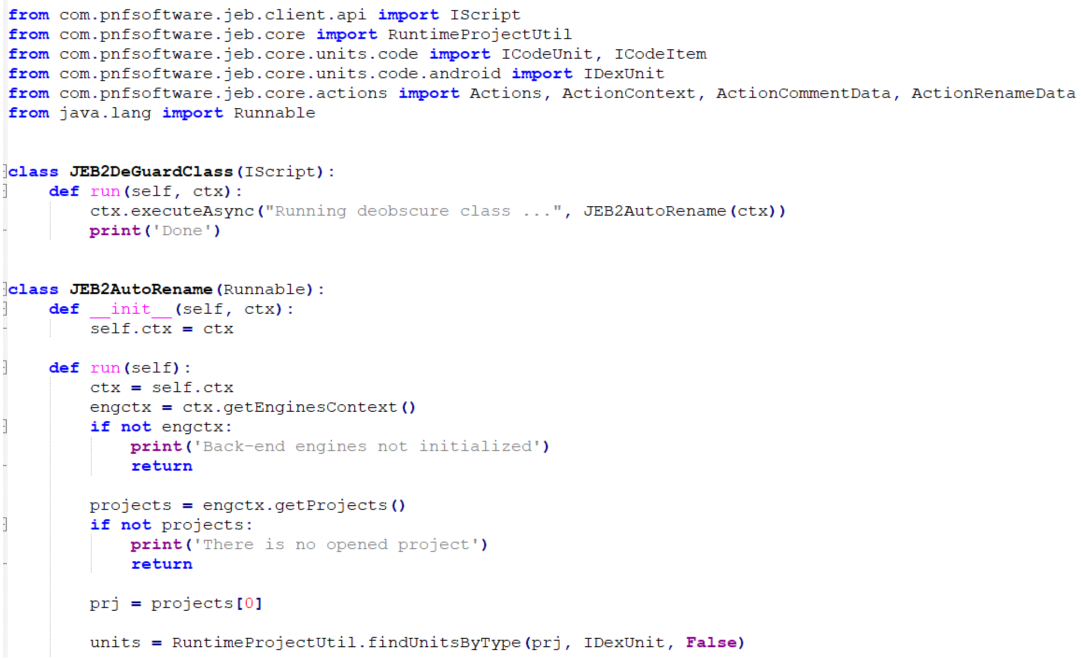
Jeb plugin/script report of PoRE

Student ID　20307130044
Name　徐俊伟

- **Task**
- **Which part of Jeb do you plan to improve?**

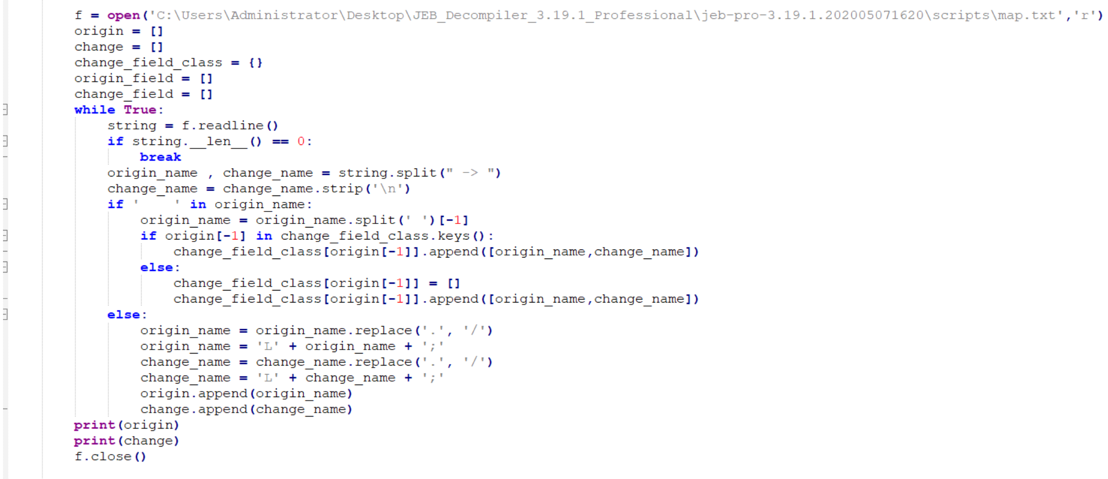
  Jeb 自动加载 deguard 给出的 map.txt，通过 deguard 给出的来 rename

- **How does your plugin work?**



```
from com.pnfsoftware.jeb.client.api import IScript
from com.pnfsoftware.jeb.core import RuntimeProjectUtil
from com.pnfsoftware.jeb.core.units.code import ICodeUnit, ICodeItem
from com.pnfsoftware.jeb.core.units.code.android import IDexUnit
from com.pnfsoftware.jeb.core.actions import Actions, ActionContext, ActionCommentData, ActionRenameData
from java.lang import Runnable

class JEB2DeGuardClass(IScript):
    def run(self, ctx):
        ctx.executeAsync("Running deobscure class ...", JEB2AutoRename(ctx))
        print('Done')

class JEB2AutoRename(Runnable):
    def __init__(self, ctx):
        self.ctx = ctx

    def run(self):
        ctx = self.ctx
        engctx = ctx.getEnginesContext()
        if not engctx:
            print('Back-end engines not initialized')
            return

        projects = engctx.getProjects()
        if not projects:
            print('There is no opened project')
            return

        prj = projects[0]

        units = RuntimeProjectUtil.findUnitsByType(prj, IDexUnit, False)
```

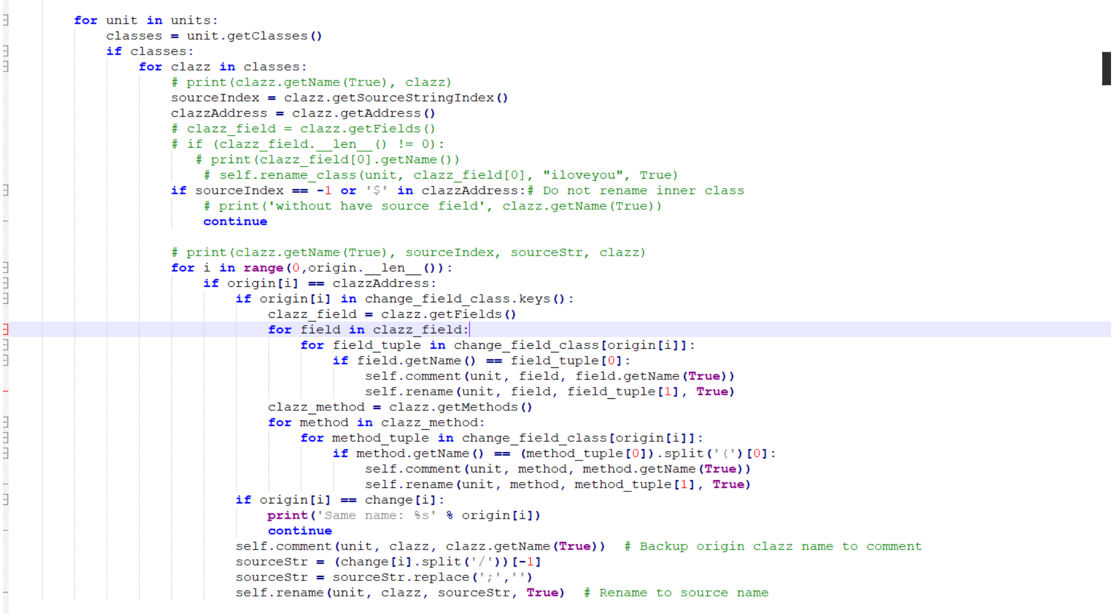
这一段是基本启动代码，把原来的助教给的代码弄了一下



```
        f = open('C:\Users\Administrator\Desktop\JEB_Decompiler_3.19.1_Professional\jeb-pro-3.19.1.202005071620\scripts\map.txt','r')
        origin = []
        change = []
        change_field_class = {}
        origin_field = []
        change_field = []
        while True:
            string = f.readline()
            if string.__len__() == 0:
                break
            origin_name , change_name = string.split(" -> ")
            change_name = change_name.strip('\n')
            if '    ' in origin_name:
                origin_name = origin_name.split(' ')[-1]
                if origin[-1] in change_field_class.keys():
                    change_field_class[origin[-1]].append([origin_name,change_name])
                else:
                    change_field_class[origin[-1]] = []
                    change_field_class[origin[-1]].append([origin_name,change_name])
            else:
                origin_name = origin_name.replace('.', '/')
                origin_name = 'L' + origin_name + ';'
                change_name = change_name.replace('.', '/')
                change_name = 'L' + change_name + ';'
                origin.append(origin_name)
                change.append(change_name)
        print(origin)
        print(change)
        f.close()
```

这一段是将 deguard 给的 mapping 进行读取**（如果助教重新跑记得改一下文**

件名称和 open 函数的 mapping 地址，否则会报错），将 origin（原本的类名）和 change（更改后的类名）进行一一对应，同时如果看到开头是空格的，说明是这个类里面的变量名更改，这个时候单独开了一个 change_field_class 的 dict，做成{原来的类名：[[原来 field 名,更改后 field 名],[原来 field 名 2,更改后 field 名 2], …]}这样一个对应形式。



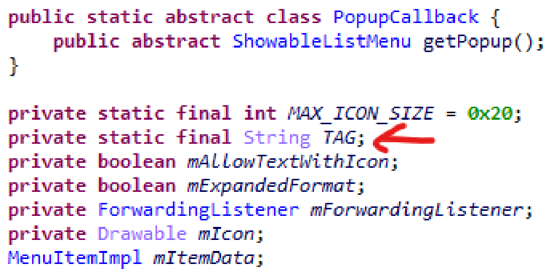
```
for unit in units:
    classes = unit.getClasses()
    if classes:
        for clazz in classes:
            # print(clazz.getName(True), clazz)
            sourceIndex = clazz.getSourceStringIndex()
            clazzAddress = clazz.getAddress()
            # clazz_field = clazz.getFields()
            # if (clazz_field.__len__() != 0):
               # print(clazz_field[0].getName())
                # self.rename_class(unit, clazz_field[0], "iloveyou", True)
            if sourceIndex == -1 or '$' in clazzAddress:# Do not rename inner class
                # print('without have source field', clazz.getName(True))
                continue

            # print(clazz.getName(True), sourceIndex, sourceStr, clazz)
            for i in range(0,origin.__len__()):
                if origin[i] == clazzAddress:
                    if origin[i] in change_field_class.keys():
                        clazz_field = clazz.getFields()
                        for field in clazz_field:
                            for field_tuple in change_field_class[origin[i]]:
                                if field.getName() == field_tuple[0]:
                                    self.comment(unit, field, field.getName(True))
                                    self.rename(unit, field, field_tuple[1], True)
                        clazz_method = clazz.getMethods()
                        for method in clazz_method:
                            for method_tuple in change_field_class[origin[i]]:
                                if method.getName() == (method_tuple[0]).split('(')[0]:
                                    self.comment(unit, method, method.getName(True))
                                    self.rename(unit, method, method_tuple[1], True)
                    if origin[i] == change[i]:
                        print('Same name: %s' % origin[i])
                        continue
                    self.comment(unit, clazz, clazz.getName(True))  # Backup origin clazz name to comment
                    sourceStr = (change[i].split('/'))[-1]
                    sourceStr = sourceStr.replace(';','')
                    self.rename(unit, clazz, sourceStr, True)  # Rename to source name
```

这里一些注释是我的测试过程，测试这些函数能不能用，以及 Address 在 jeb 里的存储形式。然后就是对所有类进行循环，如果是在需要重命名的里面的，就先看要不要改变量，要改就改，最后看一下类名需不需要修改（因为有许多类名是完全一样的，不知道 deguard 为什么都写在里面哈哈哈~~~~~算法还不太行~~）

最后的 rename 和 comment 就是参照助教的进行修改的，因为 field 也都有这些 method，所以基本不用动

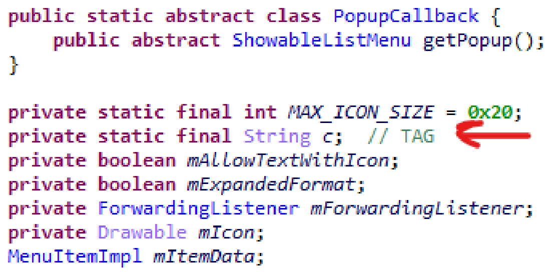
- **Screenshots of your plugin’s execution result.**

更改前



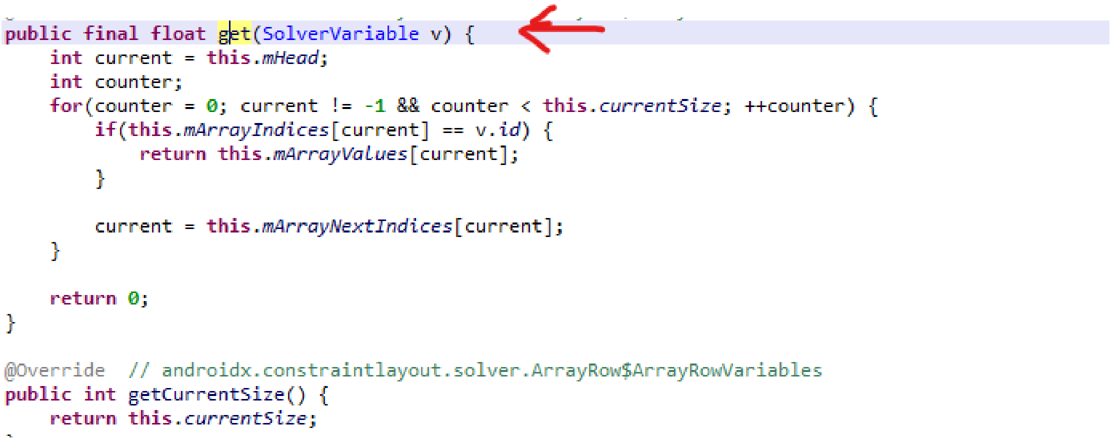
```
public static abstract class PopupCallback {
    public abstract ShowableListMenu getPopup();
}

private static final int MAX_ICON_SIZE = 0x20;
private static final String TAG;
private boolean mAllowTextWithIcon;
private boolean mExpandedFormat;
private ForwardingListener mForwardingListener;
private Drawable mIcon;
MenuItemImpl mItemData;
```

更改后



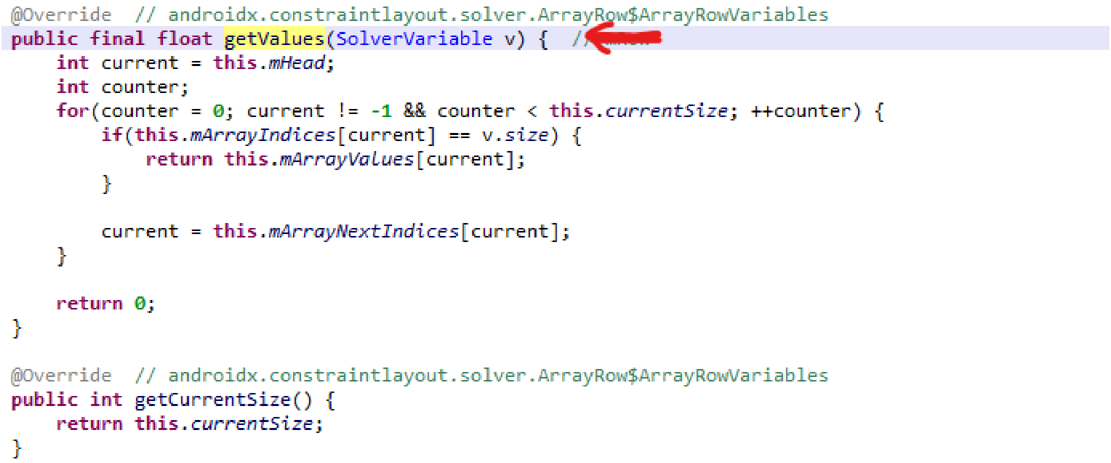
```
public static abstract class PopupCallback {
    public abstract ShowableListMenu getPopup();
}

private static final int MAX_ICON_SIZE = 0x20;
private static final String c;  // TAG
private boolean mAllowTextWithIcon;
private boolean mExpandedFormat;
private ForwardingListener mForwardingListener;
private Drawable mIcon;
MenuItemImpl mItemData;
```

更改前



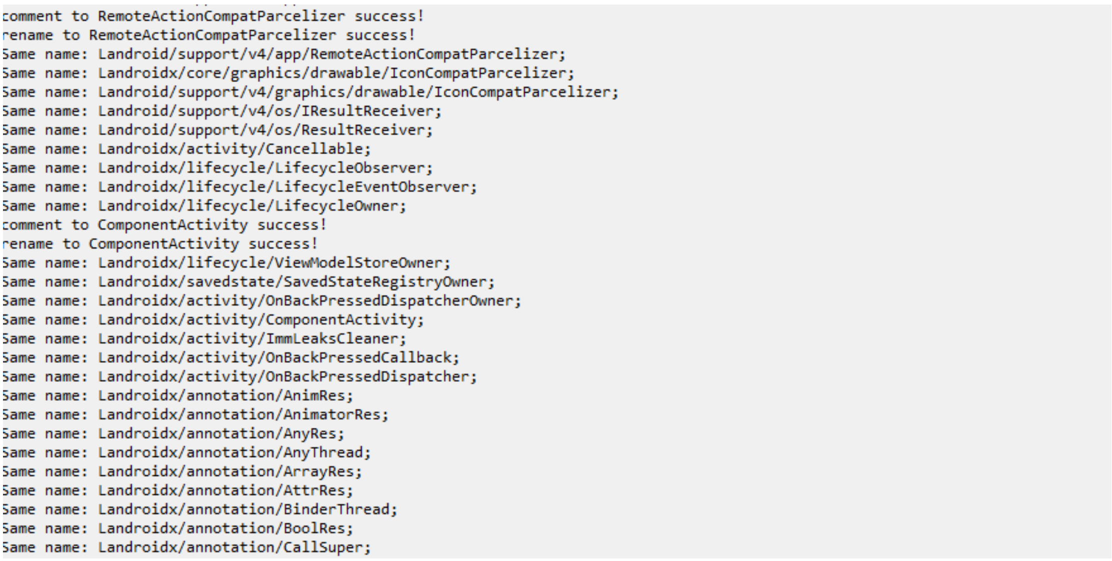
```
public final float get(SolverVariable v) {
    int current = this.mHead;
    int counter;
    for(counter = 0; current != -1 && counter < this.currentSize; ++counter) {
        if(this.mArrayIndices[current] == v.id) {
            return this.mArrayValues[current];
        }

        current = this.mArrayNextIndices[current];
    }

    return 0;
}

@Override  // androidx.constraintlayout.solver.ArrayRow$ArrayRowVariables
public int getCurrentSize() {
    return this.currentSize;
```

更改后

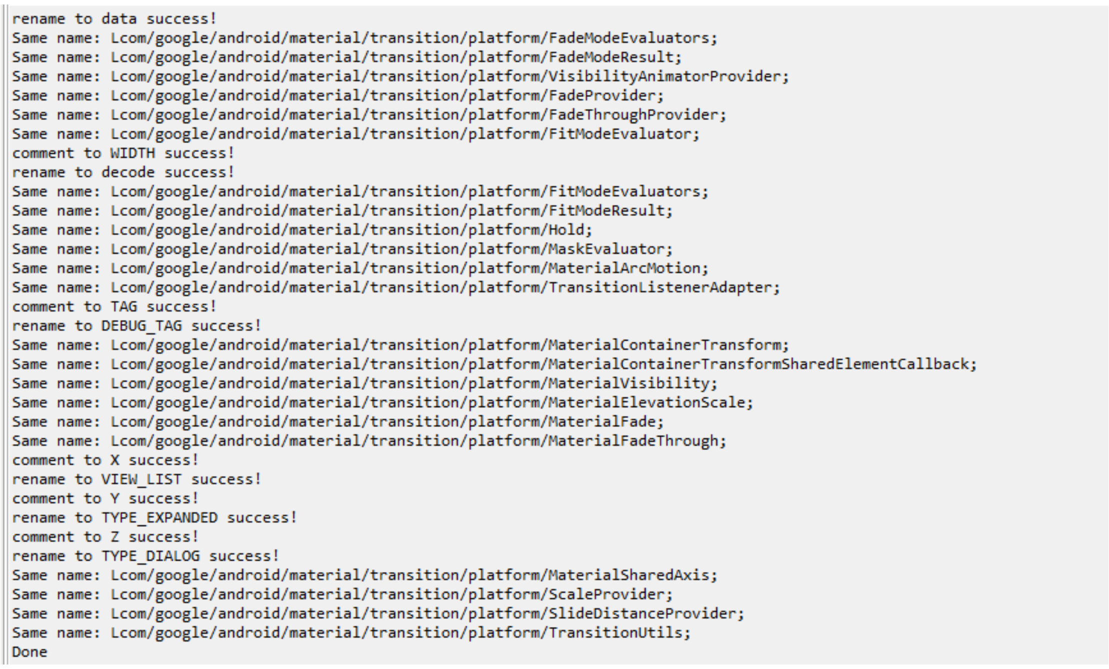


```
@Override  // androidx.constraintlayout.solver.ArrayRow$ArrayRowVariables
public final float getValues(SolverVariable v) {  //[illegible]
    int current = this.mHead;
    int counter;
    for(counter = 0; current != -1 && counter < this.currentSize; ++counter) {
        if(this.mArrayIndices[current] == v.size) {
            return this.mArrayValues[current];
        }

        current = this.mArrayNextIndices[current];
    }

    return 0;
}

@Override  // androidx.constraintlayout.solver.ArrayRow$ArrayRowVariables
public int getCurrentSize() {
    return this.currentSize;
}
```

运行记录



```
comment to RemoteActionCompatParcelizer success!
rename to RemoteActionCompatParcelizer success!
Same name: Landroid/support/v4/app/RemoteActionCompatParcelizer;
Same name: Landroidx/core/graphics/drawable/IconCompatParcelizer;
Same name: Landroid/support/v4/graphics/drawable/IconCompatParcelizer;
Same name: Landroid/support/v4/os/IResultReceiver;
Same name: Landroid/support/v4/os/ResultReceiver;
Same name: Landroidx/activity/Cancellable;
Same name: Landroidx/lifecycle/LifecycleObserver;
Same name: Landroidx/lifecycle/LifecycleEventObserver;
Same name: Landroidx/lifecycle/LifecycleOwner;
comment to ComponentActivity success!
rename to ComponentActivity success!
Same name: Landroidx/lifecycle/ViewModelStoreOwner;
Same name: Landroidx/savedstate/SavedStateRegistryOwner;
Same name: Landroidx/activity/OnBackPressedDispatcherOwner;
Same name: Landroidx/activity/ComponentActivity;
Same name: Landroidx/activity/ImmLeaksCleaner;
Same name: Landroidx/activity/OnBackPressedCallback;
Same name: Landroidx/activity/OnBackPressedDispatcher;
Same name: Landroidx/annotation/AnimRes;
Same name: Landroidx/annotation/AnimatorRes;
Same name: Landroidx/annotation/AnyRes;
Same name: Landroidx/annotation/AnyThread;
Same name: Landroidx/annotation/ArrayRes;
Same name: Landroidx/annotation/AttrRes;
Same name: Landroidx/annotation/BinderThread;
Same name: Landroidx/annotation/BoolRes;
Same name: Landroidx/annotation/CallSuper;
```



```
rename to data success!
Same name: Lcom/google/android/material/transition/platform/FadeModeEvaluators;
Same name: Lcom/google/android/material/transition/platform/FadeModeResult;
Same name: Lcom/google/android/material/transition/platform/VisibilityAnimatorProvider;
Same name: Lcom/google/android/material/transition/platform/FadeProvider;
Same name: Lcom/google/android/material/transition/platform/FadeThroughProvider;
Same name: Lcom/google/android/material/transition/platform/FitModeEvaluator;
comment to WIDTH success!
rename to decode success!
Same name: Lcom/google/android/material/transition/platform/FitModeEvaluators;
Same name: Lcom/google/android/material/transition/platform/FitModeResult;
Same name: Lcom/google/android/material/transition/platform/Hold;
Same name: Lcom/google/android/material/transition/platform/MaskEvaluator;
Same name: Lcom/google/android/material/transition/platform/MaterialArcMotion;
Same name: Lcom/google/android/material/transition/platform/TransitionListenerAdapter;
comment to TAG success!
rename to DEBUG_TAG success!
Same name: Lcom/google/android/material/transition/platform/MaterialContainerTransform;
Same name: Lcom/google/android/material/transition/platform/MaterialContainerTransformSharedElementCallback;
Same name: Lcom/google/android/material/transition/platform/MaterialVisibility;
Same name: Lcom/google/android/material/transition/platform/MaterialElevationScale;
Same name: Lcom/google/android/material/transition/platform/MaterialFade;
Same name: Lcom/google/android/material/transition/platform/MaterialFadeThrough;
comment to X success!
rename to VIEW_LIST success!
comment to Y success!
rename to TYPE_EXPANDED success!
comment to Z success!
rename to TYPE_DIALOG success!
Same name: Lcom/google/android/material/transition/platform/MaterialSharedAxis;
Same name: Lcom/google/android/material/transition/platform/ScaleProvider;
Same name: Lcom/google/android/material/transition/platform/SlideDistanceProvider;
Same name: Lcom/google/android/material/transition/platform/TransitionUtils;
Done
```