**Methods:**

getActiveNetworkInfo

getAllNetworkInfo

getBackgroundDataSetting

getInterfaceName

getLastTetherError

getMobileDataEnabled

getNetworkInfo

getNetworkPreference

getTetherableIfaces

getTetherableUsbRegexs

getTetherableWifiRegexs

getTetheredIfaces

getTetheringErroredIfaces

isTetheringSupported

onSwitchToSim1DataNetworkCallback

onSwitchToSim2DataNetworkCallback

reportInetCondition

requestRouteToHost

setBackgroundDataSetting

setMobileDataEnabled

setNetworkPreference

setRadio

setRadios

startUsingNetworkFeature

stopUsingNetworkFeature

switchToSim1DataNetwork

switchToSim2DataNetwork

tether

untether

isNetworkTypeValid

**Program:**

**public** **static** String setTetheringState(**boolean** state)

{

*tetheringState*=state;

String methodNames="";

**try**

{

*tetheringManager*=(ConnectivityManager)*context*.getSystemService(Context.*CONNECTIVITY\_SERVICE*);

Method[] tetheringMethods = *tetheringManager*.getClass().getDeclaredMethods();

String str = "";

**if** (*tetheringState*)

str = "tether";

**else**

str = "untether";

**for** (Method method : tetheringMethods)

{

methodNames+=method.getName()+"\n";

**if** (method.getName().equals(str))

{

**try**

{

method.invoke(*tetheringManager*, "usb0");

//Integer code = (Integer) method.invoke(tetheringManager, "usb0");

//code = (Integer) method.invoke(tetheringManager, "setting TH");

}

**catch** (IllegalArgumentException e)

{

e.printStackTrace();

}

**catch** (IllegalAccessException e)

{

e.printStackTrace();

}

**catch** (InvocationTargetException e)

{

e.printStackTrace();

}

}

}

}

**catch**(Exception e)

{

}

**return** methodNames;

}