



**(19) 대한민국특허청(KR)**  
**(12) 공개특허공보(A)**

(11) 공개번호 10-2018-0098584  
(43) 공개일자 2018년09월04일

(51) 국제특허분류(Int. Cl.)  
G06F 9/44 (2018.01) G06F 8/61 (2018.01)  
(52) CPC특허분류  
G06F 9/44 (2013.01)  
G06F 8/61 (2013.01)  
(21) 출원번호 10-2018-7020535  
(22) 출원일자(국제) 2016년12월14일  
심사청구일자 없음  
(85) 번역문제출일자 2017년07월17일  
(86) 국제출원번호 PCT/CN2016/109917  
(87) 국제공개번호 WO 2017/107824  
국제공개일자 2017년06월29일  
(30) 우선권주장  
201510971750.8 2015년12월22일 중국(CN)

(71) 출원인  
**알리바바 그룹 홀딩 리미티드**  
케이만군도, 그랜드 케이만, 피오박스 847, 원 캐피탈 플레이스 4층  
(72) 발명자  
**정 웨양**  
중국 항저우 310099 완탕 로드 넘버 18 후양룽 타임즈 플라자 빌딩 비 17층 앤츠 패턴 팀 내  
**우 이평**  
중국 항저우 310099 완탕 로드 넘버 18 후양룽 타임즈 플라자 빌딩 비 17층 앤츠 패턴 팀 내  
**바이 하오관**  
중국 항저우 310099 완탕 로드 넘버 18 후양룽 타임즈 플라자 빌딩 비 17층 앤츠 패턴 팀 내  
(74) 대리인  
**김태홍, 김진희**

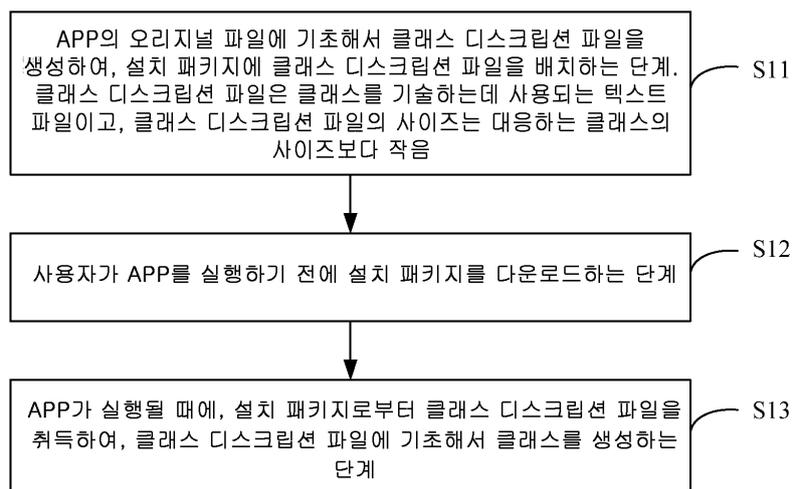
전체 청구항 수 : 총 11 항

(54) 발명의 명칭 **APP 프로그램 실행 방법 및 장치**

**(57) 요약**

본 개시내용은 APP 실행 방법 및 장치를 제안하고, APP 실행 방법은, APP의 오리지널 파일에 기초해서 클래스 디스크립션 파일을 생성하여, 설치 패키지에 상기 클래스 디스크립션 파일을 배치하는 단계로서, 상기 클래스 디스크립션 파일은 클래스를 기술하는데 사용되는 텍스트 파일이고, 클래스 디스크립션 파일의 사이즈는 대응하는 클래스의 사이즈보다 작은 것인 상기 클래스 디스크립션 파일을 생성하여 배치하는 단계와, 사용자가 상기 APP를 실행하기 전에 상기 설치 패키지를 다운로드하는 단계와, 상기 APP가 실행될 때에, 상기 설치 패키지로부터 클래스 디스크립션 파일을 취득하여, 상기 클래스 디스크립션 파일에 기초해서 클래스를 생성하는 단계를 포함한다. 본 방법은 APP 설치 패키지의 사이즈를 줄여서 리소스 오버헤드를 감소시킬 수 있다.

**대표도** - 도1



## 명세서

### 청구범위

#### 청구항 1

애플리케이션(APP) 실행 방법에 있어서,

APP의 오리지널 파일에 기초해서 클래스 디스크립션 파일(class description file)을 생성하여, 설치 패키지(installation package)에 상기 클래스 디스크립션 파일을 배치하는 단계—상기 클래스 디스크립션 파일은 클래스를 기술하는데 사용되는 텍스트 파일이고, 상기 클래스 디스크립션 파일의 사이즈는 대응하는 클래스의 사이즈보다 작음—와,

사용자가 APP를 실행하기 전에 상기 설치 패키지를 다운로드하는 단계와,

APP가 실행될 때에, 상기 설치 패키지로부터 상기 클래스 디스크립션 파일을 취득하여, 상기 클래스 디스크립션 파일에 기초해서 클래스를 생성하는 단계

를 포함하는 APP 실행 방법.

#### 청구항 2

제1항에 있어서, 상기 APP의 오리지널 파일에 기초해서 클래스 디스크립션 파일을 생성하는 것은,

미리 결정된 스크립트에 기초하여 포맷 변환을 수행하는 것과, 상기 APP의 오리지널 파일을 상기 클래스 디스크립션 파일로 변환하는 것을 포함하는 것인 APP 실행 방법.

#### 청구항 3

제1항에 있어서, 상기 클래스 디스크립션 파일에 기초해서 클래스를 생성하는 단계는,

상기 클래스 디스크립션 파일에 기초하여, 클래스명, 패런트(parent) 클래스명, 및 속성을 취득하는 단계와,

상기 클래스명, 상기 패런트 클래스명, 및 상기 속성에 기초하여 그리고 iOS 런타임 기술이 제공하는 메소드를 이용하여 클래스를 생성하는 단계를 포함하는 것인 APP 실행 방법.

#### 청구항 4

제1항 내지 제3항 중 어느 한 항에 있어서, 상기 오리지널 파일은 헤더 파일과 소스 파일을 포함하는 것인 APP 실행 방법.

#### 청구항 5

제4항에 있어서, 상기 APP의 오리지널 파일에 기초해서 클래스 디스크립션 파일을 생성하여, 설치 패키지에 상기 클래스 디스크립션 파일을 배치하는 단계는,

상기 헤더 파일과 상기 소스 파일에 기초하여 상기 클래스 디스크립션 파일을 생성하는 단계와,

상기 소스 파일을 삭제하는 단계와,

상기 소스 파일이 삭제된 후에, 사용자가 다운로드하기 위해 상기 설치 패키지에 상기 헤더 파일과 상기 클래스 디스크립션 파일을 배치하는 단계를 포함하는 것인 APP 실행 방법.

#### 청구항 6

제5항에 있어서, 상기 설치 패키지에 상기 헤더 파일과 상기 클래스 디스크립션 파일을 배치한 후에,

상기 설치 패키지를 압축하여 사용자가 압축된 설치 패키지를 다운로드하게 하는 단계를 더 포함하는 APP 실행 방법.

#### 청구항 7

애플리케이션(APP) 실행 장치에 있어서,

APP의 오리지널 파일에 기초해서 클래스 디스크립션 파일을 생성하여, 설치 패키지에 상기 클래스 디스크립션 파일을 배치하도록 구성된 생성 모듈로서, 상기 클래스 디스크립션 파일은 클래스를 기술하는데 사용되는 텍스트 파일이고, 상기 클래스 디스크립션 파일의 사이즈는 대응하는 클래스의 사이즈보다 작은 것인 상기 생성 모듈과,

사용자가 APP를 실행하기 전에 상기 설치 패키지를 다운로드하도록 구성된 다운로드 모듈과,

APP가 실행될 때에, 상기 설치 패키지로부터 상기 클래스 디스크립션 파일을 취득하여, 상기 클래스 디스크립션 파일에 기초해서 클래스를 생성하도록 구성된 실행 모듈

을 포함하는 APP 실행 장치.

#### 청구항 8

제7항에 있어서, 상기 생성 모듈은, 상기 APP의 오리지널 파일에 기초해서 클래스 디스크립션 파일을 생성하도록 구성되고, 상기 클래스 디스크립션 파일의 생성은,

미리 결정된 스크립트에 기초하여 포맷 변환을 수행하는 것과, 상기 APP의 오리지널 파일을 상기 클래스 디스크립션 파일로 변환하는 것을 포함하는 것인 APP 실행 장치.

#### 청구항 9

제7항에 있어서, 상기 실행 모듈은, 상기 클래스 디스크립션 파일에 기초하여 클래스를 생성하도록 구성되고, 상기 클래스의 생성은,

상기 클래스 디스크립션 파일에 기초하여, 클래스명, 패런트 클래스명, 및 속성을 취득하는 것과,

상기 클래스명, 상기 패런트 클래스명, 및 상기 속성에 기초하여 그리고 iOS 런타임 기술이 제공하는 메소드를 이용하여 클래스를 생성하는 것을 포함하는 것인 APP 실행 장치.

#### 청구항 10

제7항 내지 제9항 중 어느 한 항에 있어서, 상기 오리지널 파일은 헤더 파일과 소스 파일을 포함하고, 상기 생성 모듈은,

상기 헤더 파일과 상기 소스 파일에 기초하여 상기 클래스 디스크립션 파일을 생성하도록 구성된 제1 유닛과,

상기 소스 파일을 삭제하도록 구성된 제2 유닛과,

상기 소스 파일이 삭제된 후에, 상기 설치 패키지에 상기 헤더 파일과 상기 클래스 디스크립션 파일을 배치하여 사용자가 다운로드하게 하도록 구성된 제3 유닛을 포함하는 것인 APP 실행 장치.

#### 청구항 11

제10항에 있어서, 상기 생성 모듈은,

상기 설치 패키지를 압축하여 사용자가 압축된 설치 패키지를 다운로드하게 하도록 구성된 제4 유닛을 더 포함하는 것인 APP 실행 장치.

### 발명의 설명

#### 기술 분야

[0001] 본 개시내용은 인터넷 기술 분야에 관한 것으로서, 구체적으로는 APP 실행 방법 및 장치에 관한 것이다.

#### 배경 기술

[0002] 스마트 기술 시대에서는, 사용자들이 점점 더 많은 애플리케이션(APP)을 사용하는데, APP는 사용 전에 다운로드되어 설치되어야 한다. 일반적으로, APP의 설치 패키지는 예컨대 약 100 M로 대개 대형이다. 사용자가 APP 스토어로부터 설치 패키지를 다운로드할 때에, 단말 디바이스의 대형 디스크 공간이 요구된다. 또한, 대규모 다운로드

드 트래픽 및 긴 다운로드 시간이 필요하여, 리소스 오버헤드를 상승시킨다.

### 발명의 내용

- [0003] 본 개시내용은 관련 기술에 있어서의 기술적 문제 중 하나를 적어도 소정 범위까지 해결하는 것을 목적으로 한다.
- [0004] 따라서, 본 개시내용의 목표는 애플리케이션(APP) 실행 방법을 제공하는 것이고, 이 방법은 APP 설치 패키지의 사이즈를 줄여서, 리소스 오버헤드를 감소시킬 수 있다.
- [0005] 본 개시내용의 다른 목표는 APP 실행 장치를 제공하는 것이다.
- [0006] 전술한 목표를 달성하기 위하여, 본 개시내용의 제1 양태의 구현예에서 제안하는 APP 실행 방법은, APP의 오리지널 파일에 기초해서 클래스 디스크립션 파일(class description file)을 생성하여, 설치 패키지에 상기 클래스 디스크립션 파일을 배치하는 단계로서, 상기 클래스 디스크립션 파일은 클래스를 기술하는데 사용되는 텍스트 파일이고, 상기 클래스 디스크립션 파일의 사이즈는 대응하는 클래스의 사이즈보다 작은 것인 상기 클래스 디스크립션 파일을 생성하여 배치하는 단계와, 사용자가 상기 APP를 실행하기 전에 상기 설치 패키지를 다운로드하는 단계와, 상기 APP가 실행될 때에, 상기 설치 패키지로부터 클래스 디스크립션 파일을 취득하여, 상기 클래스 디스크립션 파일에 기초해서 클래스를 생성하는 단계를 포함한다.
- [0007] 본 개시내용의 제1 양태의 구현예에서 제안하는 APP 실행 방법에 있어서, 상기 설치 패키지는 클래스 디스크립션 파일을 포함하고, 상기 클래스 디스크립션 파일의 사이즈는 상기 클래스의 사이즈보다 작다. 따라서, 설치 패키지에 클래스를 직접 포함시키는 기존의 방법과 비교할 때에, 설치 패키지에 클래스 디스크립션 파일을 포함시키면 APP 설치 패키지의 사이즈를 줄여서, 리소스 오버헤드를 감소시킬 수 있다.
- [0008] 전술한 목표를 달성하기 위하여, 본 개시내용의 제2 양태의 구현예에서 제안하는 APP 실행 장치는, APP의 오리지널 파일에 기초해서 클래스 디스크립션 파일을 생성하여, 설치 패키지에 상기 클래스 디스크립션 파일 - 상기 클래스 디스크립션 파일은 클래스를 기술하는데 사용되는 텍스트 파일이고, 상기 클래스 디스크립션 파일의 사이즈는 대응하는 클래스의 사이즈보다 작음 - 을 배치하도록 구성된 생성 모듈과, 사용자가 상기 APP를 실행하기 전에 상기 설치 패키지를 다운로드하도록 구성된 다운로드 모듈과, 상기 APP가 실행될 때에, 상기 설치 패키지로부터 클래스 디스크립션 파일을 취득하여, 상기 클래스 디스크립션 파일에 기초해서 클래스를 생성하도록 구성된 실행 모듈을 포함한다.
- [0009] 본 개시내용의 제2 양태의 구현예에서 제안하는 APP 실행 장치에 있어서, 상기 설치 패키지는 클래스 디스크립션 파일을 포함하고, 상기 클래스 디스크립션 파일의 사이즈는 상기 클래스의 사이즈보다 작다. 따라서, 설치 패키지에 클래스를 직접 포함시키는 기존의 방법과 비교할 때에, 설치 패키지에 클래스 디스크립션 파일을 포함시키면 APP 설치 패키지의 사이즈를 줄여서, 리소스 오버헤드를 감소시킬 수 있다.
- [0010] 본 개시내용의 몇몇 추가 양태 및 이점은 이어지는 설명에 제공되거나 다음의 설명에서 명백해지며, 본 개시내용의 실시를 통해 이해될 수도 있다.

### 도면의 간단한 설명

- [0011] 본 개시내용의 전술한 양태와 이점 그리고/또는 추가의 양태와 이점은 첨부 도면을 참조한 이하의 구현예의 설명에서 명백해지며 이해될 수 있다.
- 도 1은 본 개시내용의 구현예에 따른 APP 실행 방법을 예시하는 개략 흐름도이다.
- 도 2는 기존의 기술에 따른 APP 실행 방법을 예시하는 개략도이다.
- 도 3은 본 개시내용의 또 다른 구현예에 따른 APP 실행 방법을 예시하는 개략 흐름도이다.
- 도 4는 본 개시내용의 구현예에 따라, 클래스 디스크립션 파일에 기초하여 클래스를 생성하는 단계의 개략 흐름도이다.
- 도 5는 본 개시내용의 구현예에 따른 APP 실행 장치의 개략 구조도이다.
- 도 6은 본 개시내용의 또 다른 구현예에 따른 APP 실행 장치의 개략 구조도이다.

### 발명을 실시하기 위한 구체적인 내용

- [0012] 이하, 본 개시내용의 구현예들에 대해 자세하게 설명하며, 구현예들은 첨부 도면에 도시된다. 동일하거나 유사한 참조 번호는 동일하거나 유사한 모듈 또는 동일하거나 유사한 기능을 갖는 모듈을 표현한다. 첨부 도면을 참조하여 후술하는 구현예들은 본 개시내용을 설명하기 위해 의도된 예시일 뿐이며, 본 개시내용의 제한으로서 해석되어서는 안 된다. 반면, 본 개시내용의 구현예들은 첨부하는 청구범위의 사상 및 범위 내에 있는 모든 변경, 변형 및 등가물을 포함한다.
- [0013] 도 1은 본 개시내용의 구현예에 따른 애플리케이션(APP) 실행 방법을 예시하는 개략 흐름도이다. 이 방법은 다음의 단계를 포함한다.
- [0014] S11. APP의 오리지널 파일에 기초해서 클래스 디스크립션 파일을 생성하여, 설치 패키지에 클래스 디스크립션 파일을 배치하는 단계. 클래스 디스크립션 파일은 클래스를 기술하는데 사용되는 텍스트 파일이고, 클래스 디스크립션 파일의 사이즈는 대응하는 클래스의 사이즈보다 작다.
- [0015] 예를 들어, 오리지널 파일을 헤더 파일(.h 파일)과 소스 파일(.m 파일)을 포함한다.
- [0016] 클래스 디스크립션 파일은 텍스트 파일로서 클래스의 관련 정보를 기술하는데 사용된다. 클래스는 컴퓨터 프로그래밍의 기본 정보이다. 클래스 디스크립션 파일의 사이즈는 대응하는 클래스의 사이즈보다 작다.
- [0017] 예를 들어, 클래스명(className)이 캐서(MERCHANTAPPOperator)이고, 캐서는 클래스 디스크립션(classDescription)을 갖는다. 클래스 디스크립션은 속성 디스크립션을 포함하는데, 예컨대 캐서의 캐서 ID(operatorId), 캐서 명칭(operatorName), 샵 명칭(shopName) 등을 포함할 수 있다.
- [0018] 예로서 iOS를 사용한다면, 사전 컴파일된 스크립트가 오리지널 파일을 클래스 디스크립션 파일(.plist 파일)로 변환하는 데에 이용될 수 있다. 클래스 디스크립션 파일에 필요한 콘텐츠를 오리지널 파일에서 찾을 수 있지만, 오리지널 파일과 클래스 디스크립션 파일은 상이한 포맷을 갖는다. 이에, 오리지널 파일과 클래스 디스크립션 파일 간의 변환을 구현하기 위해 스크립트가 사전 컴파일될 수 있다.
- [0019] 또한, 클래스 디스크립션 파일은 생성된 후에, 리소스 번들(resource bundle)과 같은 미리 결정된 디렉토리에 저장될 수 있다.
- [0020] 클래스 디스크립션 파일은 생성된 후에, 사용자가 다운로드하여 설치할 수 있도록 설치 패키지에 배치될 수 있다.
- [0021] 일부 구현예에 있어서, 도 2를 참조하면, APP의 오리지널 파일에 기초해서 클래스 디스크립션 파일을 생성하여, 설치 패키지에 클래스 디스크립션 파일을 배치하는 단계는 다음의 단계들을 포함한다.
- [0022] S111. 헤더 파일과 소스 파일에 기초하여 클래스 디스크립션 파일을 생성하는 단계.
- [0023] 생성 프로시저에 대해서는, 전술한 설명을 참조하면 된다. 여기서는 다시 상세하게 설명하지 않는다.
- [0024] S112. 소스 파일을 삭제하는 단계.
- [0025] S113. 소스 파일이 삭제된 후에, 설치 패키지에 헤더 파일과 클래스 디스크립션 파일을 배치하여 사용자가 다운로드하게 하는 단계.
- [0026] 클래스 디스크립션 파일과 헤더 파일 둘 다 작고, 취득된 설치 패키지도 작기 때문에, 사용자는 APP 스토어에서 설치 패키지를 직접 다운로드할 수 있다.
- [0027] 선택사항으로서, 설치 패키지의 사이즈를 더욱 줄이기 위해, 취득되는 설치 패키지는 또한 압축될 수 있다. 예를 들어, 헤더 파일과 클래스 디스크립션 파일이 설치 패키지에 포함된 후에, 본 방법은 다음의 단계들을 더 포함할 수 있다.
- [0028] S114. 설치 패키지를 압축하여 사용자가 압축된 설치 패키지를 다운로드하게 하는 단계.
- [0029] 예를 들어, 압축된 설치 패키지가 APP 스토어에 제공되고, 사용자는 압축된 설치 패키지를 다운로드할 수 있다.
- [0030] S12. 사용자가 APP를 실행하기 전에 설치 패키지를 다운로드하는 단계.
- [0031] 예를 들어, 사용자는 APP 스토어로부터 APP의 설치 패키지를 다운로드할 수 있고, 설치 패키지는 클래스 디스크립션 파일을 포함한다.
- [0032] 본 개시내용을 기존의 기술과 한층 더 구별하기 위해, 도 3은 기존의 기술에 따른 APP 실행 방법의 개략 흐름도

를 보여준다. 도 3을 참조하면, 기존의 기술에서는, 설치 패키지가 두 단계로, 즉 소스 파일을 컴파일하여 중간 파일(.o 파일)을 생성하는 단계(S31)와, 이어서 중간 파일의 링크를 패키징하여 설치 패키지를 생성하는 단계(S32)로 생성된다. 예를 들어, iOS 시스템에서는, 설치 패키지가 ipa 패키지로써 칭해질 수 있다. 컴파일링 및 링크 패키징 후에, ipa 패키지는 APP에 대응하는 클래스를 포함한다. 그후에, 사용자가 다운로드할 수 있도록 ipa 패키지는 APP 스토어에 배치될 수 있다. 다운로드 후에, ipa 패키지는 단말기 디바이스의 디스크에 저장된다(S33). APP가 실행될 때에, ipa 패키지는 실행을 위해 디스크로부터 단말기 디바이스의 메모리로 판독된다(S34). 이에, APP의 정상 실행(normal running)이 구현된다. ipa 패키지는 일반적으로 대형이기 때문에, 다운로드 및 저장에 대량의 리소스가 필요하다.

- [0033] 기존의 기술과 달리, 본 구현예에서는 오리지널 파일의 컴파일링 및 링크 패키징이 필요하지 않고, 대신에 클래스 디스크립션 파일이 사용된다. 클래스 디스크립션 파일은 텍스트 파일이다. 기존의 ipa 패키지와 비교하여, 클래스 디스크립션 파일의 사이즈가 작기 때문에, 다운로드 및 저장에 필요한 리소스를 줄일 수 있다.
- [0034] S13. APP가 실행될 때에, 설치 패키지로부터 클래스 디스크립션 파일을 취득하여, 클래스 디스크립션 파일에 기초해서 클래스를 생성하는 단계.
- [0035] 다운로드된 설치 패키지가 압축된 설치 패키지인 경우, 설치 패키지가 사용될 때에, 클래스 디스크립션 파일 등의 정보를 취득하기 위해 먼저 설치 패키지가 압축해제될 수 있다. 다운로드된 설치 패키지가 압축되지 않은 경우, 클래스 디스크립션 파일 등의 정보는 설치 패키지로부터 직접 취득될 수 있다.
- [0036] APP의 정상 실행을 보장하려면, APP에 포함된 클래스가 생성되어야 한다. 기존 기술의 경우, 소스 파일을 컴파일링하고 패키징함으로써 클래스가 생성될 수 있고, 그래서 디스크에 저장된 ipa 패키지가 클래스를 포함한다. APP가 실행될 때에, 클래스가 디스크로부터 직접 취득되어 APP를 정상으로 실행할 수 있다.
- [0037] 본 구현예에서는, 설치 패키지의 사이즈를 줄이기 위해, 설치 패키지가 클래스 대신에 클래스 디스크립션 파일을 포함한다. 이에, APP의 정상 실행을 보장하려면, 클래스 디스크립션 파일에 기초하여 클래스가 먼저 생성되어야 한다.
- [0038] 예를 들어, 설치 패키지를 다운로드한 후에, 사용자는 설치 패키지를 단말기 디바이스의 디스크에 저장할 수 있다. APP가 실행될 때에, 설치 패키지 내의 클래스 디스크립션 파일이 디스크로부터 메모리로 판독될 수 있는데, 이 클래스 디스크립션 파일에 기초하여 클래스가 생성된다.
- [0039] 일부 구현예에 있어서, 도 4를 참조하면, 클래스 디스크립션 파일에 기초하여 클래스를 생성하는 프로시저는 다음의 단계들을 포함할 수 있다.
- [0040] S41. 클래스 디스크립션 파일에 기초하여, 클래스명, 패런트(parent) 클래스명, 및 속성을 취득하는 단계.
- [0041] 예를 들어, 클래스 디스크립션 파일은 생성 대기중인 클래스의 클래스명 및 속성 디스크립션을 포함한다. 예를 들어, 클래스명(className)이 캐서(MERCHANTAPPOperator)이고, 속성 디스크립션은 캐서의 캐서 ID(operatorId), 캐서 명칭(operatorName), 샵 명칭(shopName) 등을 포함할 수 있다. 따라서, 클래스명과 속성 디스크립션은 클래스 디스크립션 파일로부터 취득될 수 있다.
- [0042] 속성 디스크립션이 취득된 후에, 속성 디스크립션에 기초하여 속성이 생성될 수 있다. 예를 들어, 속성 디스크립션에 기초하여 속성 변수(ivar), 접근자(프로퍼티), 및 포인터(IMP)가 직접 생성될 수 있다.
- [0043] 클래스 디스크립션 파일이 클래스명의 대응하는 패런트 클래스명을 포함하면, 클래스 디스크립션 파일로부터 패런트 클래스명이 직접 취득될 수 있다. 클래스 디스크립션 파일이 클래스명의 대응하는 패런트 클래스명을 포함하지 않으면, 패런트 클래스명을 취득하기 위해 패런트 클래스명 생성 규칙 및 클래스명에 기초하여 패런트 클래스명이 생성되어 반환될 수 있다.
- [0044] S42. 클래스명, 패런트 클래스명, 및 속성에 기초하여 그리고 iOS 런타임 기술이 제공하는 메소드를 이용하여 클래스를 생성하는 단계.
- [0045] iOS 런타임 기술은 클래스를 생성하기 위한 복수의 메소드를 제공한다. 예를 들어, 한 메소드가 objc\_allocateClassPair(Class superclass, const char \*name, size\_textraBytes)이다.
- [0046] 노트: 제1 파라미터 superclass는 패런트 클래스의 클래스 오브젝트이고, 제2 파라미터명은 클래스명이며, 제3 파라미터는 추가 할당 바이트로서 일반적으로 0이다.

- [0047] 패런트 클래스명과 클래스명이 메소드의 파라미터에 사용된다. 또한, 패런트 클래스명, 클래스명, 및 속성에 기초하여 그리고 전술한 메소드를 이용하여 클래스를 생성하는 데에, 속성도 메소드가 수행될 때에, 사용된다.
- [0048] 본 구현예에서는, 설치 패키지가 클래스 디스크립션 파일을 포함하고, 클래스 디스크립션 파일의 사이즈는 클래스의 사이즈보다 작다. 따라서, 설치 패키지에 클래스를 직접 포함시키는 기존의 방법과 비교할 때에, 설치 패키지에 클래스 디스크립션 파일을 포함시키면 APP 설치 패키지의 사이즈를 줄이고, 리소스 오버헤드를 감소시킬 수 있다.
- [0049] 도 5는 본 개시내용의 구현예에 따른 APP 실행 장치의 개략 구조도이다. 장치(50)는 생성 모듈(51), 다운로드 모듈(52), 및 실행 모듈(53)을 포함한다.
- [0050] 생성 모듈(51)은, APP의 오리지널 파일에 기초해서 클래스 디스크립션 파일을 생성하여, 설치 패키지에 클래스 디스크립션 파일을 배치하도록 구성되며, 클래스 디스크립션 파일은 클래스를 기술하는데 사용되는 텍스트 파일이고, 상기 클래스 디스크립션 파일의 사이즈는 대응하는 클래스의 사이즈보다 작다.
- [0051] 예를 들어, 오리지널 파일을 헤더 파일(.h 파일)과 소스 파일(.m 파일)을 포함한다.
- [0052] 클래스 디스크립션 파일은 텍스트 파일로서 클래스의 관련 정보를 기술하는데 사용된다. 클래스는 컴퓨터 프로그래밍의 기본 정보이다. 클래스 디스크립션 파일의 사이즈는 대응하는 클래스의 사이즈보다 작다.
- [0053] 예를 들어, 클래스명(className)이 캐서(MERCHANTAPPOperator)이고, 캐서는 클래스 디스크립션(classDescription)을 갖는다. 예를 들어, 클래스 디스크립션은 속성 디스크립션을 포함하는데, 예컨대 캐서의 캐서 ID(operatorId), 캐서 명칭(operatorName), 샵 명칭(shopName) 등을 포함할 수 있다.
- [0054] 선택사항으로서, 생성 모듈은 APP의 오리지널 파일에 기초하여 클래스 디스크립션 파일을 생성하도록 구성되는데, 클래스 디스크립션 파일의 생성은, 미리 결정된 스크립트에 기초하여 포맷 변환을 수행하는 것과, APP의 오리지널 파일을 클래스 디스크립션 파일로 변환하는 것을 포함한다.
- [0055] 예를 들어, iOS 시스템에서는, 사전 컴파일된 스크립트가 오리지널 파일을 클래스 디스크립션 파일(.plist 파일)로 변환하는 데에 이용될 수 있다. 클래스 디스크립션 파일에 필요한 콘텐츠를 오리지널 파일에서 찾을 수 있지만, 오리지널 파일과 클래스 디스크립션 파일은 상이한 포맷을 갖는다. 이에, 오리지널 파일과 클래스 디스크립션 파일 간의 변환을 구현하기 위해 스크립트가 사전 컴파일될 수 있다.
- [0056] 또한, 클래스 디스크립션 파일은 생성된 후에, 미리 결정된 디렉토리, 예컨대 리소스 번들에 저장될 수 있다.
- [0057] 클래스 디스크립션 파일은 생성된 후에, 사용자가 다운로드하여 설치할 수 있도록 설치 패키지에 배치될 수 있다.
- [0058] 일부 구현예에 있어서, 도 6을 참조하면, 생성 모듈(51)은 제1 유닛(511), 제2 유닛(512), 및 제3 유닛(513)을 포함한다.
- [0059] 제1 유닛(511)은 헤더 파일과 소스 파일에 기초하여 클래스 디스크립션 파일을 생성하도록 구성된다.
- [0060] 생성 프로시저에 대해서는, 전술한 설명을 참조하면 된다. 여기서는 다시 상세하게 설명하지 않는다.
- [0061] 제2 유닛(512)은 소스 파일을 삭제하도록 구성된다.
- [0062] 제3 유닛(513)은 소스 파일이 삭제된 후에, 설치 패키지에 헤더 파일과 클래스 디스크립션 파일을 배치하여 사용자가 다운로드하게 하도록 구성된다.
- [0063] 클래스 디스크립션 파일과 헤더 파일 둘 다 작고, 취득된 설치 패키지도 작기 때문에, 사용자는 APP 스토어에서 설치 패키지를 직접 다운로드할 수 있다.
- [0064] 선택사항으로서, 생성 모듈(51)은 설치 패키지를 압축하여 사용자가 압축된 설치 패키지를 다운로드하게 하도록 구성된 제4 유닛(514)을 더 포함한다.
- [0065] 설치 패키지의 사이즈를 더욱 줄이기 위해, 취득되는 설치 패키지는 또한 압축될 수 있다.
- [0066] 예를 들어, 압축된 설치 패키지가 APP 스토어에 제공되고, 사용자는 압축된 설치 패키지를 다운로드할 수 있다.
- [0067] 다운로드 모듈(52)은 사용자가 APP를 실행하기 전에 설치 패키지를 다운로드하도록 구성된다.
- [0068] 예를 들어, 사용자는 APP 스토어로부터 APP의 설치 패키지를 다운로드할 수 있고, 설치 패키지는 클래스 디스크

립션 파일을 포함한다.

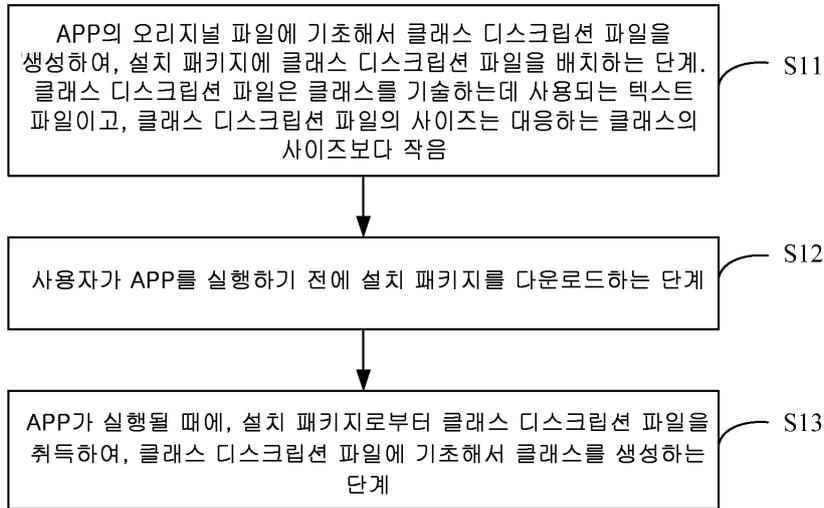
- [0069] 기존의 기술과 달리, 본 구현예에서는 오리지널 파일의 컴파일링 및 패키징이 필요하지 않지만, 대신에 클래스 디스크립션 파일이 사용된다. 클래스 디스크립션 파일은 텍스트 파일이다. 기존의 ipa 패키지와 비교하여, 클래스 디스크립션 파일의 사이즈가 작기 때문에, 다운로드 및 저장에 필요한 리소스를 줄일 수 있다.
- [0070] 실행 모듈(53)은 APP가 실행될 때에, 설치 패키지로부터 클래스 디스크립션 파일을 취득하여, 클래스 디스크립션 파일에 기초해서 클래스를 생성하도록 구성된다.
- [0071] 다운로드된 설치 패키지가 압축된 설치 패키지인 경우, 설치 패키지가 사용될 때에, 클래스 디스크립션 파일 등의 정보를 취득하기 위해 먼저 설치 패키지가 압축해제될 수 있다. 다운로드된 설치 패키지가 압축되지 않은 경우, 클래스 디스크립션 파일 등의 정보는 설치 패키지로부터 직접 취득될 수 있다.
- [0072] APP의 정상 실행을 보장하려면, APP에 포함된 클래스가 생성되어야 한다. 기존 기술의 경우, 소스 파일을 컴파일링하고 패키징함으로써 클래스가 생성될 수 있고, 그래서 디스크에 저장된 ipa 패키지가 클래스를 포함한다. APP가 실행될 때에, 클래스가 디스크로부터 직접 취득되어 APP의 정상 실행을 구현할 수 있다.
- [0073] 본 구현예에서는, 설치 패키지의 사이즈를 줄이기 위해, 설치 패키지가 클래스 대신에 클래스 디스크립션 파일을 포함한다. 이에, APP의 정상 실행을 보장하려면, 클래스 디스크립션 파일에 기초하여 먼저 클래스가 생성되어야 한다.
- [0074] 예를 들어, 설치 패키지를 다운로드한 후에, 사용자는 설치 패키지를 단말기 디바이스의 디스크에 저장할 수 있다. APP가 실행될 때에, 설치 패키지 내의 클래스 디스크립션 파일이 디스크로부터 메모리로 관독될 수 있는데, 이 클래스 디스크립션 파일에 기초하여 클래스가 생성된다.
- [0075] 일부 구현예에 있어서, 실행 모듈(53)은 클래스 디스크립션 파일에 기초하여 클래스를 생성하도록 구성되며, 클래스의 생성은, 클래스 디스크립션 파일에 기초하여 파일명, 패러미터 클래스명, 및 속성을 취득하는 것과, 클래스명, 패러미터 클래스명 및 속성에 기초하여 그리고 iOS 런타임 기술이 제공하는 메소드를 이용하여 클래스를 생성하는 것을 포함한다.
- [0076] 예를 들어, 클래스 디스크립션 파일은 생성 대기중인 클래스의 클래스명 및 속성 디스크립션을 포함한다. 예를 들어, 클래스명(className)이 캐셔(MERCHANTAPPOperator)이고, 속성 디스크립션은 캐셔의 캐셔 ID(operatorId), 캐셔 명칭(operatorName), 샵 명칭(shopName) 등을 포함할 수 있다. 따라서, 클래스명과 속성 디스크립션은 클래스 디스크립션 파일로부터 취득될 수 있다.
- [0077] 속성 디스크립션이 취득된 후에, 속성 디스크립션에 기초하여 속성이 생성될 수 있다. 예를 들어, 속성 디스크립션에 기초하여 속성 변수(ivar), 접근자(프로퍼티), 및 포인터(IMP)가 직접 생성될 수 있다.
- [0078] 클래스 디스크립션 파일이 클래스명의 대응하는 패러미터 클래스명을 포함하면, 클래스 디스크립션 파일로부터 패러미터 클래스명이 직접 취득될 수 있다. 클래스 디스크립션 파일이 클래스명의 대응하는 패러미터 클래스명을 포함하지 않으면, 패러미터 클래스명을 취득하기 위해 패러미터 클래스명 생성 규칙 및 클래스명에 기초하여 패러미터 클래스명이 생성되어 반환될 수 있다.
- [0079] iOS 런타임 기술은 클래스를 생성하기 위한 복수의 방법을 제공한다. 예를 들어, 한 메소드가 objc\_allocateClassPair(Class superclass, const char \*name, size\_t extraBytes)이다.
- [0080] 노트: 제1 파라미터 superclass는 패러미터 클래스의 클래스 오브젝트이고, 제2 파라미터명은 클래스명이며, 제3 파라미터는 추가 할당 바이트로서 일반적으로 0이다.
- [0081] 패러미터 클래스명과 클래스명이 메소드의 파라미터에 사용된다. 또한, 패러미터 클래스명, 클래스명, 및 속성에 기초하여 그리고 전술한 메소드를 이용하여 클래스를 생성하는 데에, 속성도 메소드가 수행될 때에, 사용된다.
- [0082] 본 구현예에서는, 설치 패키지가 클래스 디스크립션 파일을 포함하고, 클래스 디스크립션 파일의 사이즈는 클래스의 사이즈보다 작다. 따라서, 설치 패키지에 클래스를 직접 포함시키는 기존의 방법과 비교할 때에, 설치 패키지에 클래스 디스크립션 파일을 포함시키면 APP 설치 패키지의 사이즈를 줄이고, 리소스 오버헤드를 감소시킬 수 있다.
- [0083] 본 개시내용의 설명에서, "제1" 및 "제2" 같은 용어는 설명의 용도로 의도된 것일 뿐이며, 상대적 중요성을 나타내는 표시 또는 의미로 이해되어서는 안 된다는 점을 유의해야 한다. 또한, 본 개시내용의 설명에서는, 달리

언급되지 않는다면 "복수"는 "적어도 2개"를 의미한다.

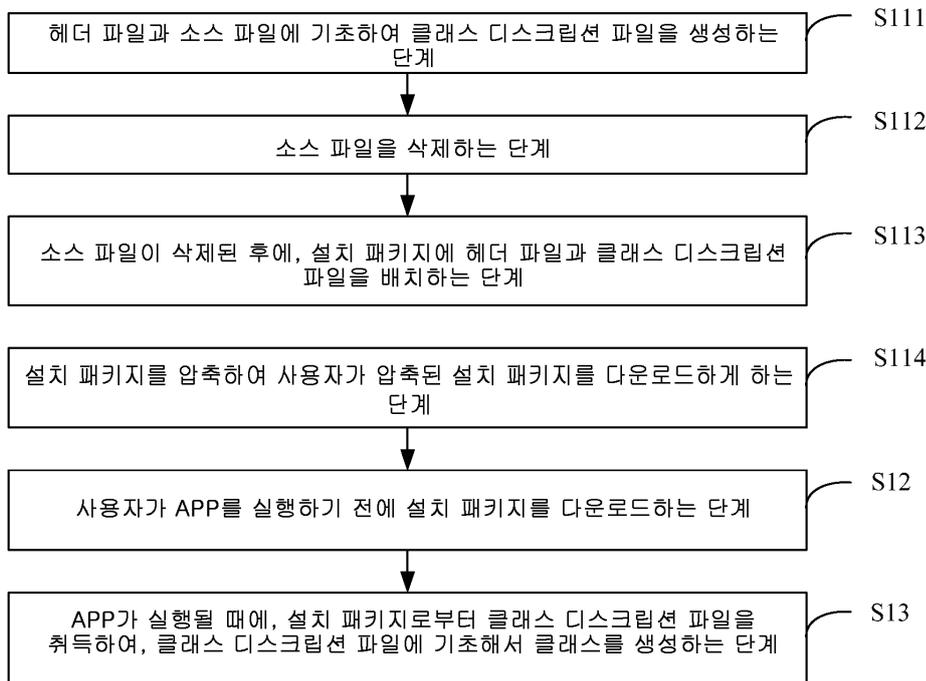
- [0084] 여기에서 흐름도로 또는 다른 방식으로 기술하는 임의의 프로세스 또는 방법에 대한 설명은 다음과 같이 이해될 수 있는데, 특정 논리적 기능 또는 동작 단계를 구현하는데 사용되는 실행 가능한 명령어의 코드의 하나 이상의 모듈, 세그먼트 또는 부분이 포함된다는 것과, 본 개시내용의 바람직한 구현예의 범위가 다른 구현예를 포함하는 것과, 관련된 기능에 따라 기본적으로 동시적인 방식 또는 역순을 포함해, 도시하거나 논리적인 순서 이외의 순서로 기능이 수행될 수 있다는 것이다. 이것은 본 개시내용의 구현 분야의 당업자라면 이해되어야 한다.
- [0085] 본 개시내용 중의 부분들은 하드웨어, 소프트웨어, 펌웨어, 또는 이들의 조합을 사용하여 구현될 수 있음을 이해해야 한다. 전술한 구현예에서는, 메모리에 저장되어 적절한 명령어 실행 시스템에 의해 실행되는 소프트웨어 또는 펌웨어를 사용하여 복수의 단계 또는 방법이 구현될 수 있다. 예를 들어, 하드웨어가 또 다른 구현예와 유사한 구현예에 사용되는 경우, 당업계에 공지된 다음과 같은 기술, 즉 데이터 신호에 대한 논리적 기능을 구현하는데 사용되는 로직 게이트 회로를 구비한 이산 로직 회로, 적절한 조합 로직 회로를 구비한 APP 특유의 집적 회로, 프로그래머블 게이트 어레이(PGA), 필드 프로그래머블 게이트 어레이(FPGA) 등등 중 임의의 하나 또는 조합도 구현에 사용될 수 있다.
- [0086] 당업자라면, 관련 하드웨어에 지시하는 컴퓨터 구현 프로그램에 의해 방법 구현의 단계들의 일부 또는 전부가 구현될 수 있음을 이해할 수 있다. 이 프로그램은 컴퓨터 판독 가능한 저장 매체에 저장될 수 있다. 프로그램이 실행될 때에, 방법 구현의 단계들 중 하나 또는 조합이 수행된다.
- [0087] 또한, 본 개시내용의 구현예에서의 기능 유닛들이 하나의 프로세싱 모듈에 통합될 수 있거나, 각각의 유닛이 물리적으로 단독으로 존재할 수도 있고, 2 이상의 유닛이 하나의 모듈로 통합될 수도 있다. 통합 모듈은 하드웨어의 형태로 구현될 수 있거나 소프트웨어 기능 모듈의 형태로 구현될 수 있다. 통합 모듈이 소프트웨어 기능 모듈의 형태로 구현되고 독립적인 제품으로서 판매 또는 사용될 경우, 통합 모듈은 컴퓨터 판독 가능한 저장 매체에 저장될 수 있다.
- [0088] 전술한 저장 매체는 판독 전용 메모리, 자기 디스크 또는 광 디스크일 수 있다.
- [0089] 본 명세서의 설명에서, "구현예", "일부 구현예", "예", "특정 예" 및 "일부 예"와 같은 참조 용어에 대한 설명은, 구현예 또는 예를 참조하여 설명한 특정 특징, 구조, 재료, 또는 특성이 본 개시내용의 적어도 하나의 구현예 또는 예에 포함되는 것을 의미한다. 본 개시내용에 있어서, 용어에 대한 전술한 예시적인 표현은 반드시 동일한 구현예 또는 예에 관한 것은 아니다. 또한, 설명한 특정 특징, 구조, 재료 또는 특성은 임의의 하나 이상의 구현예 또는 예에서 적절한 방식으로 조합될 수 있다.
- [0090] 본 개시내용의 구현예들을 도시하여 전술하였지만, 전술한 구현예들은 예시일 뿐이며, 본 개시내용에 대한 제한으로서 해석될 수 없음이 이해될 수 있다. 본 개시내용의 범위 내에서, 당업자라면 전술한 구현예들에 대해 변경, 수정, 대체 및 변형을 행할 수 있다.

도면

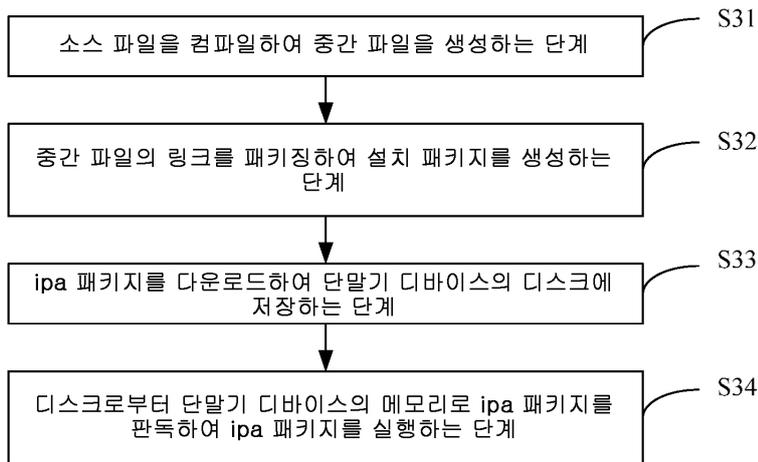
도면1



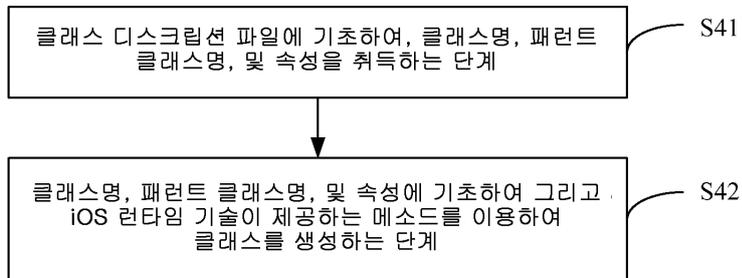
도면2



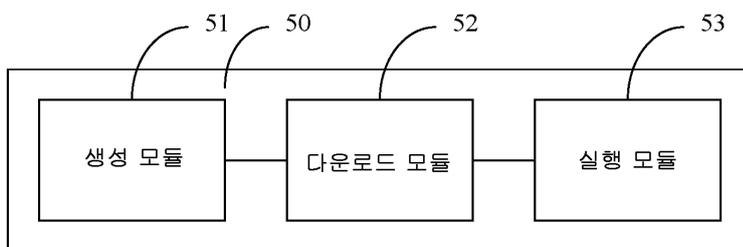
도면3



도면4



도면5



도면6

