

SDK 的修复方法、装置、终端、设备、系统及介质

相关申请的交叉引用

本申请要求享有于 2022 年 08 月 26 日提交的名称为“SDK 的修复方法、装置、终端、设备、系统及介质”的中国专利申请 202211035360.6 的优先权，该申请的全部内容通过引用并入本文中。

技术领域

本申请属于数据处理领域，尤其涉及一种 SDK 的修复方法、装置、终端、设备、系统及介质。

背景技术

软件开发工具包（Software Development Kit，SDK）可为为软件框架、硬件平台、操作系统或应用程序提供某项功能的工具包。SDK 可嵌入至不同渠道方的软件框架、硬件平台、操作系统或应用程序等宿主中。SDK 在与宿主共同运行的情况下，可能会产生异常，需要对 SDK 进行修复，使得 SDK 能够正常运行。

但由于渠道方不同，各渠道方的开发方式、宿主环境也具有一定的差异，SDK 出现的缺陷也可能不同，对 SDK 单一的修复难以满足不同渠道方的 SDK 的修复需求。

发明内容

本申请实施例提供一种 SDK 的修复方法、装置、终端、设备、系统及介质，能够满足不同渠道方的 SDK 的修复需求。

第一方面，本申请实施例提供一种 SDK 的修复方法，应用于修复平台，该方法包括：获取软件开发工具包 SDK 信息、宿主信息和不同异常类型的 SDK 异常信息；根据 SDK 信息、宿主信息和每个异常类型的 SDK

异常信息，得到目标 SDK 的每个异常类型对应的异常影响参数，目标 SDK 与宿主信息指示的宿主程序和 SDK 信息指示的 SDK 版本对应；基于目标 SDK 的每个异常类型对应的异常影响参数和预设的异常分级处理规则，得到目标 SDK 修复策略，目标 SDK 修复策略包括异常分级处理规则中与异常影响参数对应的异常级别的 SDK 修复策略；向用户终端下发目标 SDK 修复策略，以使用户终端执行目标 SDK 修复策略以修复目标 SDK。

第二方面，本申请实施例提供一种 SDK 的修复方法，应用于用户终端，该方法包括：向修复平台提供软件开发工具包 SDK 信息、宿主信息和不同异常类型的 SDK 异常信息，以使修复平台基于目标 SDK 的每个异常类型对应的异常影响参数和预设的异常分级处理规则得到目标 SDK 修复策略，目标 SDK 修复策略包括异常分级处理规则中与异常影响参数对应的异常级别的 SDK 修复策略，目标 SDK 与宿主信息指示的宿主程序和 SDK 信息指示的 SDK 版本对应，目标 SDK 的每个异常类型对应的异常影响参数根据 SDK 信息、宿主信息和每个异常类型的 SDK 异常信息得到；从修复平台获取目标 SDK 修复策略，并执行目标 SDK 修复策略修复目标 SDK。

第三方面，本申请实施例提供一种 SDK 的修复装置，应用于修复平台，该装置包括：获取模块，用于获取软件开发工具包 SDK 信息、宿主信息和不同异常类型的 SDK 异常信息；计算模块，用于根据 SDK 信息、宿主信息和每个异常类型的 SDK 异常信息，得到目标 SDK 的每个异常类型对应的异常影响参数，目标 SDK 与宿主信息指示的宿主程序和 SDK 信息指示的 SDK 版本对应；策略配置模块，用于基于目标 SDK 的每个异常类型对应的异常影响参数和预设的异常分级处理规则，得到目标 SDK 修复策略，目标 SDK 修复策略包括异常分级处理规则中与异常影响参数对应的异常级别的 SDK 修复策略；下发模块，用于向用户终端下发目标 SDK 修复策略，以使用户终端执行目标 SDK 修复策略以修复目标 SDK。

第四方面，本申请实施例提供一种用户终端，包括：发送模块，用于向修复平台提供软件开发工具包 SDK 信息、宿主信息和不同异常类型的

SDK 异常信息，以使修复平台基于目标 SDK 的每个异常类型对应的异常影响参数和预设的异常分级处理规则得到目标 SDK 修复策略，目标 SDK 修复策略包括异常分级处理规则中与异常影响参数对应的异常级别的 SDK 修复策略，目标 SDK 与宿主信息指示的宿主程序和 SDK 信息指示的 SDK 版本对应，目标 SDK 的每个异常类型对应的异常影响参数根据 SDK 信息、宿主信息和每个异常类型的 SDK 异常信息得到；接收模块，用于从修复平台获取目标 SDK 修复策略；执行模块，用于执行目标 SDK 修复策略修复目标 SDK。

第五方面，本申请实施例提供一种电子设备，应用于修复平台，电子设备包括：处理器以及存储有计算机程序指令的存储器；处理器执行计算机程序指令时实现第一方面的 SDK 的修复方法。

第六方面，本申请实施例提供一种用户终端，包括：处理器以及存储有计算机程序指令的存储器；处理器执行计算机程序指令时实现第二方面的 SDK 的修复方法。

第七方面，本申请实施例提供一种 SDK 的修复系统，包括：修复平台，用于执行第一方面的 SDK 的修复方法；用户终端，用于执行第二方面的 SDK 的修复方法。

第八方面，本申请实施例提供一种计算机可读存储介质，计算机可读存储介质上存储有计算机程序指令，计算机程序指令被处理器执行时实现第一方面的 SDK 的修复方法或第二方面的 SDK 的修复方法。

本申请实施例提供一种 SDK 的修复方法、装置、终端、设备、系统及介质，修复平台能够根据 SDK 信息、宿主信息和采集到的不同异常类型的 SDK 异常信息，得到表征异常类型对 SDK 的影响的异常影响参数，从而基于每个异常类型对应的异常影响参数和预设的异常分级处理规则，得到对应异常级别的 SDK 修复策略并下发给用户终端。用户终端可执行对应异常级别的 SDK 修复策略，对 SDK 进行修复。不同的宿主信息对应不同的宿主程序，不同的宿主程序对应不同的渠道方，不同版本的 SDK 也对应有不同的 SDK 信息，嵌入在不同渠道方的宿主程序中的不同版本的 SDK 产生的 SDK 异常信息不同，通过采集到的 SDK 异常信息，可得到与

嵌入每个渠道方的宿主程序的每个版本的 SDK 对应的异常级别的 SDK 修复策略，即，可为嵌入不同渠道方的宿主程序的不同版本的 SDK 自动分配适配的 SDK 修复策略，从而能够满足不同渠道方的 SDK 的修复需求。

附图说明

为了更清楚地说明本申请实施例的技术方案，下面将对本申请实施例中所需要使用的附图作简单的介绍，对于本领域普通技术人员来讲，在不付出创造性劳动的前提下，还可以根据这些附图获得其他的附图。

图 1 为本申请实施例提供的 SDK 的修复方法的应用场景的一示例的示意图；

图 2 为本申请实施例提供的 SDK 的修复方法的应用场景的另一示例的示意图；

图 3 为本申请第一方面一实施例提供的 SDK 的修复方法的流程图；

图 4 为本申请第一方面另一实施例提供的 SDK 的修复方法的流程图；

图 5 为本申请第一方面又一实施例提供的 SDK 的修复方法的流程图；

图 6 为本申请第二方面一实施例提供的 SDK 的修复方法的流程图；

图 7 为本申请第二方面另一实施例提供的 SDK 的修复方法的流程图；

图 8 为本申请第二方面又一实施例提供的 SDK 的修复方法的流程图；

图 9 为本申请实施例提供的 SDK 的修复方法的一示例的逻辑示意图；

图 10 为本申请第三方面一实施例提供的 SDK 的修复装置的结构示意图；

图 11 为本申请第四方面一实施例提供的用户终端的结构示意图；

图 12 为本申请第五方面一实施例提供的电子设备的结构示意图。

具体实施方式

下面将详细描述本申请的各个方面的特征和示例性实施例，为了使本申请的目的、技术方案及优点更加清楚明白，以下结合附图及具体实施例，对本申请进行进一步详细描述。应理解，此处所描述的具体实施例仅意在解释本申请，而不是限定本申请。对于本领域技术人员来说，本申请可以在不需要这些具体细节中的一些细节的情况下实施。下面对实施例的描述仅仅是为了通过示出本申请的示例来提供对本申请更好的理解。

软件开发工具包（Software Development Kit，SDK）可为为软件框架、硬件平台、操作系统或应用程序提供某项功能的工具包。SDK 可嵌入至不同渠道方的软件框架、硬件平台、操作系统或应用程序等宿主中。SDK 在与宿主共同运行的情况下，可能会产生异常，需要对 SDK 进行修复，使得 SDK 能够正常运行。但由于渠道方不同，各渠道方的开发方式、宿主环境也具有一定的差异，SDK 出现的缺陷也可能不同，对 SDK 单一的修复难以满足不同渠道方的 SDK 的修复需求。

本申请提供一种 SDK 的修复方法、装置、终端、设备、系统及介质，修复平台能够根据 SDK 信息、宿主信息和采集到的不同异常类型的 SDK 异常信息，得到能够表征异常类型对 SDK 的影响的异常影响参数，从而基于每个异常类型对应的异常影响参数和预设的异常分级处理规则，得到对应异常级别的 SDK 修复策略，并下发给用户终端。用户终端可执行应异常级别的 SDK 修复策略，对 SDK 进行修复。不同渠道方对应有不同的宿主信息，不同版本的 SDK 对应有不同的 SDK 信息，不同渠道方、不同版本的 SDK 对应产生的 SDK 异常信息也不同，通过不同的信息，可得到与每个渠道方、每个版本的 SDK 对应的异常级别的 SDK 修复策略，即，不同渠道方、不同版本的 SDK 可对应得到适配的 SDK 修复策略，能够满足不同渠道方的 SDK 的修复需求。

本申请实施例提供的 SDK 的修复方法可应用于对 SDK 修复的场景，可涉及修复平台 and 用户终端。图 1 为本申请实施例提供的 SDK 的修复方法的应用场景的一示例的示意图。如图 1 所示，用户终端 11 可与修复平台 12 通信交互。

用户终端 11 中具有至少一个宿主程序 111 和至少一个 SDK 112。同一用户终端中可具有多个不同的宿主程序 111，不同的宿主程序 111 对应的渠道方不同。SDK 112 可嵌入在宿主程序 111 中，宿主程序 111 可调用 SDK 112，从而使用 SDK 112 提供的功能。用户终端 11 具体可包括手机、平板电脑、电子计算机、智能穿戴设备等，在此并不限定。SDK 112 可作为信息提供者向修复平台 12 提供 SDK 112 的 SDK 信息、SDK 异常信息以及 SDK 112 所在的宿主程序 111 的宿主信息等。用户终端 11 可从修复平台 12 获取具有针对性的 SDK 修复策略，执行该 SDK 修复策略，以修复作为被修复对象的 SDK 112。

修复平台 12 可为用于为用户终端提供 SDK 修复服务的后台系统，可包括一台以上的电子设备。电子设备可包括服务器、网关、路由器等设备，在此并不限定修复平台 12 中电子设备的类型和数量。修复平台 12 可为多个用户终端 11 提供 SDK 修复服务。不同的用户终端中具有宿主程序 111 可以相同，也可以不同。修复平台 12 可维护不同的渠道方、宿主程序、用户、SDK、SDK 异常信息和 SDK 修复策略等之间的对应关系，可分类归纳得到 SDK 异常的异常类型，并执行报警、提示等动作。修复平台 12 可为不同的用户终端 11 下发与对该用户终端 11 中需要修复的 SDK 112 具有针对性的修复策略。且修复平台 12 可通过全量、灰度、指定用户的方式向用户终端下发具有针对性的修复策略，在此并不限定。

在一些实施例中，本申请实施例提供的 SDK 的修复方法还可涉及渠道平台。图 2 为本申请实施例提供的 SDK 的修复方法的应用场景的另一示例的示意图。如图 2 所示，渠道平台 13 可与修复平台 12 通信交互。

渠道平台 13 可为 SDK 修复策略的审核平台，可包括一台以上的电子设备。电子设备可包括服务器、网关、路由器等设备，在此并不限定渠道平台 13 中电子设备的类型和数量。每个渠道方可在渠道平台 13 开设一个账号，通过渠道平台 13 对针对渠道方的宿主程序 11 嵌入的 SDK 112 提供的 SDK 修复策略进行审核。为宿主程序 111 中嵌入的 SDK 112 提供的 SDK 修复策略可先发送至与宿主程序 111 对应的渠道平台 13 进行审核，在渠道平台 13 对 SDK 修复策略审核通过后，修复平台 12 才会将该 SDK

修复策略下发给用户终端 11。在一些示例中，渠道平台 13 也可向修复平台 13 提供 SDK 异常信息。

下面对本申请中的 SDK 的修复方法、装置、终端、设备、系统及介质分别进行说明。

本申请第一方面提供一种 SDK 的修复方法，可应用于修复平台，即该 SDK 的修复方法可由修复平台执行。图 3 为本申请第一方面一实施例提供的 SDK 的修复方法的流程图。如图 3 所示，该 SDK 的修复方法可包括步骤 S201 至步骤 S204。

在步骤 S201 中，获取 SDK 信息、宿主信息和不同异常类型的 SDK 异常信息。

SDK 信息包括 SDK 的基础信息，例如，SDK 信息可包括 SDK 版本等，在此并不限定。SDK 信息可指示 SDK 版本。宿主信息包括 SDK 嵌入的宿主程序的信息，例如，宿主信息可包括宿主程序标识、宿主程序所属渠道方标识、宿主程序环境信息等，在此并不限定。宿主信息可指示宿主程序，宿主程序与渠道方具有对应关系。异常类型为 SDK 的异常类型，可根据场景、需求、经验等设定异常类型。例如，异常类型可包括但不限于数据异常类型、文件输入输出异常类型、业务不可达异常类型、崩溃异常类型等。一条 SDK 异常信息对应一次 SDK 异常。一条 SDK 异常信息包括与 SDK 发生的一次异常相关的信息，例如，一条 SDK 异常信息可包括 SDK 异常描述信息、发生异常的 SDK 嵌入的宿主程序的渠道方标识、SDK 发生的异常是否有渠道方反馈、SDK 上线时间、SDK 异常影响的用户、SDK 异常影响的终端类型、发生异常的 SDK 的业务成功率、发生异常的 SDK 的业务重要性等，在此并不限定。

在一些示例中，SDK 异常信息包括第一 SDK 异常信息和/或第二 SDK 异常信息，第一 SDK 异常信息由用户终端提供，第二 SDK 由渠道平台提供。更进一步地，第一 SDK 异常信息可由用户终端中发生 SDK 异常的 SDK 提供。第一 SDK 异常信息可从用户终端中 SDK 的日志得到，第二 SDK 异常信息可从渠道平台的日志中得到。第一 SDK 异常信息也可包括埋点信息，以便于后续分析过程中可回溯 SDK 发生异常的位置、时间、

原因等。第二 SDK 异常信息可为渠道平台提供的技术人员录入的异常信息、运行时间（即 runtime）信息等。

在步骤 S202 中，根据 SDK 信息、宿主信息和每个异常类型的 SDK 异常信息，得到目标 SDK 的每个异常类型对应的异常影响参数。

目标 SDK 与宿主信息指示的宿主程序和 SDK 信息指示的 SDK 版本对应。可根据宿主信息和 SDK 信息，确定目标 SDK。目标 SDK 不同，目标 SDK 对应的 SDK 异常信息也有可能不同，不同的 SDK 异常信息表征的 SDK 异常对 SDK 的影响程度也不同。异常影响参数可表征 SDK 异常对 SDK 的影响程度。目标 SDK 的每个异常类型对应的异常影响参数保证该异常类型的 SDK 异常对目标 SDK 的影响程度。不同目标 SDK 的不同异常类型对应的异常影响参数可能不同，如，不同宿主程序中不同版本的 SDK 的不同异常类型对应的异常影响参数可能不同。不同目标 SDK 的同一异常类型对应的异常影响参数可能不同，如，不同宿主程序中不同版本的 SDK 的同一异常类型对应的异常影响参数可能不同，不同宿主程序中同一版本的 SDK 的同一异常类型对应的异常影响参数可能不同，相同宿主程序中不同版本 SDK 的同一异常类型对应的异常影响参数可能不同。

异常影响参数可为利用标准化算法计算得到的能够表征 SDK 异常对 SDK 的影响程度的参数，即异常影响参数标准化且具有可比性。如，不同目标 SDK 的不同异常类型对应的异常影响参数具有可比性。在一些示例中，可预先为至少部分 SDK 异常信息设置对应的计算值和权重，以建立统一的标准，从而使得利用权重算法计算得到不同目标 SDK 下不同异常类型对应的异常影响参数标准化且具有可比性。

在步骤 S203 中，基于目标 SDK 的每个异常类型对应的异常影响参数和预设的异常分级处理规则，得到目标 SDK 修复策略。

异常分级处理规则可包括异常类型、异常影响参数、异常级别和 SDK 修复策略的对应关系。目标 SDK 修复策略包括异常分级处理规则中与异常影响参数对应的异常级别的 SDK 修复策略。即，根据目标 SDK 的每个异常类型对应的异常影响参数，可在异常分级处理规则中确定与目标 SDK 的异常类型、异常影响参数对应的异常级别和 SDK 修复策略，确定的该

SDK 修复策略即可作为目标 SDK 修复策略。

异常级别可体现异常的严重程度，不同异常级别对应的目标 SDK 修复策略不同。异常的严重程度越高，修复该异常所需的 SDK 修复策略的修复程度越高。同理，异常的严重程度较低，采用修复程度较低的 SDK 修复策略就可以实现异常的修复。也就是说，体现异常的严重程度越高的异常级别对应的 SDK 修复策略的修复程度也越高。

异常分级处理规则中可记录有多种 SDK 修复策略，不同的 SDK 修复策略的修复程度不同，不同的 SDK 修复策略能够修复不同严重程度的 SDK 异常，可利用异常分级处理规则来为目标 SDK 选择合适的 SDK 修复策略。例如，设存在两种 SDK，分别为 SDK1 和 SDK2，通过两种 SDK 的异常类型和异常影响参数，确定 SDK1 的异常级别为三级，SDK2 的异常级别为一级，异常级别为三级的异常严重程度高于异常级别为一级的严重程度，则为 SDK1 分配修复程度更高的 SDK 修复策略，为 SDK2 分配修复程度更低的 SDK 修复策略。

在一些示例中，目标 SDK 修复策略可包括但不限于以下一项或两项以上：第一文件、第二文件、第三文件。

第一文件包括缓存数据清除指令，缓存数据清除指令用于指示清除目标 SDK 对应的至少一个缓存文件夹中的数据。缓存数据清除指令还可根据修复程度相对的高低细分为不同级别的子指令，例如，缓存数据清除指令包括第一子指令和第二子指令，第一子指令可用于指示清除目标 SDK 对应的个别缓存文件夹中的数据，第二子指令可用于指示清除目标 SDK 对应的所有缓存文件夹中的数据；相对于第一子指令，第二子指令的修复程度更高。缓存数据清除指令可通过指令语句实现，在此并不限定指令语句的类型。

第二文件包括异常源阻断指令。异常源阻断指令用于指示将目标 SDK 的至少一种业务功能和/或子业务功能的开关状态设置为关闭状态。SDK 可具有多种业务功能，每种业务功能还可具有至少一种子业务功能，业务功能与子业务功能可呈树状结构。对应地，业务功能的开关状态与子业务功能的开关状态也可呈树状结构，每种业务功能、子业务功能均可通过设

置开关状态开控制该业务功能、子业务功能的开启或关闭。

异常源阻断指令可根据修复程度相对的高低划分为不同的异常源阻断指令。对于不同异常类型和异常影响参数，可得到不同的异常级别，对应不同的异常级别，可分配不同的异常源阻断指令，不同的异常源阻断指令指示关闭的业务功能和/或子业务功能不同。例如，异常级别为二级，第二文件中的异常源阻断指令指示将目标 SDK 中的业务功能 1 和业务功能 2 的开关状态均设置为关闭状态；异常级别为一级，第二文件中的异常源阻断指令指示将目标 SDK 中的业务功能 1 的开关状态设置为关闭状态。在一些示例中，异常源阻断指令对业务功能的开关状态设置的内容可实现为字符串，如“111110”表示将六种业务功能中的第六种业务功能的开关状态设置为关闭状态。异常源阻断指令可在业务功能和/或子业务功能初始化时执行配置动作。

第三文件包括补丁文件，补丁文件用于对目标 SDK 进行升级修复。补丁文件包括修复前 SDK 的文件与修复后 SDK 的文件的差异文件。补丁文件可针对 SDK 的需要修复处进行修复。在一些示例中，第三文件可为 apk 文件。

在一些示例中，目标 SDK 修复策略包括第三文件。修复平台可生成补丁文件。具体地，修复平台可获取目标 SDK 的修复前 SDK 归档文件和修复后 SDK 归档文件；获取第一 class 文件和第二 class 文件，第一 class 文件包括修复前 SDK 归档文件中的 class 文件，第二 class 文件包括修复后 SDK 归档文件中的 class 文件；利用比对算法，比对第一 class 文件和第二 class 文件，得到第三 class 文件，第三 class 文件包括第二 class 文件与第一 class 文件存在差异的 class 文件；将第三 class 文件转换为补丁文件。

修复前 SDK 归档文件和修复后 SDK 归档文件可为 arr 文件，也可为 jar 文件。若修复前 SDK 归档文件和修复后 SDK 归档文件为 arr 文件，可对 arr 文件进行解压，从中提取 jar 文件。jar 文件包括 class 文件与清单文件，可对 jar 文件解压，从中提取 class 文件，即，得到第一 class 文件和第二 class 文件。可将第三 class 文件压缩为 jar 文件，并通过 Gradle transform 应用程序接口（Application Program Interface, API）将第三 class

文件转化为 dex 文件，dex 文件即为补丁文件。该 dex 文件具有可读性，即该 dex 文件被扫描时，可从中获取到具有可读性的语句、类等，可便于审核人员进行审核，不会出现无法复原为具有可读性语句从而无法审核的情况。

第三文件可实现 SDK 的热修复，针对嵌入不同宿主程序的不同版本的 SDK，第三文件中的补丁文件可不同。

上述第一文件、第二文件、第三文件之间也可相互组合。在一些示例中，两个以上文件组合后得到的目标 SDK 修复策略的修复程度高于两个以上文件中的任意一个文件对应的目标 SDK 修复策略的修复程度。例如，包括第一文件和第二文件的目标 SDK 修复策略的修复程度高于只包括第一文件的目标 SDK 修复策略的修复程度。

在步骤 S204 中，向用户终端下发目标 SDK 修复策略，以使用户终端执行目标 SDK 修复策略以修复目标 SDK。

具体地，修复平台可向目标 SDK 所在的用户终端下发目标 SDK 修复策略。这里并不限定目标 SDK 所在的用户终端的数量，修复平台可向所有目标 SDK 所在的用户终端下发目标 SDK 修复策略，也可采用灰度下发的方式，分批次向目标 SDK 所在的用户终端下发目标 SDK 修复策略，还可向指定的个别目标 SDK 所在的用户终端下发目标 SDK 修复策略。目标 SDK 修复策略可以文件或其他形式发送至用户终端，在此并不限定目标 SDK 修复策略的形式。

用户终端获取到目标 SDK 修复策略后，执行目标 SDK 修复策略以修复目标 SDK。

在一些示例中，用户终端可建立专门用于执行目标 SDK 修复策略的修复进程，即，目标 SDK 修复策略由修复进程执行。该修复进程与目标 SDK 的进程相互独立，即使在目标 SDK 的进程发生中断，修复进程可记录修复进度，不受目标 SDK 的进程的中断影响，在目标 SDK 的进程再次启动时，可根据修复进度继续进行修复，不需要重新开始修复，提高了 SDK 修复的效率。

在本申请实施例中，修复平台能够根据 SDK 信息、宿主信息和采集

到的不同异常类型的 SDK 异常信息，得到表征异常类型对 SDK 的影响的异常影响参数，从而基于每个异常类型对应的异常影响参数和预设的异常分级处理规则，得到对应异常级别的 SDK 修复策略并下发给用户终端。用户终端可执行应异常级别的 SDK 修复策略，对 SDK 进行修复。不同的宿主信息对应不同的宿主程序，不同的宿主程序对应不同的渠道方，不同版本的 SDK 也对应有不同的 SDK 信息，嵌入在不同渠道方的宿主程序中的不同版本的 SDK 产生的 SDK 异常信息不同，通过采集到的 SDK 异常信息，可得到与嵌入每个渠道方的宿主程序的每个版本的 SDK 对应的异常级别的 SDK 修复策略，即，可为嵌入不同渠道方的宿主程序的不同版本的 SDK 自动分配适配的 SDK 修复策略，从而能够满足不同渠道方的 SDK 的修复需求。

在一些实施例中，可根据每个异常类型对 SDK 的影响和每个异常类型的 SDK 异常发生的次数共同确定每个异常类型对应的异常影响参数。还可根据异常影响参数的值最大的异常类型、该异常类型的异常影响参数，结合异常分级处理规则，确定目标 SDK 修复策略。图 4 为本申请第一方面另一实施例提供的 SDK 的修复方法的流程图。图 4 与图 3 的不同之处在于，图 3 中的步骤 S202 可具体细化为图 4 中的步骤 S2021 至步骤 S2024，图 3 中的步骤 S203 可具体细化为图 4 中的步骤 S2031 和步骤 S2032。

在步骤 S2021 中，根据 SDK 信息和宿主信息，确定目标 SDK，并获取目标 SDK 的每个异常类型的 SDK 异常信息中的特征信息，以及特征信息的权重。

修复平台会获取到大量的 SDK 信息、宿主信息和 SDK 异常信息，为了能够为嵌入不同宿主程序的不同版本的 SDK 提供适配的 SDK 修复策略，对于嵌入一种宿主程序的一个版本的 SDK，需要根据与该 SDK 相关的 SDK 异常信息来得到与该 SDK 适配的 SDK 修复策略。

根据 SDK 信息和宿主信息，即可确定 SDK 的版本和 SDK 嵌入的宿主程序，也就确定了目标 SDK。SDK 信息、宿主信息和 SDK 异常信息是成组获取的，即 SDK 发生一次异常，可对应提供 SDK 信息、宿主信息和

SDK 异常信息。确定目标 SDK，可汇总目标 SDK 的每个异常类型的 SDK 异常信息。SDK 异常信息除了异常描述信息以外，还可包括特征信息，特征信息可表征 SDK 异常的至少部分特征。在一些示例中，特征信息可包括但不限于发生异常的 SDK 嵌入的宿主程序的渠道方标识、SDK 发生的异常是否有渠道方反馈、SDK 上线时间、SDK 异常影响的用户、SDK 异常影响的终端类型、发生异常的 SDK 的业务成功率、发生异常的 SDK 的业务重要性等。

可预先为特征信息设置权重，不同种类的特征信息的权重可不同，权重的设置可根据应用场景、需求、经验等设定，在此并不限定。各种类的特征信息的权重之和可为 1。

在步骤 S2022 中，根据特征信息以及特征信息的权重，计算得到目标 SDK 的每个异常类型对应的异常影响因子。

对于一个异常类型的 SDK 异常信息，可先计算该异常类型的 SDK 异常信息中每个种类特征信息的值与各自的权重的乘积，将各个乘积的加和确定为目标 SDK 的该异常类型对应的异常影响因子。在一些示例中，特征信息的值可以是特征信息的原值。在另一些示例中，特征信息的值也可以是特征信息的原值经处理后得到的值，如经过归一化计算、标准化计算后的值，或按照预设的分类规则，根据每个种类特征信息的原值所属的分类，将该分类下的设定值确定为特征信息的值。

例如，特征信息包括发生异常的 SDK 嵌入的宿主程序的渠道方标识、SDK 发生的异常是否有渠道方反馈、发生异常的 SDK 的业务成功率和发生异常的 SDK 的业务重要性。目标 SDK 的某一异常类型对应的发生异常的 SDK 嵌入的宿主程序的渠道方标识为 AA、SDK 发生的异常有渠道方反馈、发生异常的 SDK 的业务成功率为 85%、发生异常的 SDK 的业务重要性为中。预设的分类规则可如下表一至表四所示，表一表征发生异常的 SDK 嵌入的宿主程序的渠道方标识的原值的分类规则，表二表征 SDK 发生的异常是否有渠道方反馈的原值的分类规则，表三表征发生异常的 SDK 的业务成功率的分类规则，表四表征发生异常的 SDK 的业务重要性的分类规则。

表一

分类	AA 和 AB	BB
设定值	c1	c2

表二

分类	有	无
设定值	c3	c4

表三

分类	[0, 50%)	[50%, 75%)	[75%, 100%]
设定值	c5	c6	c7

表四

分类	低	中	高
设定值	c8	c9	c10

若发生异常的 SDK 嵌入的宿主程序的渠道方标识、SDK 发生的异常是否有渠道方反馈、发生异常的 SDK 的业务成功率和发生异常的 SDK 的业务重要性各自对应的权重分别为 d1、d2、d3 和 d4，则按照上述表一和表四，目标 SDK 的该异常类型对应的异常影响因子 α 可根据下式（1）得到：

$$\alpha = c1 \times d1 + c3 \times d2 + c7 \times d3 + c9 \times d4$$

(1)

一种异常类型对应的异常影响因子可表征该异常类型的 SDK 异常对 SDK 的影响，该影响体现的是异常类型在特征信息这个维度上对 SDK 的影响。

在步骤 S2023 中，根据目标 SDK 的每个异常类型的 SDK 异常信息的数量和 SDK 异常信息的总数量，计算得到目标 SDK 的每个异常类型对应的产出比。

目标 SDK 的一个异常类型对应的产出比可表征该异常类型在目标 SDK 的所有异常类型中出现的比重。目标 SDK 的一个异常类型的对应的产出比可为目标 SDK 的该异常类型的 SDK 异常信息的数量和目标 SDK 的所有异常类型的 SDK 异常信息的重数量的比值。例如，目标 SDK 共有 n 个异常类型的异常信息， n 个异常类型的异常信息的数量分别为 b_1 至 b_n ，

则第 i 个异常类型对应的产出比可根据下式 (2) 得到:

$$\beta_i = \frac{b_i}{b_1 + \dots + b_n} \quad (2)$$

其中, β_i 为第 i 个异常类型对应的产出比; b_1 为第 1 个异常类型的异常信息的数量; b_i 为第 i 个异常类型的异常信息的数量, $1 \leq i \leq n$; b_n 为第 n 个异常类型的异常信息的数量。

在步骤 S2024 中, 利用目标 SDK 的每个异常类型对应的异常影响因子和产出比, 计算得到目标 SDK 的每个异常类型对应的异常影响参数。

异常影响参数除了体现异常类型在特征信息这个维度上对 SDK 的影响之外, 还可体现该异常类型在 SDK 的所有异常类型中出现的比重这个维度上对 SDK 的影响。可将目标 SDK 的一个异常类型对应的异常影响因子与该异常类型对应的产出比的乘积作为目标 SDK 的该异常类型对应的异常影响参数。例如, 第 i 个异常类型对应的异常影响参数可根据下式 (3) 得到:

$$E(i) = \alpha_i \times \beta_i \quad (3)$$

其中, $E(i)$ 为第 i 个异常类型对应的异常影响参数; α_i 为第 i 个异常类型对应的异常影响因子; β_i 为第 i 个异常类型对应的产出比。

在步骤 S2031 中, 根据每个异常类型对应的异常影响参数, 确定目标异常类型。

目标异常类型为值最大的异常影响参数对应的异常类型。例如, 假设目标 SDK 共对应具有四种异常类型的 SDK 异常信息, 第一种异常类型对应的异常影响参数的值至第四种异常类型对应的异常影响参数的值依次为 f_1 、 f_2 、 f_3 和 f_4 , 若 $f_2 < f_4 < f_1 < f_3$, 则目标异常类型为第三种异常类型。

在步骤 S2032 中, 依据异常分级处理规则, 确定与目标异常类型和目标异常类型的异常影响参数对应的异常级别, 并根据异常级别, 确定目标 SDK 修复策略。

目标 SDK 的每个异常类型对应的异常影响参数保证该异常类型的 SDK 异常对目标 SDK 的影响程度。值最大的异常影响参数对应的异常类型在各异常类型中, 是对目标 SDK 的影响最大的一种异常类型。目标异常类型和目标异常类型对应的异常影响参数可体现目标 SDK 受到的影响

主要是由哪种异常类型的 SDK 异常引起的，以及该异常类型的 SDK 异常对目标 SDK 的影响程度。不同的目标异常类型以及不同值的异常影响参数在异常分级处理规则中对应的异常级别可能不同，对应的目标 SDK 修复策略也可能不同。

例如，假设异常类型包括但数据异常类型、文件输入输出异常类型、业务不可达异常类型和崩溃异常类型四种异常类型。若目标异常类型为数据异常类型或文件输入输出异常类型，异常级别可为轻微级别，对应的目标 SDK 修复策略可包括上述实施例中的第一文件；若目标异常类型为业务不可达异常类型，异常级别可为中等级别，对应的目标 SDK 修复策略可包括上述实施例中的第二文件；若目标异常类型为崩溃异常类型，异常级别可为严重级别，对应的目标 SDK 修复策略可包括上述实施例中的第三文件。

在一些实施例中，为了使目标 SDK 修复策略下发至用户终端，只能对目标 SDK 生效，可在目标 SDK 修复策略中添加能够体现宿主程序即渠道方的签名。具体地，目标 SDK 修复策略还可包括第一签名文件，修复平台可根据宿主信息，生成第一签名，以及，根据第一签名，生成第一签名文件，第一签名文件包括第一签名。生成第一签名的签名算法在此并不限定，例如，可根据宿主信息，利用 SM3 签名算法，生成第一签名。

用户终端获取到目标 SDK 修复策略后，可通过对第一签名的验签，确定该目标 SDK 修复策略针对的目标 SDK，从而执行目标 SDK 修复策略对目标 SDK 进行修复。可调用宿主程序对第一签名进行验签，验签成功，表示该宿主程序具有目标 SDK，可对该宿主程序嵌入的目标 SDK 进行修复；验签失败，表示该宿主程序不具有目标 SDK，不对该宿主程序嵌入的 SDK 进行修复。

在一些实施例中，在修复平台得到目标 SDK 修复策略后，修复平台可将目标 SDK 修复策略向渠道平台发送，由渠道平台进行审核，在渠道平台审核通过的情况下，才向用户终端下发目标 SDK 修复策略，以保证目标 SDK 修复策略的安全性。图 5 为本申请第一方面又一实施例提供的 SDK 的修复方法的流程图。图 5 与图 3 的不同之处在于，图 5 所示的 SDK

的修复方法还可包括步骤 S205 和步骤 S206，图 3 中的步骤 S204 可具体细化为图 5 中的步骤 S2041。

在步骤 S205 中，向渠道平台发送目标 SDK 修复策略。

渠道平台接收目标 SDK 修复策略，可扫描目标 SDK 修复策略，得到目标 SDK 修复策略的内容，从而对目标 SDK 修复策略进行审核。

若目标 SDK 修复策略审核通过，渠道平台为该目标 SDK 修复策略授权，根据宿主信息生成第二签名，并根据第二签名生成第二签名文件，第二签名文件包括第二签名。渠道平台将第二签名文件添加入目标 SDK 修复策略，并向修复平台反馈目标 SDK 修复策略。在一些示例中，渠道平台可通过自有的签名算法，利用宿主信息对目标 SDK 修复策略的哈希值进行签名，得到第二签名。

若目标 SDK 修复策略审核未通过，渠道平台不会为该目标 SDK 修复策略授权，渠道平台可向修复平台反馈通知消息，通知修复平台审核未通过。在渠道平台扫描目标 SDK 修复策略未授权的情况下，修复平台不得下发目标 SDK 修复策略。

在步骤 S206 中，接收渠道平台反馈的目标 SDK 修复策略。

反馈的目标 SDK 修复策略包括第二签名文件。第二签名文件包括第二签名。第二签名由渠道平台在扫描目标 SDK 修复策略并授权的情况下根据宿主信息生成。

在步骤 S2041 中，向用户终端下发渠道平台反馈的目标 SDK 修复策略。

修复平台可与渠道平台交互，以使渠道平台可对目标 SDK 修复策略进行审核，在渠道平台审核通过的情况下，才允许修复平台向用户终端下发目标 SDK 修复策略，保证了目标 SDK 修复策略的安全性。而且，渠道平台可通过第二签名标识目标 SDK 的宿主程序所属的渠道方，用户终端在获取到目标 SDK 修复策略的情况下，可利用宿主程序的宿主信息对第二签名验签，目标 SDK 修复策略只对验签通过的宿主程序中的目标 SDK 生效，以保证目标 SDK 修复策略生效的针对性。

本申请第二方面提供一种 SDK 的修复方法，可应用于用户终端，即

该 SDK 的修复方法可由用户终端执行。图 6 为本申请第二方面一实施例提供的 SDK 的修复方法的流程图。如图 6 所示，该 SDK 的修复方法可包括步骤 S301 和步骤 S302。

在步骤 S301 中，向修复平台提供软件开发工具包 SDK 信息、宿主信息和不同异常类型的 SDK 异常信息，以使修复平台基于目标 SDK 的每个异常类型对应的异常影响参数和预设的异常分级处理规则得到目标 SDK 修复策略。

目标 SDK 修复策略包括异常分级处理规则中与异常影响参数对应的异常级别的 SDK 修复策略。目标 SDK 与宿主信息指示的宿主程序和 SDK 信息指示的 SDK 版本对应。目标 SDK 的每个异常类型对应的异常影响参数根据 SDK 信息、宿主信息和每个异常类型的 SDK 异常信息得到。

在步骤 S302 中，从修复平台获取目标 SDK 修复策略，并执行目标 SDK 修复策略修复目标 SDK。

在一些示例中，目标 SDK 在初始化时，可建立修复进程，并向修复平台发送宿主信息、SDK 信息、用户信息等，修复平台可根据宿主信息和 SDK 信息，查找与宿主信息和 SDK 信息对应的目标 SDK 修复策略，并下发目标 SDK 修复策略。用户终端从修复平台下载目标 SDK 修复策略，并通过修复进程执行该目标 SDK 修复策略对目标 SDK 进行修复。

在一些示例中，目标 SDK 修复策略包括但不限于以下一项或两项以上：第一文件、第二文件、第三文件。

第一文件包括缓存数据清除指令，缓存数据清除指令用于指示清除目标 SDK 对应的至少一个缓存文件夹中的数据。在目标 SDK 修复策略包括第一文件的情况下，用户终端可按照缓存数据清除指令，清除目标 SDK 的缓存文件夹中的脏数据。

第二文件包括异常源阻断指令，异常源阻断指令用于指示将目标 SDK 的至少一种业务功能和/或子业务功能的开关状态设置为关闭状态。在目标 SDK 修复策略包括第二文件的情况下，用户终端可按照异常源阻断指令将异常源阻断指令指示的业务功能和/或子业务功能的开关状态设置为关闭状态。

第三文件包括补丁文件，补丁文件用于对目标 SDK 进行升级修复。在目标 SDK 修复策略包括第三文件的情况下，用户终端可提取第三文件中的补丁文件，按照类加载的方式加载修复后的类完成目标 SDK 的修复。

在一些示例中，目标 SDK 修复策略包括第三文件，补丁文件由第三 class 文件转换而成，第三 class 文件包括第二 class 文件与第一 class 文件存在差异的 class 文件，第一 class 文件包括目标 SDK 的修复前 SDK 归档文件中的 class 文件，第二 class 文件包括目标 SDK 的修复后 SDK 归档文件中的 class 文件。

上述第一文件、第二文件和第三文件的具体内容可参见上述实施例中的相关说明，在此不再赘述。

上述步骤 S301 和步骤 S302 的具体内容可参见上述实施例中的相关说明，在此不再赘述。

在本申请实施例中，修复平台能够根据 SDK 信息、宿主信息和采集到的不同异常类型的 SDK 异常信息，得到表征异常类型对 SDK 的影响的异常影响参数，从而基于每个异常类型对应的异常影响参数和预设的异常分级处理规则，得到对应异常级别的 SDK 修复策略并下发给用户终端。用户终端可执行应异常级别的 SDK 修复策略，对 SDK 进行修复。不同的宿主信息对应不同的宿主程序，不同的宿主程序对应不同的渠道方，不同版本的 SDK 也对应有不同的 SDK 信息，嵌入在不同渠道方的宿主程序中的不同版本的 SDK 产生的 SDK 异常信息不同，通过采集到的 SDK 异常信息，可得到与嵌入每个渠道方的宿主程序的每个版本的 SDK 对应的异常级别的 SDK 修复策略，即，可为嵌入不同渠道方的宿主程序的不同版本的 SDK 自动分配适配的 SDK 修复策略，从而能够满足不同渠道方的 SDK 的修复需求。

在一些实施例中，目标 SDK 的每个类型异常对应的异常影响参数利用目标 SDK 的每个异常类型对应的异常影响因子和产出比计算得到。目标 SDK 的每个异常类型对应的产出比根据目标 SDK 的每个异常类型的 SDK 异常信息的数量和 SDK 异常信息的总数量计算得到。目标 SDK 的每

个异常类型对应的异常影响因子根据目标 SDK 的每个异常类型的 SDK 异常信息中的特征信息，以及特征信息的权重计算得到。这部分的具体内容可参见上述实施例中的相关说明，在此不再赘述。

在一些实施例中，目标 SDK 修复策略对应的异常级别，依据异常分级处理规则、目标异常类型和目标异常类型的异常影响参数得到，目标异常类型为值最大的异常影响参数对应的异常类型。这部分的具体内容可参见上述实施例中的相关说明，在此不再赘述。

在一些实施例中，目标 SDK 修复策略还包括第一签名文件，第一签名文件包括第一签名，第一签名由修复平台根据宿主信息生成，具体内容可参见上述实施例中的相关说明，在此不再赘述。

用户终端可在从修复平台获取目标 SDK 修复策略后，调用宿主信息指示的宿主程序对第一签名进行验签，在宿主信息指示的宿主程序对第一签名验签成功的情况下，执行目标 SDK 修复策略修复目标 SDK。对第一签名进行验签的验签算法，用户终端可预先与修复平台约定，在此并不限定。修复平台可利用约定的私钥生成第一签名，用户终端可利用约定的公钥对第一签名进行验签，一方面可验证目标 SDK 修复策略的完整性和是否被篡改，另一方面也可验证目标 SDK 修复策略对宿主程序即渠道方的唯一性，从而保证目标 SDK 修复策略的内容安全性和渠道方的接入安全性。

在一些实施例中，目标 SDK 修复策略还包括第二签名文件，第二签名文件包括第二签名，第二签名由渠道平台在扫描目标 SDK 修复策略并确定授权的情况下根据宿主信息生成，具体内容可参见上述实施例中的相关说明，在此不再赘述。图 7 为本申请第二方面另一实施例提供的 SDK 的修复方法的流程图。图 7 与图 6 的不同之处在于，图 6 中的步骤 S302 可具体细化为图 7 中的步骤 S3021 至步骤 S3023。

在步骤 S3021 中，从修复平台获取目标 SDK 修复策略。

在步骤 S3022 中，调用宿主信息指示的宿主程序对第二签名进行验签。

对第二签名进行验签的验签算法，用户终端可预先与渠道平台约定，

在此并不限定。渠道平台可利用约定的私钥生成第二签名，用户终端可利用约定的公钥对第二签名进行验签，第一方面可验证目标 SDK 修复策略的完整性和是否被篡改，第二方面也可验证目标 SDK 修复策略对宿主程序即渠道方的唯一性，第三方面还可验证渠道方对目标 SDK 修复策略的审核通过，从而保证目标 SDK 修复策略的内容安全性和渠道方的接入安全性。

在步骤 S3023 中，在宿主信息指示的宿主程序对第二签名验签成功的情况下，执行目标 SDK 修复策略修复目标 SDK。

在一些实施例中，用户终端可建立修复进程，通过修复进程从修复平台获取目标 SDK 修复策略，并通过修复进程执行目标 SDK 修复策略以修复目标 SDK。该修复进程与目标 SDK 的进程相互独立，可最大化减小目标 SDK 的进程对 SDK 的修复的影响，提高 SDK 的修复效率。图 8 为本申请第二方面又一实施例提供的 SDK 的修复方法的流程图。图 8 与图 6 的不同之处在于，图 6 中的步骤 S302 可具体细化为图 8 中的步骤 S3024 至步骤 S3026。

在步骤 S3024 中，通过修复进程监听目标 SDK 的进程。

在步骤 S3025 中，在目标 SDK 修复完成之前，若监听到目标 SDK 的进程中止且目标 SDK 修复策略未下载完成，则通过修复进程继续下载目标 SDK 修复策略，直至目标 SDK 修复策略下载完成后中断修复进程，在监听到目标 SDK 的进程再次启动时，通过修复进程执行目标 SDK 修复策略修复目标 SDK。

下载得到目标 SDK 修复策略后，可对目标 SDK 修复策略中主要起修复作用的内容加密保存，例如，对缓存数据清除指令、异常源阻断指令、补丁文件等加密保存。

若在修复进程下载目标 SDK 修复策略的过程中，修复进程可能会因为其他原因中止，可利用用户终端的操作系统层提供的断点续传功能，以保证用户终端可获取到完整的目标 SDK 修复策略。

在步骤 S3026 中，在目标 SDK 修复完成之前，若监听到目标 SDK 的进程中止且已开始执行目标 SDK 修复策略，则通过修复进程记录目标

SDK 的修复进度，在监听到目标 SDK 的进程再次启动时，通过修复进程基于修复进度继续修复目标 SDK。

由上可得，无论宿主程序、目标 SDK 的进程是否存活，都不会影响用户终端获取目标 SDK 修复策略，能够确保目标 SDK 修复策略有效执行，提高 SDK 的修复率。

为了便于理解，下面通过用户终端、修复平台和渠道平台之间交互的逻辑图来对 SDK 的修复方法进行说明。图 9 为本申请实施例提供的 SDK 的修复方法的一示例的逻辑示意图。如图 9 所示，用户终端 41 中具有宿主程序 411 和 SDK 412，SDK 412 嵌入宿主程序 411，用户终端 41 可建立修复进程 413。修复平台 42 具有日志分析功能 421、策略配置功能 422 和文件管理功能 423。渠道平台 43 具有渠道方日志提供功能 431、策略内容审核功能 432 和策略签名功能 433。

修复平台 42 可从 SDK 412 获取 SDK 的日志，日志中可包括宿主信息、SDK 信息和 SDK 异常信息。修复平台 42 也可从渠道平台 43 获取渠道方的日志，渠道平台 43 的渠道方日志提供功能 431 可提供渠道方的日志。修复平台 42 中的日志分析功能 421 可从日志中提取宿主信息、SDK 信息和 SDK 异常信息。策略配置功能 422 可从日志分析功能 421 获取宿主信息、SDK 信息和 SDK 异常信息，根据宿主信息、SDK 信息和 SDK 异常信息确定异常级别，并确定异常级别对应的目标 SDK 修复策略。文件管理功能 423 管理各种 SDK 修复策略。策略配置功能 422 可从文件管理功能 423 获取目标 SDK 修复策略。修复平台 42 向渠道平台 43 发送目标 SDK 修复策略。渠道平台 43 通过策略内容审核功能 432 对目标 SDK 修复策略进行审核，在审核通过的情况下，通过策略签名功能 433 对目标 SDK 修复策略进行签名，将添加签名的目标 SDK 修复策略反馈给修复平台 42。SDK 412 可向修复平台发送宿主信息和 SDK 信息，以触发修复平台 42 向用户终端 41 提供目标 SDK 修复策略。SDK 412 可启动修复进程 413，修复进程 413 从修复平台 42 下载目标 SDK 修复策略。用户终端 41 通过修复进程 413 对目标 SDK 修复策略中的签名进行验签，验签通过后，执行该目标 SDK 修复策略，对 SDK 412 进行修复。

需要说明的是，本申请中对信息、数据的获取、存储、使用、处理等均得到用户或相关机构的授权，符合国家法律法规的相关规定。

本申请第三方面提供一种 SDK 的修复装置。图 10 为本申请第三方面一实施例提供的 SDK 的修复装置的结构示意图。如图 10 所示，该 SDK 的修复装置 500 可包括获取模块 501、计算模块 502、策略配置模块 503 和下发模块 504。

获取模块 501 可用于获取 SDK 信息、宿主信息和不同异常类型的 SDK 异常信息。

计算模块 502 可用于根据 SDK 信息、宿主信息和每个异常类型的 SDK 异常信息，得到目标 SDK 的每个异常类型对应的异常影响参数。

目标 SDK 与宿主信息指示的宿主程序和 SDK 信息指示的 SDK 版本对应。

策略配置模块 503 可用于基于目标 SDK 的每个异常类型对应的异常影响参数和预设的异常分级处理规则，得到目标 SDK 修复策略。

在一些示例中，目标 SDK 修复策略包括以下一项或两项以上：

第一文件，第一文件包括缓存数据清除指令，缓存数据清除指令用于指示清除目标 SDK 对应的至少一个缓存文件夹中的数据；

第二文件，第二文件包括异常源阻断指令，异常源阻断指令用于指示将目标 SDK 的至少一种业务功能和/或子业务功能的开关状态设置为关闭状态；

第三文件，第三文件包括补丁文件，补丁文件用于对目标 SDK 进行升级修复。

目标 SDK 修复策略包括异常分级处理规则中与异常影响参数对应的异常级别的 SDK 修复策略。

下发模块，用于向用户终端下发目标 SDK 修复策略，以使用户终端执行目标 SDK 修复策略以修复目标 SDK。

在一些示例中，目标 SDK 修复策略由用户终端建立的修复进程执行，修复进程与目标 SDK 的进程相互独立。

在本申请实施例中，修复平台能够根据 SDK 信息、宿主信息和采集

到的不同异常类型的 SDK 异常信息，得到表征异常类型对 SDK 的影响的异常影响参数，从而基于每个异常类型对应的异常影响参数和预设的异常分级处理规则，得到对应异常级别的 SDK 修复策略并下发给用户终端。用户终端可执行应异常级别的 SDK 修复策略，对 SDK 进行修复。不同的宿主信息对应不同的宿主程序，不同的宿主程序对应不同的渠道方，不同版本的 SDK 也对应有不同的 SDK 信息，嵌入在不同渠道方的宿主程序中的不同版本的 SDK 产生的 SDK 异常信息不同，通过采集到的 SDK 异常信息，可得到与嵌入每个渠道方的宿主程序的每个版本的 SDK 对应的异常级别的 SDK 修复策略，即，可为嵌入不同渠道方的宿主程序的不同版本的 SDK 自动分配适配的 SDK 修复策略，从而能够满足不同渠道方的 SDK 的修复需求。

在一些实施例中，计算模块 502 可用于：根据 SDK 信息和宿主信息，确定目标 SDK，并获取目标 SDK 的每个异常类型的 SDK 异常信息中的特征信息，以及特征信息的权重；根据特征信息以及特征信息的权重，计算得到目标 SDK 的每个异常类型对应的异常影响因子；根据目标 SDK 的每个异常类型的 SDK 异常信息的数量和 SDK 异常信息的总数量，计算得到目标 SDK 的每个异常类型对应的产出比；利用目标 SDK 的每个异常类型对应的异常影响因子和产出比，计算得到目标 SDK 的每个异常类型对应的异常影响参数。

在一些实施例中，策略配置模块 503 可用于：根据每个异常类型对应的异常影响参数，确定目标异常类型，目标异常类型为值最大的异常影响参数对应的异常类型；依据异常分级处理规则，确定与目标异常类型和目标异常类型的异常影响参数对应的异常级别，并根据异常级别，确定目标 SDK 修复策略。

在一些实施例中，目标 SDK 修复策略包括第三文件。SDK 的修复装置 500 还可包括补丁生成模块。

补丁生成模块可用于：获取目标 SDK 的修复前 SDK 归档文件和修复后 SDK 归档文件；获取第一 class 文件和第二 class 文件，第一 class 文件包括修复前 SDK 归档文件中的 class 文件，第二 class 文件包括修复后

SDK 归档文件中的 class 文件；利用比对算法，比对第一 class 文件和第二 class 文件，得到第三 class 文件，第三 class 文件包括第二 class 文件与第一 class 文件存在差异的 class 文件；将第三 class 文件转换为补丁文件。

在一些实施例中，目标 SDK 修复策略还包括第一签名文件。该 SDK 的修复装置还可包括第一签名模块。第一签名模块可用于：根据宿主信息，生成第一签名；根据第一签名，生成第一签名文件，第一签名文件包括第一签名。

在一些实施例中，该 SDK 的修复装置还可包括接收模块。

上述下发模块 504 还可用于：向渠道平台发送目标 SDK 修复策略。

接收模块可用于：接收渠道平台反馈的目标 SDK 修复策略，反馈的目标 SDK 修复策略包括第二签名文件，第二签名文件包括第二签名，第二签名由渠道平台在扫描目标 SDK 修复策略并授权的情况下根据宿主信息生成。

上述下发模块 504 可用于：向用户终端下发渠道平台反馈的目标 SDK 修复策略。

本申请第四方面提供一种用户终端。图 11 为本申请第四方面一实施例提供的用户终端的结构示意图。如图 11 所示，该用户终端 600 可包括发送模块 601、接收模块 602 和执行模块 603。

发送模块 601 可用于向修复平台提供 SDK 信息、宿主信息和不同异常类型的 SDK 异常信息，以使修复平台基于目标 SDK 的每个异常类型对应的异常影响参数和预设的异常分级处理规则得到目标 SDK 修复策略。

目标 SDK 修复策略包括异常分级处理规则中与异常影响参数对应的异常级别的 SDK 修复策略。目标 SDK 与宿主信息指示的宿主程序和 SDK 信息指示的 SDK 版本对应。目标 SDK 的每个异常类型对应的异常影响参数根据 SDK 信息、宿主信息和每个异常类型的 SDK 异常信息得到。

接收模块 602 可用于从修复平台获取目标 SDK 修复策略。

在一些示例中，目标 SDK 修复策略包括以下一项或两项以上：

第一文件，第一文件包括缓存数据清除指令，缓存数据清除指令用于指示清除目标 SDK 对应的至少一个缓存文件夹中的数据；

第二文件，第二文件包括异常源阻断指令，异常源阻断指令用于指示将目标 SDK 的至少一种业务功能和/或子业务功能的开关状态设置为关闭状态；

第三文件，第三文件包括补丁文件，补丁文件用于对目标 SDK 进行升级修复。

在一些示例中，目标 SDK 修复策略包括第三文件，补丁文件由第三 class 文件转换而成，第三 class 文件包括第二 class 文件与第一 class 文件存在差异的 class 文件，第一 class 文件包括目标 SDK 的修复前 SDK 归档文件中的 class 文件，第二 class 文件包括目标 SDK 的修复后 SDK 归档文件中的 class 文件。

执行模块 603 可用于执行目标 SDK 修复策略修复目标 SDK。

在本申请实施例中，修复平台能够根据 SDK 信息、宿主信息和采集到的不同异常类型的 SDK 异常信息，得到表征异常类型对 SDK 的影响的异常影响参数，从而基于每个异常类型对应的异常影响参数和预设的异常分级处理规则，得到对应异常级别的 SDK 修复策略并下发给用户终端。用户终端可执行应异常级别的 SDK 修复策略，对 SDK 进行修复。不同的宿主信息对应不同的宿主程序，不同的宿主程序对应不同的渠道方，不同版本的 SDK 也对应有不同的 SDK 信息，嵌入在不同渠道方的宿主程序中的不同版本的 SDK 产生的 SDK 异常信息不同，通过采集到的 SDK 异常信息，可得到与嵌入每个渠道方的宿主程序的每个版本的 SDK 对应的异常级别的 SDK 修复策略，即，可为嵌入不同渠道方的宿主程序的不同版本的 SDK 自动分配适配的 SDK 修复策略，从而能够满足不同渠道方的 SDK 的修复需求。

在一些实施例中，目标 SDK 的每个类型异常对应的异常影响参数利用目标 SDK 的每个异常类型对应的异常影响因子和产出比计算得到。目标 SDK 的每个异常类型对应的产出比根据目标 SDK 的每个异常类型的 SDK 异常信息的数量和 SDK 异常信息的总数量计算得到。目标 SDK 的每个异常类型对应的异常影响因子根据目标 SDK 的每个异常类型的 SDK 异常信息中的特征信息，以及特征信息的权重计算得到。

在一些实施例中，目标 SDK 修复策略对应的异常级别依据异常分级处理规则、目标异常类型和目标异常类型的异常影响参数得到。目标异常类型为值最大的异常影响参数对应的异常类型。

在一些实施例中，目标 SDK 修复策略还包括第一签名文件，第一签名文件包括第一签名，第一签名由修复平台根据宿主信息生成。

上述执行模块 603 还可用于：调用宿主信息指示的宿主程序对第一签名进行验签，在宿主信息指示的宿主程序对第一签名验签成功的情况下，执行目标 SDK 修复策略修复目标 SDK。

在一些实施例中，目标 SDK 修复策略还包括第二签名文件，第二签名文件包括第二签名，第二签名由渠道平台在扫描目标 SDK 修复策略并确定授权的情况下根据宿主信息生成。

上述执行模块 603 还可用于：调用宿主信息指示的宿主程序对第二签名进行验签，在宿主信息指示的宿主程序对第二签名验签成功的情况下，执行目标 SDK 修复策略修复目标 SDK。

在一些实施例中，上述执行模块 603 可用于：建立修复进程，修复进程与目标 SDK 的进程相互独立；通过修复进程从修复平台获取目标 SDK 修复策略，并执行目标 SDK 修复策略修复目标 SDK。

在一些实施例中，上述执行模块 603 可用于：通过修复进程监听目标 SDK 的进程；在目标 SDK 修复完成之前，若监听到目标 SDK 的进程中止且目标 SDK 修复策略未下载完成，则通过修复进程继续下载目标 SDK 修复策略，直至目标 SDK 修复策略下载完成后中断修复进程，在监听到目标 SDK 的进程再次启动时，通过修复进程执行目标 SDK 修复策略修复目标 SDK；在目标 SDK 修复完成之前，若监听到目标 SDK 的进程中止且已开始执行目标 SDK 修复策略，则通过修复进程记录目标 SDK 的修复进度，在监听到目标 SDK 的进程再次启动时，通过修复进程基于修复进度继续修复目标 SDK。

本申请第五方面还提供了一种电子设备。图 12 为本申请第五方面一实施例提供的电子设备的结构示意图。如图 12 所示，电子设备 700 包括存储器 701、处理器 702 及存储在存储器 701 上并可在处理器 702 上运行

的计算机程序。

在一个示例中，上述处理器 702 可以包括中央处理器（CPU），或者特定集成电路（Application Specific Integrated Circuit，ASIC），或者可以被配置成实施本申请实施例的一个或多个集成电路。

存储器 701 可包括只读存储器（Read-Only Memory，ROM），随机存取存储器（Random Access Memory，RAM），磁盘存储介质设备，光存储介质设备，闪存设备，电气、光学或其他物理/有形的存储器存储设备。因此，通常，存储器包括一个或多个编码有包括计算机可执行指令的软件的有形（非暂态）计算机可读存储介质（例如，存储器设备），并且当该软件被执行（例如，由一个或多个处理器）时，其可操作来执行参考根据本申请第一方面实施例中 SDK 的修复方法所描述的操作。

处理器 702 通过读取存储器 701 中存储的可执行程序代码来运行与可执行程序代码对应的计算机程序，以用于实现上述第一方面实施例中的 SDK 的修复方法。

在一些示例中，电子设备 700 还可包括通信接口 703 和总线 704。其中，如图 12 所示，存储器 701、处理器 702、通信接口 703 通过总线 704 连接并完成相互间的通信。

通信接口 703，主要用于实现本申请实施例中各模块、装置、单元和/或设备之间的通信。也可通过通信接口 703 接入输入设备和/或输出设备。

总线 704 包括硬件、软件或两者，将电子设备 700 的部件彼此耦接在一起。举例来说而非限制，总线 704 可包括加速图形端口（Accelerated Graphics Port，AGP）或其他图形总线、增强工业标准架构（Enhanced Industry Standard Architecture，EISA）总线、前端总线（Front Side Bus，FSB）、超传输（Hyper Transport，HT）互连、工业标准架构（Industry Standard Architecture，ISA）总线、无限带宽互连、低引脚数（Low pin count，LPC）总线、存储器总线、微信道架构（Micro Channel Architecture，MCA）总线、外围组件互连（Peripheral Component Interconnect，PCI）总线、PCI-Express（PCI-E）总线、串行高级技术附件（Serial Advanced Technology Attachment，SATA）总线、视频电子标准协

会局部（Video Electronics Standards Association Local Bus，VLB）总线或其他合适的总线或者两个或更多个以上这些的组合。在合适的情况下，总线 704 可包括一个或多个总线。尽管本申请实施例描述和示出了特定的总线，但本申请考虑任何合适的总线或互连。

本申请第六方面提供一种用户终端，该用户终端可包括存储器、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。

存储器包括一个或多个编码有包括计算机可执行指令的软件的有形（非暂态）计算机可读存储介质（例如，存储器设备），并且当该软件被执行（例如，由一个或多个处理器）时，其可操作来执行参考根据本申请第二方面实施例中 SDK 的修复方法所描述的操作。

处理器通过读取存储器中存储的可执行程序代码来运行与可执行程序代码对应的计算机程序，以用于实现上述第二方面实施例中的 SDK 的修复方法。

在一些示例中，用户终端还可包括通信接口和总线。存储器、处理器、通信接口可通过总线连接并完成相互间的通信。

存储器、处理器、通信接口和总线的连接关系和具体实现可参见上述实施例中电子设备中存储器 701、处理器 702、通信接口 703 和总线 704 的连接关系和具体实现，在此不再赘述。

本申请第七方面提供一种 SDK 的修复系统，该 SDK 的修复系统可包括上述实施例中的修复平台和用户终端。修复平台可执行上述第一方面实施例中的 SDK 的修复方法，用户终端可执行上述第二方面实施例中的 SDK 的修复方法，具体内容可参见上述实施例中的相关说明，且能达到相同的技术效果，为避免重复，这里不再赘述。

在一些实施例中，该 SDK 的修复系统还可包括上述实施例中的渠道平台，渠道平台可执行上述实施例中渠道平台执行的步骤，具体可参见上述实施例中的相关说明，且能达到相同的技术效果，为避免重复，这里不再赘述。

本申请第八方面提供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指令，该计算机程序指令被处理器执行时可实现上

述第一方面的实施例中的 SDK 的修复方法或第二方面的实施例中的 SDK 的修复方法，且能达到相同的技术效果，为避免重复，这里不再赘述。其中，上述计算机可读存储介质可包括非暂态计算机可读存储介质，如只读存储器（Read-Only Memory，简称 ROM）、随机存取存储器（Random Access Memory，简称 RAM）、磁碟或者光盘等，在此并不限定。

本申请实施例还可提供一种计算机程序产品，该计算机程序产品中的指令由电子设备的处理器执行时，使得电子设备执行上述第一方面的实施例中的 SDK 的修复方法或第二方面的实施例中的 SDK 的修复方法，具体可参见上述实施例中的相关说明，且能达到相同的技术效果，为避免重复，这里不再赘述。

需要明确的是，本说明书中的各个实施例均采用递进的方式描述，各个实施例之间相同或相似的部分互相参见即可，每个实施例重点说明的都是与其他实施例的不同之处。对于装置实施例、设备实施例、用户终端实施例、系统实施例、计算机可读存储介质实施例、计算机程序产品实施例而言，相关之处可以参见方法实施例的说明部分。本申请并不局限于上文所描述并在图中示出的特定步骤和结构。本领域的技术人员可以在领会本申请的精神之后，作出各种改变、修改和添加，或者改变步骤之间的顺序。并且，为了简明起见，这里省略对已知方法技术的详细描述。

上面参考根据本申请的实施例的方法、装置（系统）和计算机程序产品的流程图和/或框图描述了本申请的各方面。应当理解，流程图和/或框图中的每个方框以及流程图和/或框图中各方框的组合可以由计算机程序指令实现。这些计算机程序指令可被提供给通用计算机、专用计算机、或其它可编程数据处理装置的处理器，以产生一种机器，使得经由计算机或其它可编程数据处理装置的处理器执行的这些指令使能对流程图和/或框图的一个或多个方框中指定的功能/动作的实现。这种处理器可以是但不限于是通用处理器、专用处理器、特殊应用处理器或者现场可编程逻辑电路。还可理解，框图和/或流程图中的每个方框以及框图和/或流程图中的方框的组合，也可以由执行指定的功能或动作的专用硬件来实现，或可由专用硬件和计算机指令的组合来实现。

本领域技术人员应能理解，上述实施例均是示例性而非限制性的。在不同实施例中出现的不同技术特征可以进行组合，以取得有益效果。本领域技术人员在研究附图、说明书及权利要求书的基础上，应能理解并实现所揭示的实施例的其他变化的实施例。在权利要求书中，术语“包括”并不排除其他装置或步骤；数量词“一个”不排除多个；术语“第一”、“第二”用于标示名称而非用于表示任何特定的顺序。权利要求中的任何附图标记均不应被理解为对保护范围的限制。权利要求中出现的多个部分的功能可以由一个单独的硬件或软件模块来实现。某些技术特征出现在不同的从属权利要求中并不意味着不能将这些技术特征进行组合以取得有益效果。

权 利 要 求 书

1. 一种 SDK 的修复方法，应用于修复平台，所述方法包括：

获取软件开发工具包 SDK 信息、宿主信息和不同异常类型的 SDK 异常信息；

根据所述 SDK 信息、所述宿主信息和每个异常类型的所述 SDK 异常信息，得到目标 SDK 的每个异常类型对应的异常影响参数，所述目标 SDK 与所述宿主信息指示的宿主程序和所述 SDK 信息指示的 SDK 版本对应；

基于所述目标 SDK 的每个异常类型对应的所述异常影响参数和预设的异常分级处理规则，得到目标 SDK 修复策略，所述目标 SDK 修复策略包括所述异常分级处理规则中与所述异常影响参数对应的异常级别的 SDK 修复策略；

向用户终端下发所述目标 SDK 修复策略，以使所述用户终端执行所述目标 SDK 修复策略以修复所述目标 SDK。

2. 根据权利要求 1 所述的方法，其中，所述根据所述 SDK 信息、所述宿主信息和每个异常类型的所述 SDK 异常信息，得到目标 SDK 的每个异常类型对应的异常影响参数，包括：

根据所述 SDK 信息和所述宿主信息，确定所述目标 SDK，并获取所述目标 SDK 的每个异常类型的所述 SDK 异常信息中的特征信息，以及所述特征信息的权重；

根据所述特征信息以及所述特征信息的权重，计算得到所述目标 SDK 的每个异常类型对应的异常影响因子；

根据所述目标 SDK 的每个异常类型的所述 SDK 异常信息的数量和所述 SDK 异常信息的总数量，计算得到所述目标 SDK 的每个异常类型对应的产出比；

利用所述目标 SDK 的每个异常类型对应的所述异常影响因子和所述产出比，计算得到所述目标 SDK 的每个异常类型对应的所述异常影响参数。

3. 根据权利要求 1 所述的方法, 其中, 所述基于所述目标 SDK 的每个异常类型对应的所述异常影响参数和预设的异常分级处理规则, 得到目标 SDK 修复策略, 包括:

根据每个异常类型对应的所述异常影响参数, 确定目标异常类型, 所述目标异常类型为值最大的所述异常影响参数对应的异常类型;

依据所述异常分级处理规则, 确定与所述目标异常类型和所述目标异常类型的所述异常影响参数对应的异常级别, 并根据异常级别, 确定所述目标 SDK 修复策略。

4. 根据权利要求 1 所述的方法, 其中, 所述目标 SDK 修复策略包括以下一项或两项以上:

第一文件, 所述第一文件包括缓存数据清除指令, 所述缓存数据清除指令用于指示清除所述目标 SDK 对应的至少一个缓存文件夹中的数据;

第二文件, 所述第二文件包括异常源阻断指令, 所述异常源阻断指令用于指示将所述目标 SDK 的至少一种业务功能和/或子业务功能的开关状态设置为关闭状态;

第三文件, 所述第三文件包括补丁文件, 所述补丁文件用于对所述目标 SDK 进行升级修复。

5. 根据权利要求 4 所述的方法, 其中, 所述目标 SDK 修复策略包括所述第三文件,

所述方法还包括:

获取所述目标 SDK 的修复前 SDK 归档文件和修复后 SDK 归档文件;

获取第一 class 文件和第二 class 文件, 所述第一 class 文件包括修复前 SDK 归档文件中的 class 文件, 所述第二 class 文件包括修复后 SDK 归档文件中的 class 文件;

利用比对算法, 比对所述第一 class 文件和所述第二 class 文件, 得到第三 class 文件, 所述第三 class 文件包括所述第二 class 文件与所述第一 class 文件存在差异的 class 文件;

将所述第三 class 文件转换为所述补丁文件。

6. 根据权利要求 1 所述的方法, 其中, 所述目标 SDK 修复策略还包

括第一签名文件，

所述方法还包括：

根据所述宿主信息，生成第一签名；

根据所述第一签名，生成第一签名文件，所述第一签名文件包括所述第一签名。

7. 根据权利要求 1 所述的方法，在所述按照预设的异常分级处理规则，基于所述目标 SDK 的每类所述 SDK 异常信息的所述异常影响因子，得到目标 SDK 修复策略之后，还包括：

向渠道平台发送所述目标 SDK 修复策略；

接收所述渠道平台反馈的所述目标 SDK 修复策略，反馈的所述目标 SDK 修复策略包括第二签名文件，所述第二签名文件包括第二签名，所述第二签名由所述渠道平台在扫描所述目标 SDK 修复策略并授权的情况下根据所述宿主信息生成；

所述向所述用户终端下发所述目标 SDK 修复策略，包括：

向所述用户终端下发所述渠道平台反馈的所述目标 SDK 修复策略。

8. 根据权利要求 1 所述的方法，其中，所述目标 SDK 修复策略由所述用户终端建立的修复进程执行，所述修复进程与所述目标 SDK 的进程相互独立。

9. 一种 SDK 的修复方法，应用于用户终端，所述方法包括：

向修复平台提供软件开发工具包 SDK 信息、宿主信息和不同异常类型的 SDK 异常信息，以使所述修复平台基于目标 SDK 的每个异常类型对应的异常影响参数和预设的异常分级处理规则得到目标 SDK 修复策略，所述目标 SDK 修复策略包括所述异常分级处理规则中与所述异常影响参数对应的异常级别的 SDK 修复策略，所述目标 SDK 与所述宿主信息指示的宿主程序和所述 SDK 信息指示的 SDK 版本对应，所述目标 SDK 的每个异常类型对应的所述异常影响参数根据所述 SDK 信息、所述宿主信息和每个异常类型的所述 SDK 异常信息得到；

从修复平台获取所述目标 SDK 修复策略，并执行所述目标 SDK 修复策略修复所述目标 SDK。

10. 根据权利要求 9 所述的方法，其中，

所述目标 SDK 的每个类型异常对应的所述异常影响参数利用所述目标 SDK 的每个异常类型对应的异常影响因子和产出比计算得到；所述目标 SDK 的每个异常类型对应的所述产出比根据所述目标 SDK 的每个异常类型的所述 SDK 异常信息的数量和所述 SDK 异常信息的总数量计算得到；所述目标 SDK 的每个异常类型对应的所述异常影响因子根据所述目标 SDK 的每个异常类型的所述 SDK 异常信息中的特征信息，以及所述特征信息的权重计算得到。

11. 根据权利要求 9 所述的方法，其中，

所述目标 SDK 修复策略对应的异常级别依据所述异常分级处理规则、目标异常类型和所述目标异常类型的所述异常影响参数得到，所述目标异常类型为值最大的所述异常影响参数对应的异常类型。

12. 根据权利要求 9 所述的方法，其中，所述目标 SDK 修复策略包括以下一项或两项以上：

第一文件，所述第一文件包括缓存数据清除指令，所述缓存数据清除指令用于指示清除所述目标 SDK 对应的至少一个缓存文件夹中的数据；

第二文件，所述第二文件包括异常源阻断指令，所述异常源阻断指令用于指示将所述目标 SDK 的至少一种业务功能和/或子业务功能的开关状态设置为关闭状态；

第三文件，所述第三文件包括补丁文件，所述补丁文件用于对所述目标 SDK 进行升级修复。

13. 根据权利要求 12 所述的方法，其中，

所述目标 SDK 修复策略包括所述第三文件，所述补丁文件由第三 class 文件转换而成，所述第三 class 文件包括第二 class 文件与第一 class 文件存在差异的 class 文件，所述第一 class 文件包括所述目标 SDK 的修复前 SDK 归档文件中的 class 文件，所述第二 class 文件包括所述目标 SDK 的修复后 SDK 归档文件中的 class 文件。

14. 根据权利要求 9 所述的方法，其中，所述目标 SDK 修复策略还包括第一签名文件，所述第一签名文件包括第一签名，所述第一签名由所

述修复平台根据所述宿主信息生成，

在所述从修复平台获取所述目标 SDK 修复策略之后，还包括：

调用所述宿主信息指示的宿主程序对所述第一签名进行验签；

所述执行所述目标 SDK 修复策略修复目标 SDK，包括：

在所述宿主信息指示的宿主程序对所述第一签名验签成功的情况下，
执行所述目标 SDK 修复策略修复目标 SDK。

15. 根据权利要求 9 所述的方法，其中，所述目标 SDK 修复策略还包括第二签名文件，所述第二签名文件包括第二签名，所述第二签名由渠道平台在扫描所述目标 SDK 修复策略并确定授权的情况下根据所述宿主信息生成，

在所述从修复平台获取所述目标 SDK 修复策略之后，还包括：

调用所述宿主信息指示的宿主程序对所述第二签名进行验签；

所述执行所述目标 SDK 修复策略修复目标 SDK，包括：

在所述宿主信息指示的宿主程序对所述第二签名验签成功的情况下，
执行所述目标 SDK 修复策略修复目标 SDK。

16. 根据权利要求 9 至 15 中任意一项所述的方法，其中，所述从修复平台获取所述目标 SDK 修复策略，并执行所述目标 SDK 修复策略修复目标 SDK，包括：

建立修复进程，所述修复进程与所述目标 SDK 的进程相互独立；

通过所述修复进程从所述修复平台获取所述目标 SDK 修复策略，并
执行所述目标 SDK 修复策略修复所述目标 SDK。

17. 根据权利要求 16 所述的方法，其中，所述通过所述修复进程从所述修复平台获取所述目标 SDK 修复策略，并执行所述目标 SDK 修复策略修复所述目标 SDK，包括：

通过所述修复进程监听所述目标 SDK 的进程；

在所述目标 SDK 修复完成之前，若监听到所述目标 SDK 的进程中止且所述目标 SDK 修复策略未下载完成，则通过所述修复进程继续下载所述目标 SDK 修复策略，直至所述目标 SDK 修复策略下载完成后中断所述修复进程，在监听到所述目标 SDK 的进程再次启动时，通过所述修复进

程执行所述目标 SDK 修复策略修复所述目标 SDK;

在所述目标 SDK 修复完成之前,若监听到所述目标 SDK 的进程中止且已开始执行所述目标 SDK 修复策略,则通过修复进程记录所述目标 SDK 的修复进度,在监听到所述目标 SDK 的进程再次启动时,通过所述修复进程基于所述修复进度继续修复所述目标 SDK。

18. 一种 SDK 的修复装置,应用于修复平台,所述装置包括:

获取模块,用于获取软件开发工具包 SDK 信息、宿主信息和不同异常类型的 SDK 异常信息;

计算模块,用于根据所述 SDK 信息、所述宿主信息和每个异常类型的所述 SDK 异常信息,得到目标 SDK 的每个异常类型对应的异常影响参数,所述目标 SDK 与所述宿主信息指示的宿主程序和所述 SDK 信息指示的 SDK 版本对应;

策略配置模块,用于基于所述目标 SDK 的每个异常类型对应的所述异常影响参数和预设的异常分级处理规则,得到目标 SDK 修复策略,所述目标 SDK 修复策略包括所述异常分级处理规则中与所述异常影响参数对应的异常级别的 SDK 修复策略;

下发模块,用于向所述用户终端下发所述目标 SDK 修复策略,以使用户终端执行所述目标 SDK 修复策略以修复所述目标 SDK。

19. 一种用户终端,包括:

发送模块,用于向修复平台提供软件开发工具包 SDK 信息、宿主信息和不同异常类型的 SDK 异常信息,以使所述修复平台基于目标 SDK 的每个异常类型对应的异常影响参数和预设的异常分级处理规则得到目标 SDK 修复策略,所述目标 SDK 修复策略包括所述异常分级处理规则中与所述异常影响参数对应的异常级别的 SDK 修复策略,所述目标 SDK 与所述宿主信息指示的宿主程序和所述 SDK 信息指示的 SDK 版本对应,所述目标 SDK 的每个异常类型对应的所述异常影响参数根据所述 SDK 信息、所述宿主信息和每个异常类型的所述 SDK 异常信息得到;

接收模块,用于从修复平台获取所述目标 SDK 修复策略;

执行模块,用于执行所述目标 SDK 修复策略修复所述目标 SDK。

20. 一种电子设备，应用于修复平台，所述电子设备包括：处理器以及存储有计算机程序指令的存储器；

所述处理器执行所述计算机程序指令时实现如权利要求 1 至 8 中任意一项所述的 SDK 的修复方法。

21. 一种用户终端，包括：处理器以及存储有计算机程序指令的存储器；

所述处理器执行所述计算机程序指令时实现如权利要求 9 至 17 中任意一项所述的 SDK 的修复方法。

22. 一种 SDK 的修复系统，包括：

修复平台，用于执行如权利要求 1 至 8 中任意一项所述的 SDK 的修复方法；

用户终端，用于执行如权利要求 9 至 17 中任意一项所述的 SDK 的修复方法。

23. 一种计算机可读存储介质，所述计算机可读存储介质上存储有计算机程序指令，所述计算机程序指令被处理器执行时实现如权利要求 1 至 17 中任意一项所述的 SDK 的修复方法。