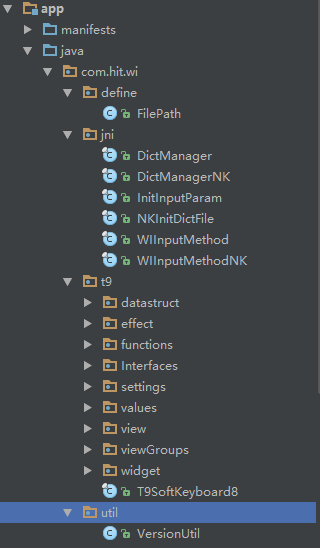
# 微逸输入法结构说明

### 目录结构说明：



define：定义了系统存放文件的路径

jni：与内核有关的类

t9：输入法相关的包

1. Datastruct:输入法的皮肤颜色信息和
2. Effect：输入法的键盘响应类，如按键声音和震动。
3. Functions：一些管理器，如文件管理器，表情包管理，SharedPreference的管理。
4. Interfaces：保存输入法的一些接口。
5. settings：输入法的设置界面，如定义皮肤，定义点划等。
6. values：全局变量、皮肤信息的管理器、表情包的管理器。
7. view：输入法键盘的各种View，Button的定义。如数字键、九键、设置键盘大小的View、点划光线等。
8. viewgroups:输入法键盘的ViewGroups。
9. widget：选择键盘透明度的进度条。

Util：工具类。现在里边只有获取输入法版本号的类。

### 二、输入法运行机制：

用户选择微逸输入法：onCreate()->onCreateInputView();

用户点击输入框：onStartInputView()->onWindowShow();

用户离开输入法界面：onWindowHide();

输入法服务停止（用户切换别的输入法，或者关闭输入法）：OnDestroy()；

\*输入法创建初期内核的加载、向服务器发送安装信息创建输入法键盘的ViewGroup，创建输入法的键盘布局Layout，获取屏幕信息，设置键盘透明度等方法。

***其中：*** 键盘ViewGroup的创建都由T9SoftkeyBoard8的内部类KeyBoardCreate来创建。

不同键盘之间的切换使用KeyBoardSwitcherC来管理。

皮肤的相关功能由SkinUpdateC来管理

屏幕的信息由ScreenInfoC来管理

所有view大小和位置更新方法使用ViewSizeUpdateC来管理

keyboardLayout、设置键盘大小的view的创建与删除使用ViewManagerC管理

#### onCreate():

实现的功能

1. 加载内核：
   1. NKInitDictFile.NKInitWiDict(this);
   2. new InitInputParam();
2. 初始化SharedPreference:主要是初始化系统默认的点划设置。
   1. SharedPreferenceManager.initSharedPreferencesData(this);
3. 实例化键盘ViewGroup（具体效果见后边的界面部分）
   1. 初始化键盘点击的声音和震动效果：KeyBoardTouchEffect
   2. 特殊符号：SpecialSymbolChooseViewGroup
   3. 上侧功能栏：FunctionViewGroup
   4. 第二层符号栏：QuickSymbolViewGroup
   5. 全选，复制，粘贴，删除: PreFixViewGroup
   6. 最底部的ViewGroup：BottomBarViewGroup
   7. 英文键盘的Candidate(候选词) QKCandidatesViewGroup
   8. 英文键盘: QKInputViewGroups
   9. 九键的ViewGroup
   10. 点划的效果：LightViewManager
4. 创建键盘的Layout：
   1. KeyboardLayout:键盘的整体布局
   2. secondLayerLayout：存放quicksymbol、special、prefix、delete的地方(因为这些布局太零散，所以用一个布局整体管理)
5. 键盘皮肤信息的加载：skinInfoManager
6. 向服务器发送安装信息：gm.generate();
7. 初始化屏幕信息：screenInfoC.refreshScreenInfo();
8. 创建输入法的布局：keyBoardCreate.createKeyboard();
9. 设置键盘的透明度：transparencyHandle.startAutoDownAlpha();

#### onStartInputView():

输入法切换时调用，判断显示全键还是九键(不显示英文键盘，即使在之前选择的是英文键盘，也会变成中文全键)。

#### OnWindowShow()

在创建的时候完成对键盘透明度的配置，文字shadow的加载，并判断当前的键盘类型，出场动画、设置空格键文字、更新皮肤，加载音效设置，并由keyBoardSwitcher选择键盘。

其中mInputViewGG为九键和全键的Layout

#### onWindowHidden()

刷新界面，开始出场动画。

### 三、界面部分：

#### 键盘ViewGroup的说明

##### SpecialSymbolChooseViewGroup



图3.1

##### FunctionViewGroup



图3.2

##### QuickSymbolViewGroup



图3.3

##### PreFixViewGroup

这个ViewGroup是长按左下角的而显示的。



图3.4

##### BottomBarViewGroup



图3.5

##### QKCandidatesViewGroup



图3.6

##### PreEdit



图3.7

##### QKInputViewGroups



图3.8

##### 九键的ViewGroup



图3.9

##### 点划的效果：LightViewManager



图3.10

##### 各个ViewGroup的继承关系

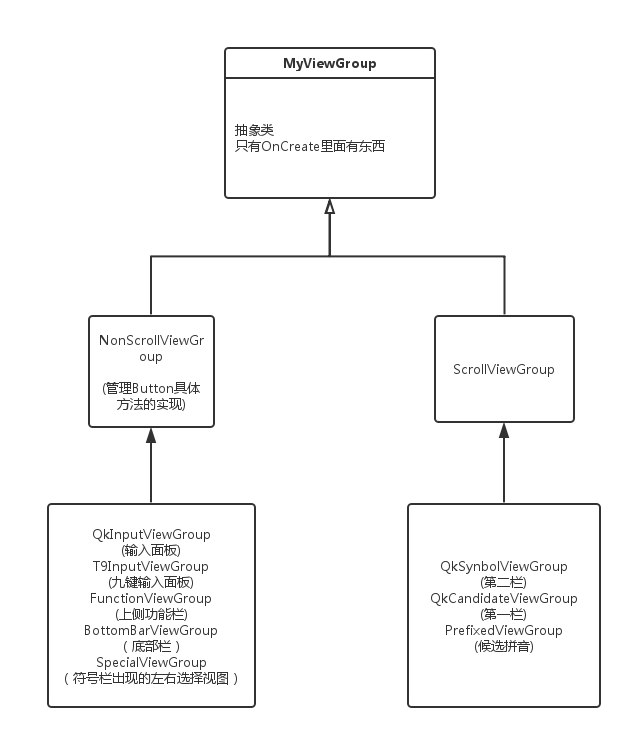


图3.11

#### 注意事项：