化，凡是具有与本申请相同或相近似的技术方案，均落在本发明的保护范围之内。

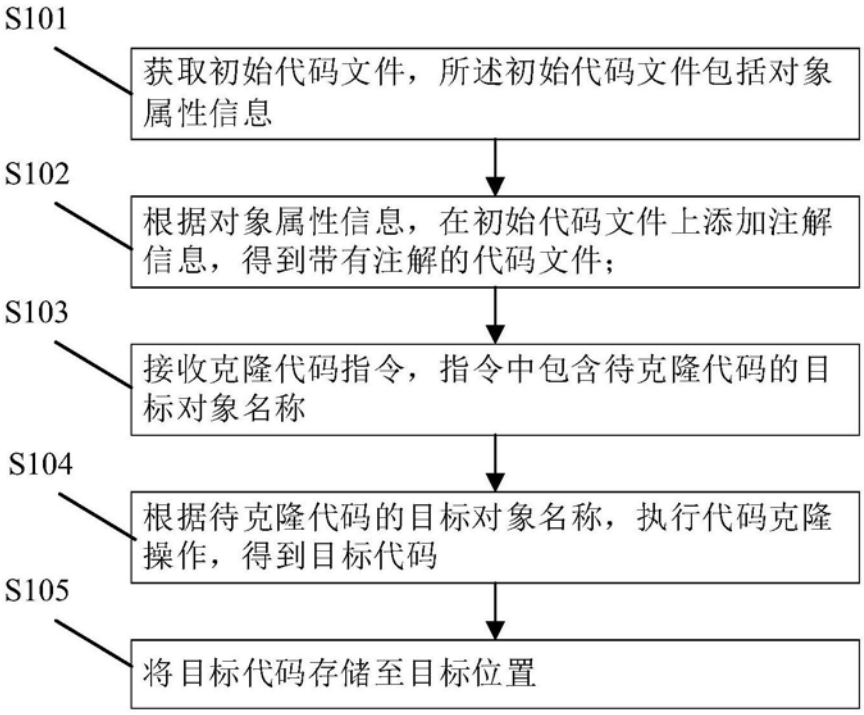

S101
获取初始代码文件，所述初始代码文件包括对象属性信息
S102
根据对象属性信息，在初始代码文件上添加注解信息，得到带有注解的代码文件；
S103
接收克隆代码指令，指令中包含待克隆代码的目标对象名称
S104
根据待克隆代码的目标对象名称，执行代码克隆操作，得到目标代码
S105
将目标代码存储至目标位置

图1

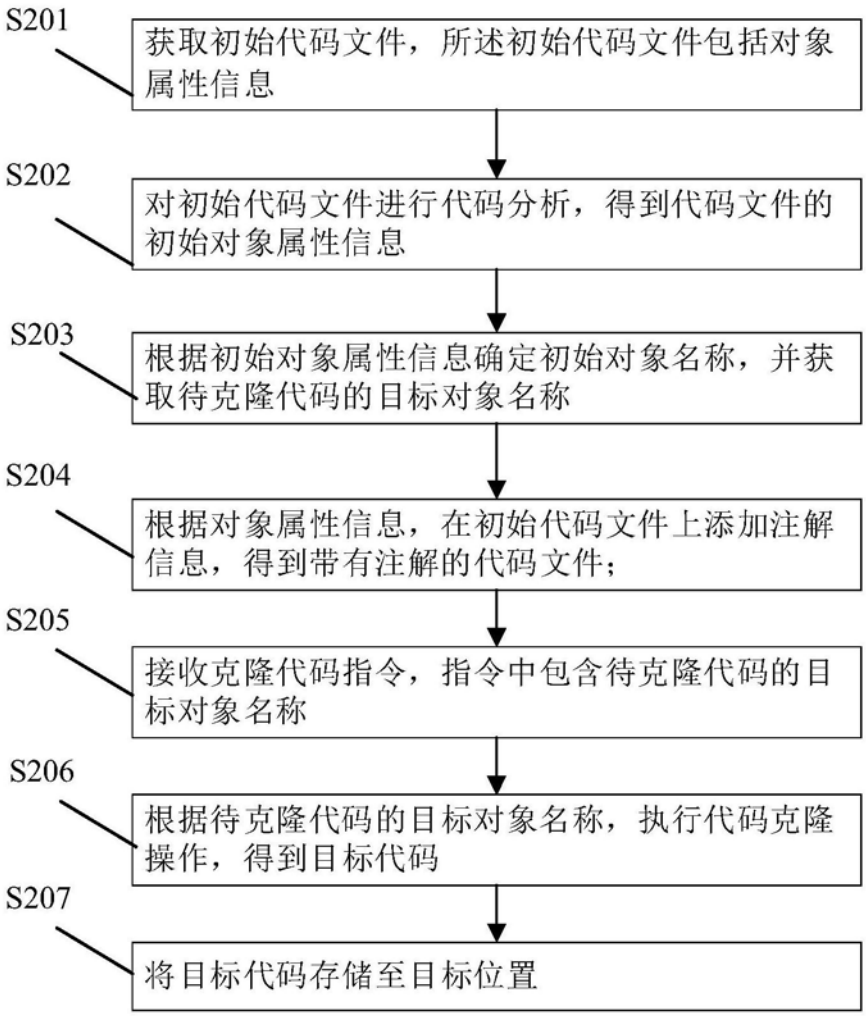

S201
获取初始代码文件，所述初始代码文件包括对象属性信息
S202
对初始代码文件进行代码分析，得到代码文件的初始对象属性信息
S203
根据初始对象属性信息确定初始对象名称，并获取待克隆代码的目标对象名称
S204
根据对象属性信息，在初始代码文件上添加注解信息，得到带有注解的代码文件；
S205
接收克隆代码指令，指令中包含待克隆代码的目标对象名称
S206
根据待克隆代码的目标对象名称，执行代码克隆操作，得到目标代码
S207
将目标代码存储至目标位置

图2

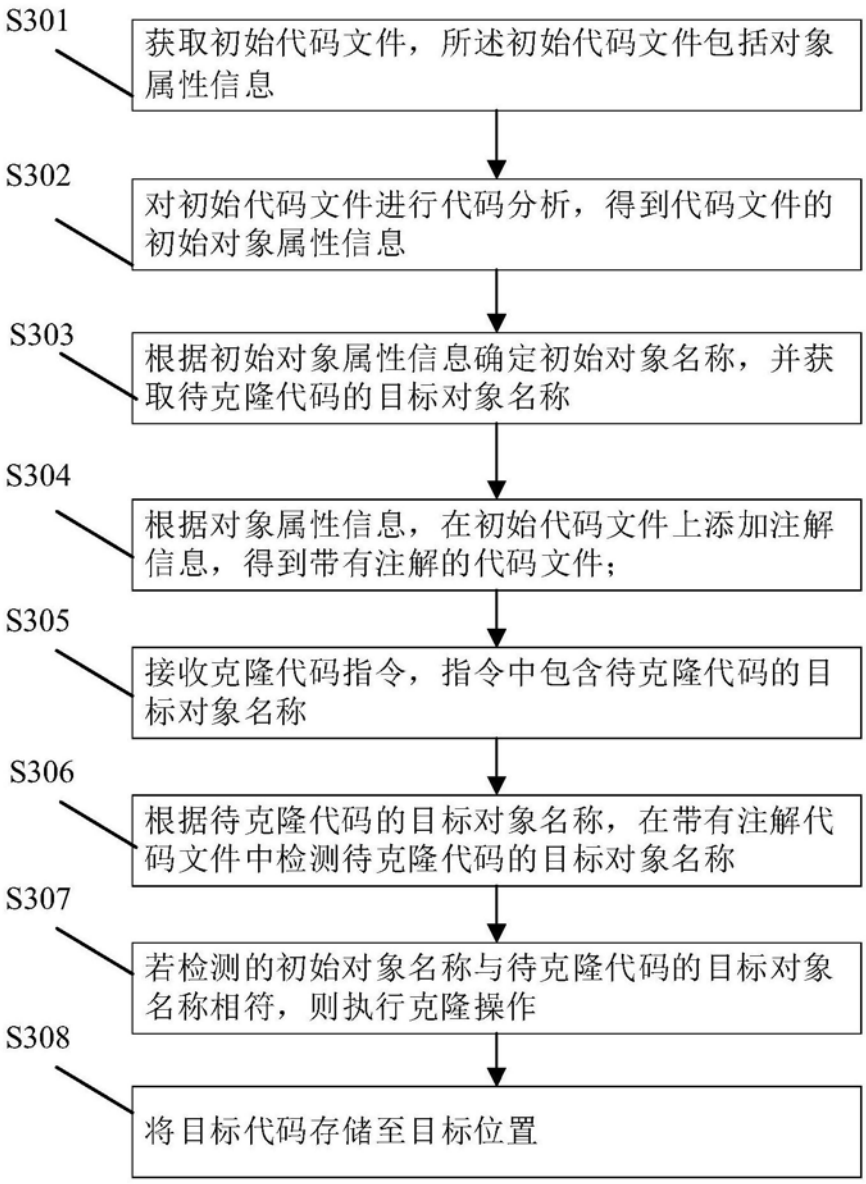

S301
获取初始代码文件，所述初始代码文件包括对象属性信息
S302
对初始代码文件进行代码分析，得到代码文件的初始对象属性信息
S303
根据初始对象属性信息确定初始对象名称，并获取待克隆代码的目标对象名称
S304
根据对象属性信息，在初始代码文件上添加注解信息，得到带有注解的代码文件；
S305
接收克隆代码指令，指令中包含待克隆代码的目标对象名称
S306
根据待克隆代码的目标对象名称，在带有注解代码文件中检测待克隆代码的目标对象名称
S307
若检测的初始对象名称与待克隆代码的目标对象名称相符，则执行克隆操作
S308
将目标代码存储至目标位置

图3

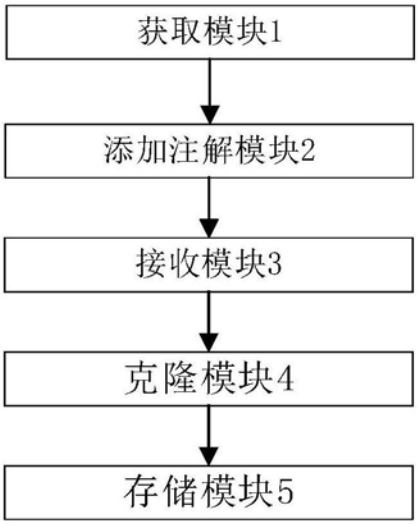


图4