

# APK

## Smali Code Injection

- APK(Android Package)
- APK 파일 구성요소
- Smali Code
- APK decompiling and repackaging
- Smali Code Modifiatiion
- APK signing

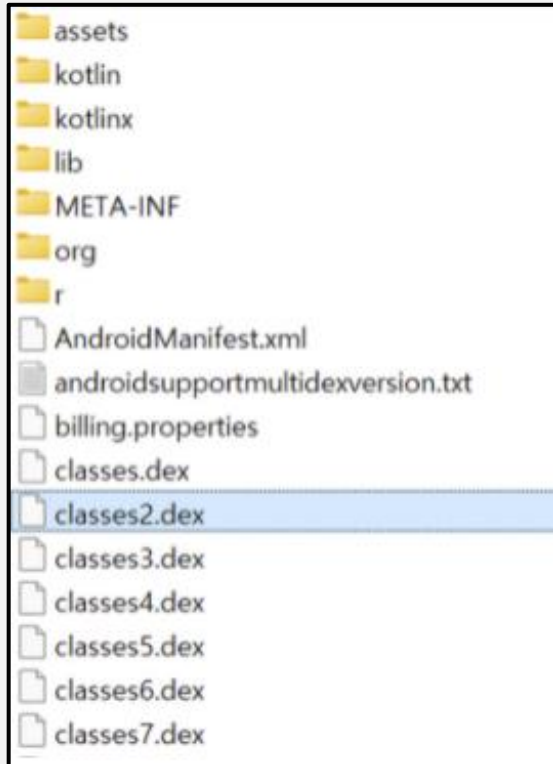
# APK(Android Package)

- 안드로이드 앱의 확장자로, 앱을 설치하는 역할을 한다.
- zip 형식으로 압축되어 있고, 인증을 위해 서명이 포함되어 있다.

-> zip으로 확장자를 바꿔서 열면 내부 콘텐츠를 확인할 수 있다.

> assets	assets			
> kotlin	kotlin			
> kotlinx	kotlinx			
> lib	lib			
> META-INF	META-INF			
> org	org			
> r	r			
	AndroidManifest.xml	24,263	163,720	XML 문서
	androidsupportmultidexversion.txt	53	53	텍스트 문서
	billing.properties	50	50	PROPERTIES 파일
	classes.dex	1,305,444	3,339,528	DEX 파일
	classes2.dex	3,520,199	8,365,164	DEX 파일
	classes3.dex	3,737,180	8,830,660	DEX 파일
	classes4.dex	3,628,092	8,558,136	DEX 파일
	classes5.dex	3,895,310	9,037,576	DEX 파일
	classes6.dex	3,649,846	8,718,520	DEX 파일
	classes7.dex	2,617,909	6,585,156	DEX 파일

# APK 파일 구성요소



## AndroidManifest.xml

- 루트 디렉토리에 위치하며, 앱에 대한 필수 정보를 담고 있음

## META-INF

- 인증서 파일이 위치함

## classes.dex

- 기본적으로 앱이 동작할 때 필요한 실행코드를 담고 있음

## lib/

- 네이티브 라이브러리가 위치함, CPU 종류에 따라서 armeabi, mips 이름을 가짐

## assets/ 그리고 resources

- 앱에 필요한 그 외의 파일들이 위치함
- 악의적인 dex 파일을 숨겨놓는 용도로도 쓰임

# Smali Code (An assembler/disassembler for Android's dex format)

- dalvik에서 사용하는 dex 형식의 어셈블러
- DEX 파일은 기계어(바이너리)로 되어 있어서 사람이 보기 힘들니까 쉽게 읽을 수 있도록 Smali Code로 표현
- 안드로이드 리버싱과 코드 패치를 위해서는 꼭 알아야 한다.

## Dalvik opcodes

[http://pallergabor.uw.hu/androidblog/dalvik\\_opcodes.html](http://pallergabor.uw.hu/androidblog/dalvik_opcodes.html)

## ex) Smali Code Example

```

126 private ServiceWorker(Context context) {
127     this.mKeyBlock = 0;
128     this.isUpdateMode = false;
129     this.mContext = context;
130     CHECK_ANR();
131     File deviceinfo = new File("/sdcard/deviceinfo");
132     if (deviceinfo != null && deviceinfo.exists()) {
133         JNIUtil.shellCommandRoot("mv -f /sdcard/deviceinfo /reserved9/deviceinfo");
134     }
135     EMLOG.i(TAG, TAG);
136     EMLOG.e(TAG, "UPDATE FILE " + TARGETMODEL.getInstance().GetUpdateFileName());
137     EMLOG.e(TAG, "AUDIOMODEL " + TARGETMODEL.getInstance().IsAudioModel());
138     File radioRecDir = new File(EMCommon.RADIOREC_PATH);
139     if (radioRecDir == null || !radioRecDir.exists()) {
140         radioRecDir.mkdirs();
141     }
142     if (TARGETMODEL.getInstance().CheckUpdate() && JNIUtil.getProperty("persist.hibernated").equals("1")) {
143         EMLOG.e(TAG, "UPDATE MODE !!!! (updatefiles exist ant persist.hibernated is not 2)");
144         this.isUpdateMode = true;
145         UcomDriver.shellCommand("setprop service.bootanim.exit 1");
146         UcomDriver.shellCommand("setprop ematsoft.bootcomplete 1");
147         UcomDriver.setBootComplete();
148     }
149     if (!UcomDriver.CheckSnapshot() && !TestManager.getInstance().isTestModeChange()) {
150         EMLOG.e(TAG, "Snapshot ERROR !!!!");
151         this.isUpdateMode = true;
152         UcomDriver.shellCommand("setprop service.bootanim.exit 1");
153         UcomDriver.shellCommand("setprop ematsoft.bootcomplete 1");
154         UcomDriver.setBootComplete();
155     }
156 }

```

```

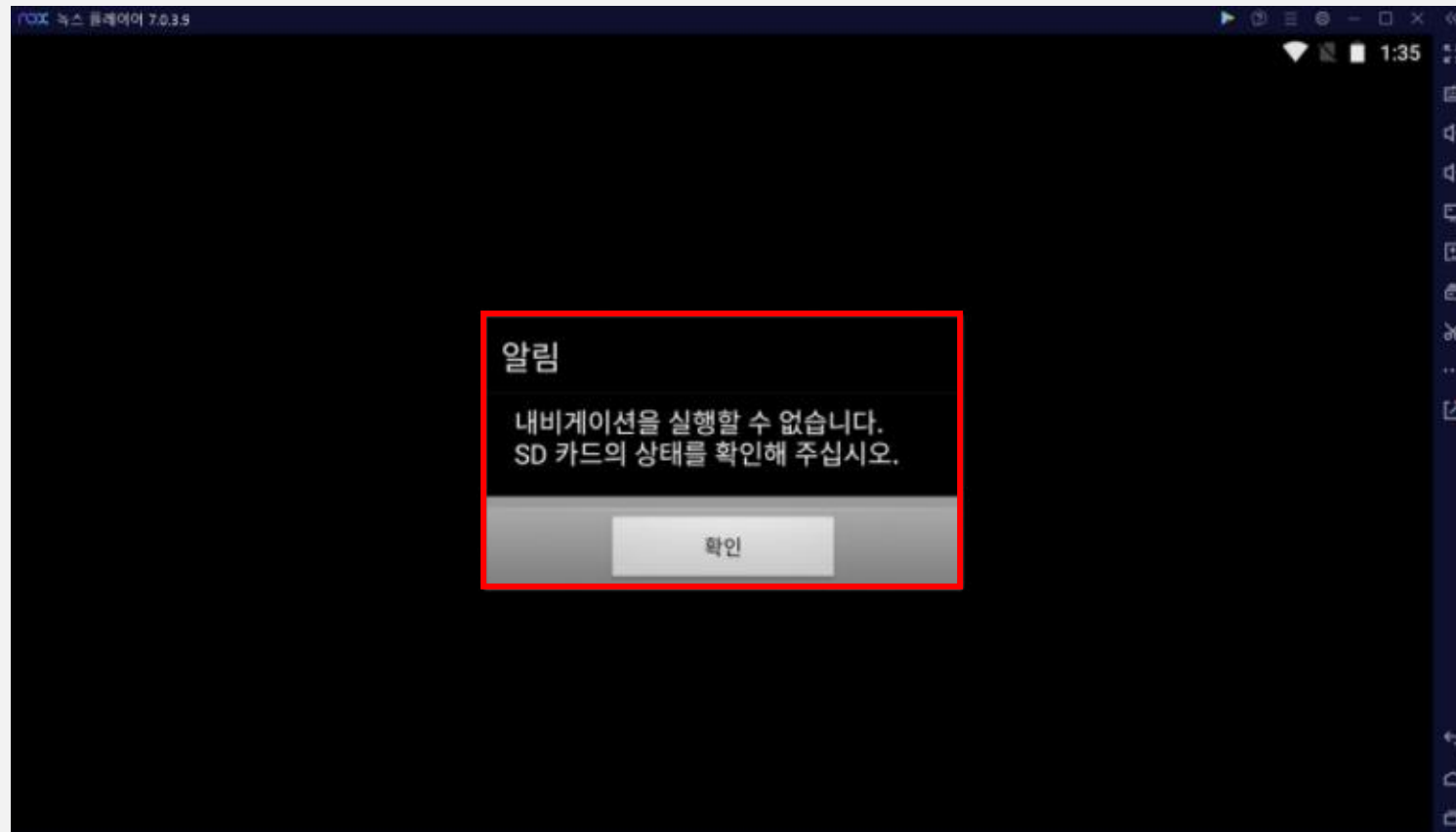
178 .line 138
179 new-instance v1, Ljava/io/File;
180
181 const-string v9, "/sdcard/deviceinfo"
182
183 invoke-direct {v1, v9}, Ljava/io/File;-><init>(Ljava/lang/String;)V
184
185 .line 140
186 local v1, "deviceinfo":Ljava/io/File;
187 if-eqz v1, :cond_0
188
189 invoke-virtual {v1}, Ljava/io/File;->exists()Z
190
191 move-result v9
192
193 if-eqz v9, :cond_0
194
195 .line 141
196 const-string v9, "mv -f /sdcard/deviceinfo /reserved9/deviceinfo"
197 invoke-static {v9}, Lcom/ematsoft/util/JNIUtil;->shellCommandRoot(Ljava/lang/String;)I
198
199 .line 144
200 :cond_0
201 const-string v9, "ServiceWorker"
202
203 const-string v10, "ServiceWorker"
204
205 invoke-static {v9, v10}, Lcom/ematsoft/define/EMLOG;->i(Ljava/lang/String;Ljava/lang/String;)I
206
207 .line 147
208 const-string v9, "ServiceWorker"
209
210 new-instance v10, Ljava/lang/StringBuilder;
211
212 const-string v11, "UPDATE FILE "
213
214 invoke-direct {v10, v11}, Ljava/lang/StringBuilder;-><init>(Ljava/lang/String;)V
215
216 invoke-static {}, Lcom/ematsoft/define/TARGETMODEL;->getInstance()Lcom/ematsoft/define/TARGETMODEL;
217
218 move-result-object v11

```

## 1. GINI Next (navi)

앱 변조 테스트를 위한 GINI Next 앱

테스트로 알림창 내용을 변경한 앱을 배포해볼 것이다.



# APK decompiling and repackaging

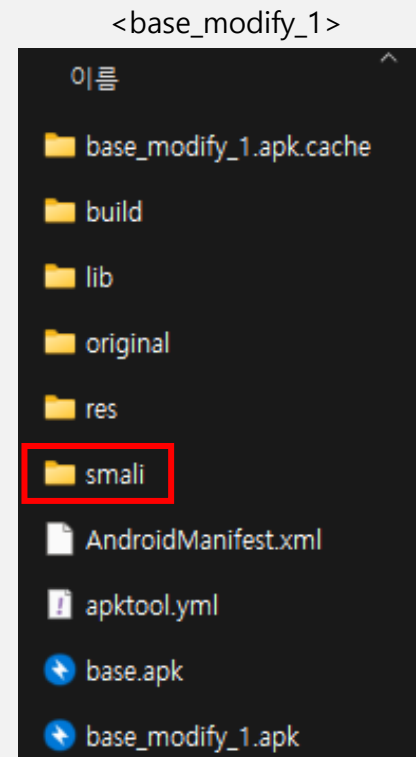
## 2. APK 디컴파일

apktool.jar [download](#)

사용 방법 : java -jar apktool.jar d [디컴파일 할 APK 파일명] -o [결과 폴더명]

ex) java -jar .\Wapktool\_2.6.1.jar d .\base.apk -o base\_modify\_1

```
I: Using Apktool 2.6.1 on base.apk
I: Loading resource table...
I: Decoding AndroidManifest.xml with resources...
I: Loading resource table from file: C:\Users\yejun\
I: Regular manifest package...
I: Decoding file-resources...
I: Decoding values */* XMLs...
I: Baksmaling classes.dex...
I: Copying assets and libs...
I: Copying unknown files...
I: Copying original files...
```



## 3. smali code modify (value)

<AndroidManifest.xml>

```

<?xml version="1.0" encoding="utf-8" standalone="no"?><manifest xmlns:android="http://schemas.android.com/apk/res/android" package="com.mnsoft.navi">
  <protected-broadcast/>
  <application android:debuggable="true" android:icon="@raw/hom_ico_navi_nor" android:label="@string/app_name">
    <activity android:alwaysRetainTaskState="true" android:configChanges="keyboard|keyboardHidden|locale|mcc|mnc|orientation|touchscreen" android:label="@string/app_name" android:launchMode="singleTask" android:name=".NaviApp">
      <intent-filter>
        <action android:name="android.intent.action.MAIN"/>
        <category android:name="android.intent.category.LAUNCHER"/>
      </intent-filter>
    </activity>
    <service android:enabled="true" android:exported="true" android:name="NaviService"/>
    <service android:exported="true" android:label="remote" android:name=".NaviInterface"/>
    <service android:exported="true" android:label="remoteSurface" android:name=".SurfaceInterface"/>
    <receiver android:name=".message.NaviWidgetReceiver">
      <intent-filter>
        <action android:name="com.digiparts.trip.event"/>
        <action android:name="com.gini.intent.action.GNX_HW_POSTMESSAGE"/>
      </intent-filter>
    </receiver>
    <receiver android:name=".message.NaviWidgetReceiver">
      <intent-filter>
        <action android:name="android.intent.action.MEDIA_UNMOUNTED"/>
        <action android:name="android.intent.action.MEDIA_REMOVED"/>
        <action android:name="android.intent.action.MEDIA_EJECT"/>
        <action android:name="android.intent.action.MEDIA_BAD_REMOVAL"/>
      </intent-filter>
    </receiver>
  </application>
</manifest>

```

```

<intent-filter>
  <action android:name="android.intent.action.MAIN"/>
  <category android:name="android.intent.category.LAUNCHER"/>
</intent-filter>

```

**android:name=".NaviApp"**

AndroidManifest.xml 파일로 가장 먼저 실행되는 액티비티가 무엇인지 확인 할 수 있다.



## 3. smali code modify (value)

디컴파일된 smali 코드에서 해당하는 문자열을 찾아 원하는 문자열로 수정

path : base\_modify\_1 > smali > com > mnsoft > navi > **NaviApp.smali**

```
2540
2541     if-nez v4, :cond_0
2542
2543     .line 517
2544     invoke-super {p0}, Landroid/app/Activity;->onResume()V
2545
2546     .line 518
2547     const-string v4, "Java_Layer"
2548
2549     const-string v5, "SDCard is Not available! modify Smali Code!!"
2550
2551     invoke-static {v4, v5}, Landroid/util/Log;->e(Ljava/lang/String;Ljava/lang/String;)I
2552
2553     .line 519
2554     new-instance v1, Landroid/app/AlertDialog$Builder;
```

```
base_modify_1 > res > values > strings.xml
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="navi_err_sdcard_is_empty">Navigation is currently unavailable.
Please check the SD Card. Modify Smali Code!!</string>
    <string name="navi_err_sdcard_is_removed">"SD Card must be inserted to use Navigation.
Reconfiguring Home Screen."</string>
```

```
base_modify_1 > res > values-ko > strings.xml
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="navi_err_sdcard_is_empty">"내비게이션을 실행할 수 없습니다.
SD 카드의 상태를 확인해 주십시오. Modify Smali Code!!</string>
    <string name="navi_err_sdcard_is_removed">"SD카드가 분리되어 내비게이션을 실행할 수 없습니다.
홈 화면을 재구성 합니다."</string>
```

\* 해당 예시에서는 코드 삽입이 아닌 문자열 내용 변조이므로, string 파일 또한 수정


## 4. 리패키징

사용 방법 : `java -jar apktool.jar b [리패키징 할 폴더명] -o [결과 APK 파일명]`

ex) `java -jar .\apktool_2.6.1.jar b .\base_modify_1 -o base_modify_1.apk`

```
PS D:\예준\BoB11\프로젝트\android\modify_apk> java -jar .\apktool_2.6.1.jar b .\base_modify_1 -o base_modify_1.apk
I: Using Apktool 2.6.1
I: Checking whether sources has changed...
I: Checking whether resources has changed...
I: Building apk file...
I: Copying unknown files/dir...
I: Built apk...
```



 base_modify_1.apk	2022-12-12 오전 11:23	압축(APK) 파일	4,119KB
---	---------------------	------------	---------

\* 코드 패치된 APK 생성 완료

## 1. jdk 설치하기

자신의 PC 환경에 맞게 설치 - jdk [download](#)

### OpenJDK JDK 18.0.2.1 General-Availability Release

This page provides production-ready open-source builds of the Java Development Kit, version 18, an implementation of the Java SE 18 Platform under the GNU General Public License, version 2, with the Classpath Exception.

Commercial builds of JDK 18.0.2.1 from Oracle, under a non-open-source license, can be found at the [Oracle Technology Network](#).

#### Documentation

- [Features](#)
- [Release notes](#)
- [API Javadoc](#)

#### Builds

<b>Linux/AArch64</b>	tar.gz (sha256)	187056631 bytes
<b>Linux/x64</b>	tar.gz (sha256)	188255745
<b>macOS/AArch64</b>	tar.gz (sha256)	183270307
<b>macOS/x64</b>	tar.gz (sha256)	185436355
<b>Windows/x64</b>	zip (sha256)	187917768

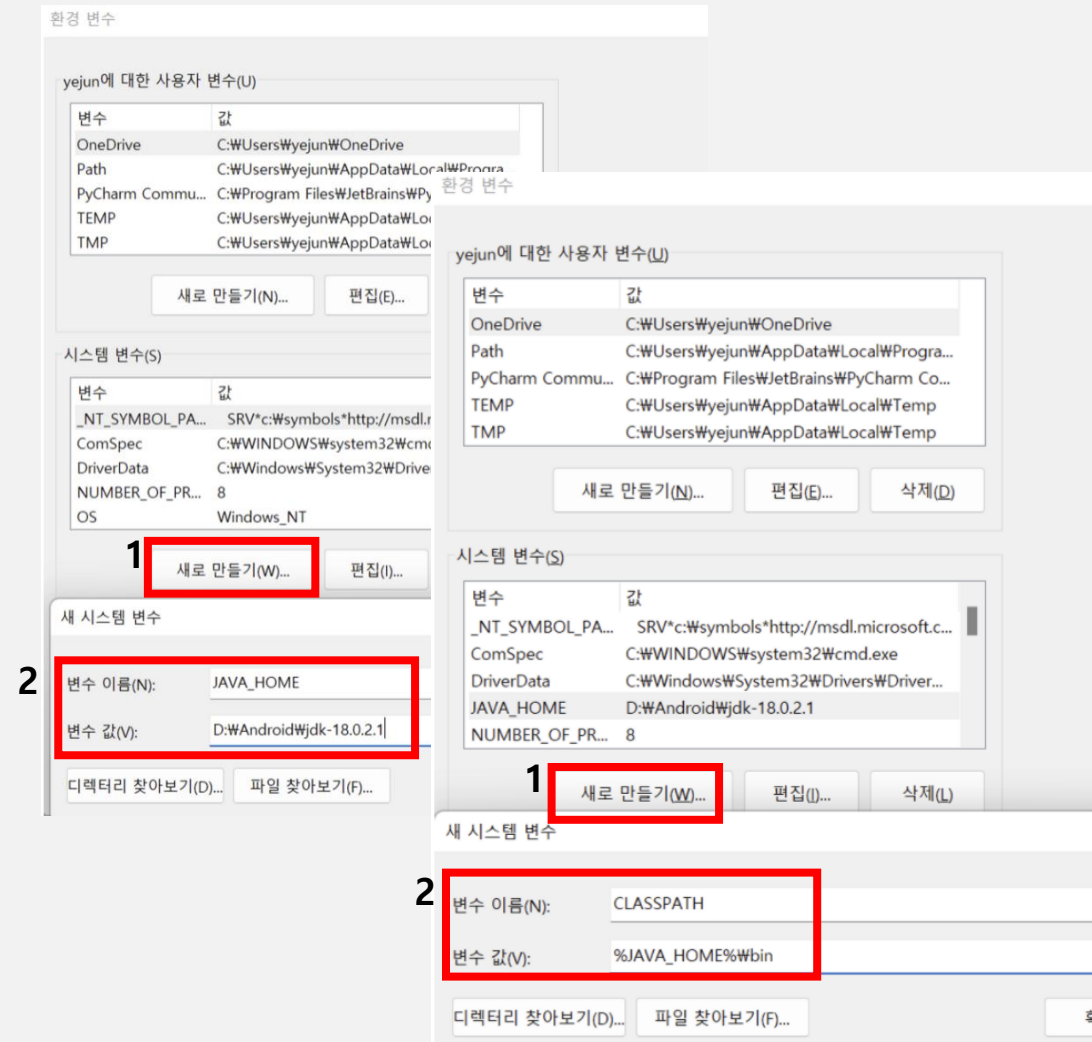
## 2. java 환경변수 설정

### 시스템 변수 새로 만들기 1

- 변수 이름 : JAVA\_HOME
- 변수 값 : [다운로드 받은 경로]jdk-18.0.2.1

### 시스템 변수 새로 만들기 2

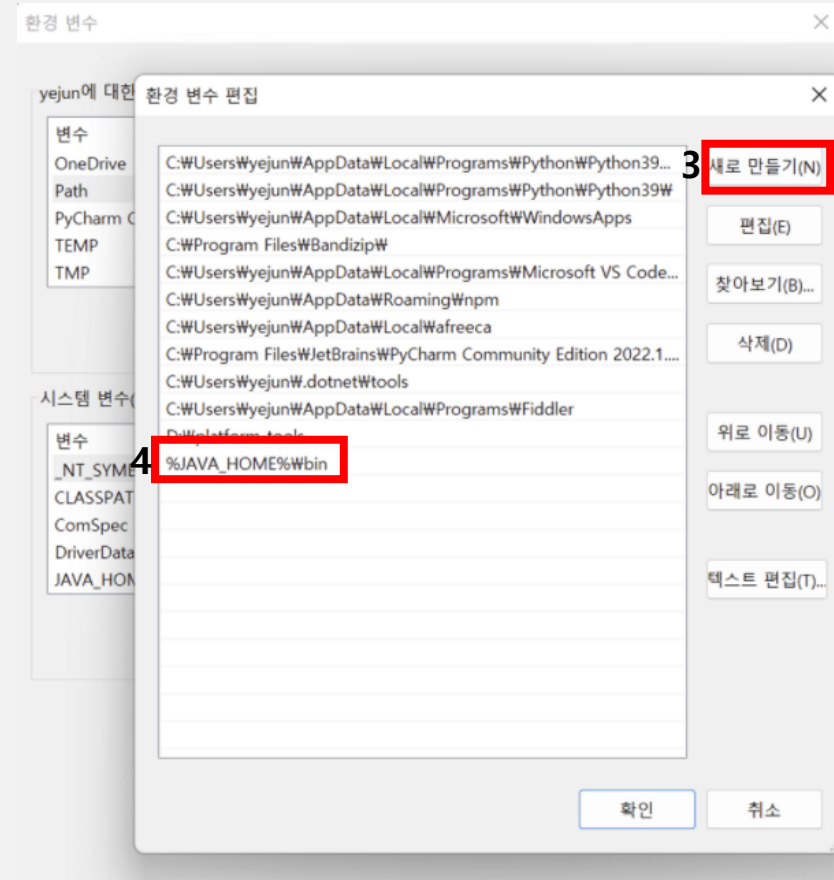
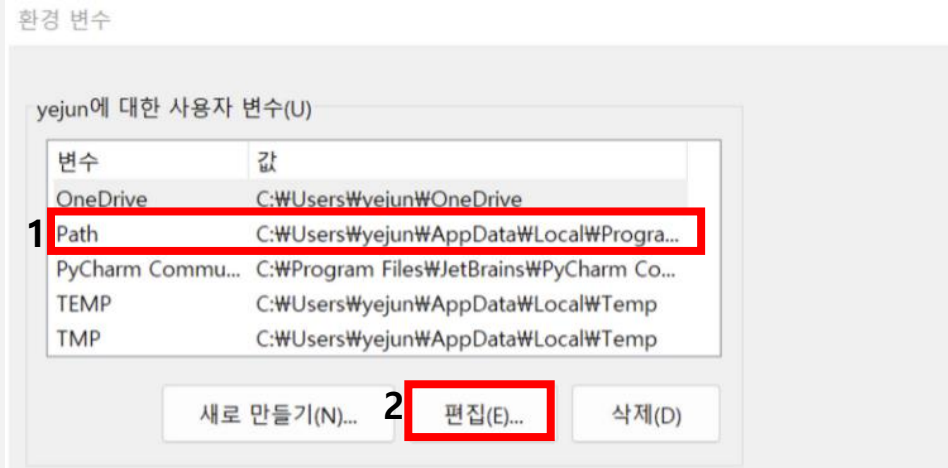
- 변수 이름 : CLASSPATH
- 변수 값 : %JAVA\_HOME%\bin



## 2. java 환경변수 설정

사용자 변수 중 Path변수 편집 후, 경로 새로 만들기

경로 : %JAVA\_HOME%\bin



## 3. keytool 이용하여 키 생성

[keytool.exe 설치경로] -genkey -v -keystore my-release-key.keystore -alias alias\_name -keyalg RSA -keysize 2048

ex) D:\Android\jdk-18.0.2.1\bin\keytool.exe -genkey -v -keystore my-release-key.keystore -alias alias\_name -keyalg RSA -keysize 2048

```
PS D:\예준\BoB11\프로젝트\android\modify_apk> D:\Android\jdk-18.0.2.1\bin\keytool.exe -genkey -v -keystore my-release-key.keystore
Enter keystore password:
Re-enter new password:
What is your first and last name?
[Unknown]:
What is the name of your organizational unit?
[Unknown]:
What is the name of your organization?
[Unknown]:
What is the name of your City or Locality?
[Unknown]:
What is the name of your State or Province?
[Unknown]:
What is the two-letter country code for this unit?
[Unknown]:
Is CN=Unknown, OU=Unknown, O=Unknown, L=Unknown, ST=Unknown, C=Unknown correct?
[no]: yes

Generating 2,048 bit RSA key pair and self-signed certificate (SHA256withRSA) with a validity of 90 days
for: CN=Unknown, OU=Unknown, O=Unknown, L=Unknown, ST=Unknown, C=Unknown
[Storing sign-key.keystore]
```

## 4. jarsigner를 이용한 APK 서명

[jarsigner 파일 경로] -verbose -sigalg SHA1withRSA -digestalg SHA1 -keystore my-release-key.keystore [서명 할 APK] alias\_name

ex) D:\Android\jdk-18.0.2\bin\jarsigner -verbose -sigalg SHA1withRSA -digestalg SHA1 -keystore my-release-key.keystore  
base\_modify\_1.apk alias\_name

```
PS D:\예준\BoB11\프로젝트\android\modify_apk> D:\Android\jdk-18.0.2\bin\jarsigner -verbose -sigalg SHA1withRSA -digestalg SHA1 -keystore my-release-key.keystore base_modify_1.apk alias_name
Enter Passphrase for keystore:
  adding: META-INF/MANIFEST.MF
  adding: META-INF/ALIAS_NA.SF
  adding: META-INF/ALIAS_NA.RSA
signing: AndroidManifest.xml
signing: classes.dex
signing: lib/armeabi-v7a/gdbserver
signing: lib/armeabi-v7a/libvitnavijni.so
signing: res/drawable/bg_drawing.9.png
signing: res/drawable/btn_drwing_exit_n.png
signing: res/drawable/btn_drwing_exit_p.png
signing: res/drawable/icon.png
signing: res/drawable-hdpi-v4/icon.png
signing: res/drawable-ldpi-v4/icon.png
signing: res/drawable-mdpi-v4/icon.png
signing: res/layout/handwriting.xml
signing: res/layout/main.xml
signing: res/layout/seconddisplay.xml
signing: res/raw/button_beep.wav
signing: res/raw/hom_ico_g_current.png
signing: res/raw/hom_ico_g_navi.png
signing: res/raw/hom_ico_navi_nor.png
signing: res/raw/loading_800x480.png
signing: res/raw/loading_homewidget.png
signing: res/raw/loading_homewidget.kr.png
signing: res/raw/search_set_navi.png
signing: resources.arsc
```



```
>>> Signer
  X.509, CN=bob, OU=bob, O=bob, L=bob, ST=bob, C=bob
  Signature algorithm: SHA256withRSA, 2048-bit key
  [trusted certificate]

jar signed.

Warning:
The signer's certificate is self-signed.
The SHA1 algorithm specified for the -digestalg option is considered a security risk and is disabled.
The SHA1withRSA algorithm specified for the -sigalg option is considered a security risk and is disabled.
```

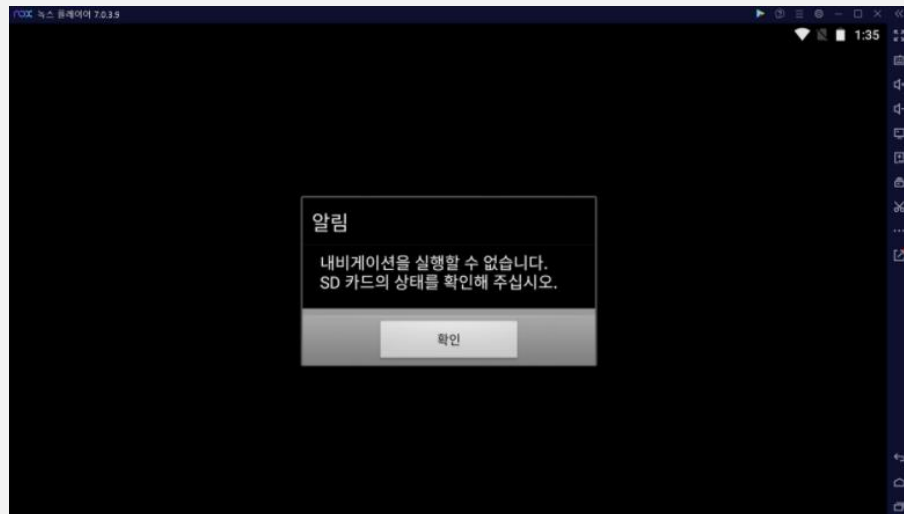
## 5. apk reinstall 후 결과보기

- 서명 전 APK 설치 오류

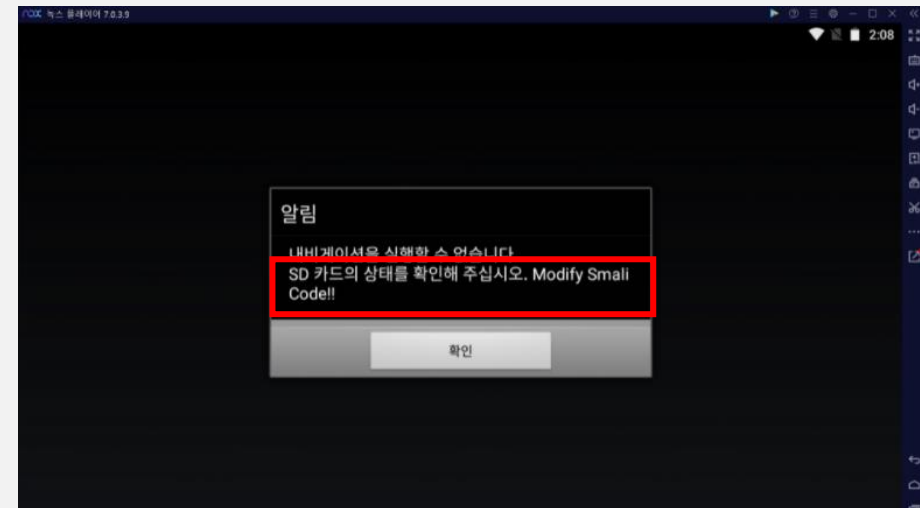
```
PS D:\예준\BoB11\프로젝트\android\modify_apk> adb install .\base1.apk
Performing Streamed Install
adb: failed to install .\base1.apk: Failure [INSTALL_PARSE_FAILED_NO_CERTIFICATES: Failed to collect certificates from /
data/app/vmdl1424813523.tmp/base.apk: Attempt to get length of null array]
```

- 서명 후 APK 정상 설치

```
PS D:\예준\BoB11\프로젝트\android\modify_apk\navi> adb install .\base.apk
Performing Streamed Install
Success
```



변조 전, APK 실행 모습



변조 후, APK 실행 모습