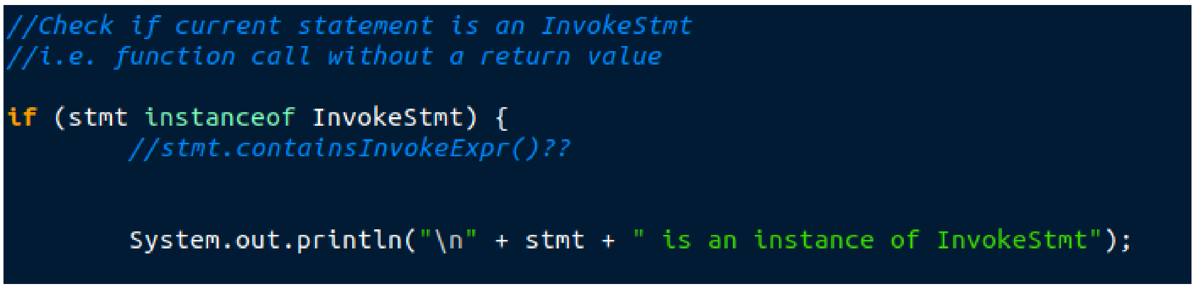
Lab 2: Static Analysis of API Usages in Java Classes

Part 1:

(a) Check if current statement is an `InvokeStmt`

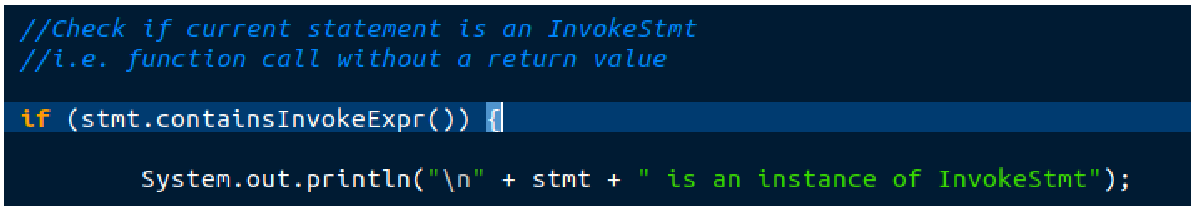
I did this using the `instanceof` operator, by checking if the statement is an instance of `InvokeStmt` in soot.



```
//Check if current statement is an InvokeStmt
//i.e. function call without a return value

if (stmt instanceof InvokeStmt) {
        //stmt.containsInvokeExpr()??


        System.out.println("\n" + stmt + " is an instance of InvokeStmt");
```

Another way to do this is possibly using `Stmt.containsInvokeExpr()` which gives us some more results (possibly duplicates) as the output.

As the question specifically asks is to check whether the current statement is an instance of `InvokeStmt`, I used the `instanceof` operator in the final code.



```
 //Check if current statement is an InvokeStmt
 //i.e. function call without a return value

 if (stmt.containsInvokeExpr()) {

         System.out.println("\n" + stmt + " is an instance of InvokeStmt");
```

(b) Check if the declaring class is defined in the Application under analysis. If not, method `m` is an API

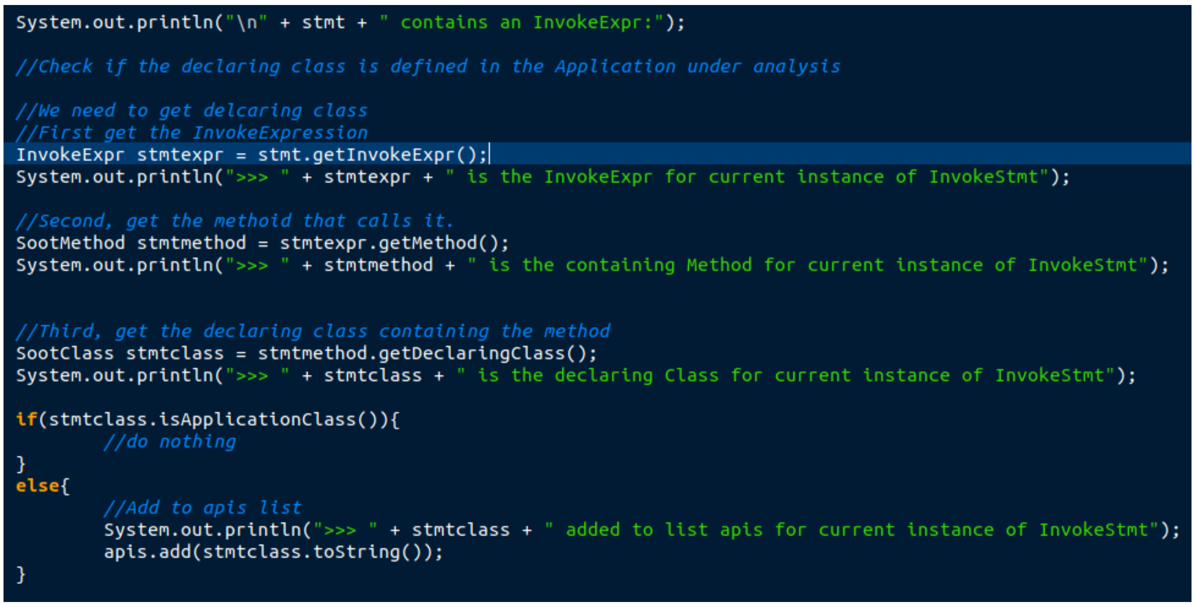
We can get the declaring class using `SootMethod.getDeclaringClass()`. But first, we need to get the `SootMethod` of the `InvokeExpr`.

Thus, first we needed to get the `InvokeExpr`. I did this using the `InvokeStmt.getInvokeExpr()` method. All these methods were found using the soot API documentation.

Then, using the `InvokeExpr`, we could get the calling Method. This was done by using the `SootMethod.getMethod()` method.

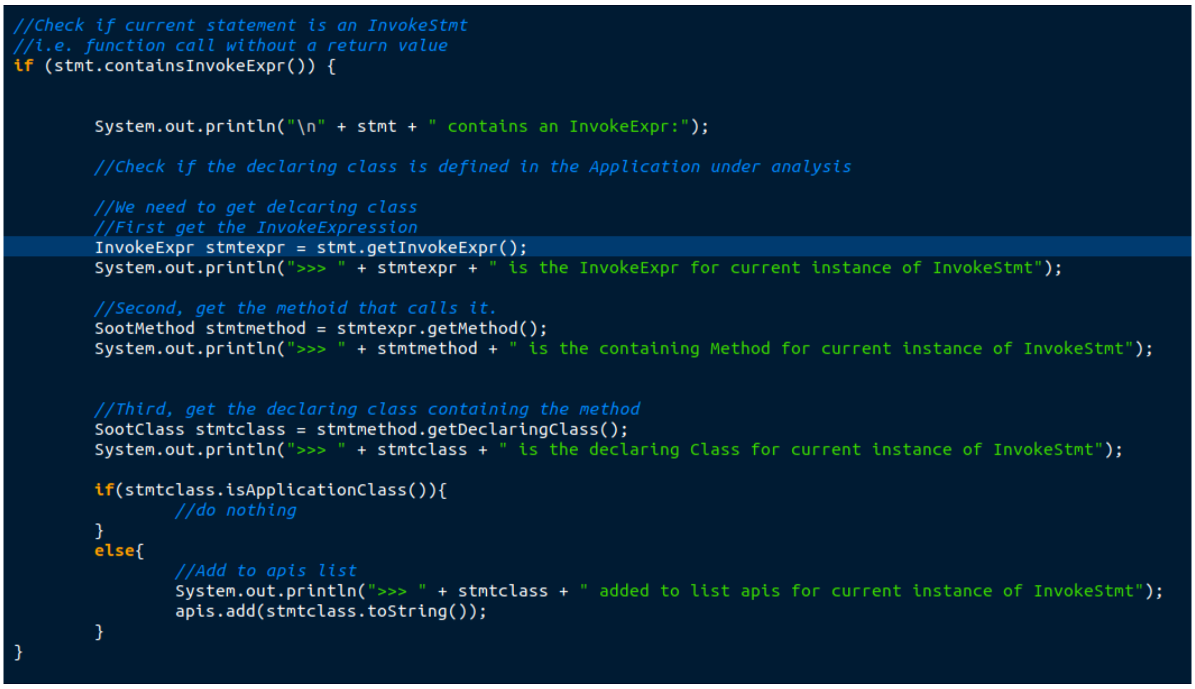
Then, finally, we could get the Declaring Class using the `SootMethod.getDeclaringClass()` method which returns a `SootClass`.

Finally, once we have the required `SootClass`, we add it to our list of `apis`.



```
System.out.println("\n" + stmt + " contains an InvokeExpr:");

//Check if the declaring class is defined in the Application under analysis

//We need to get delcaring class
//First get the InvokeExpression
InvokeExpr stmtexpr = stmt.getInvokeExpr();
System.out.println(">>> " + stmtexpr + " is the InvokeExpr for current instance of InvokeStmt");

//Second, get the methoid that calls it.
SootMethod stmtmethod = stmtexpr.getMethod();
System.out.println(">>> " + stmtmethod + " is the containing Method for current instance of InvokeStmt");


//Third, get the declaring class containing the method
SootClass stmtclass = stmtmethod.getDeclaringClass();
System.out.println(">>> " + stmtclass + " is the declaring Class for current instance of InvokeStmt");

if(stmtclass.isApplicationClass()){
        //do nothing
}
else{
        //Add to apis list
        System.out.println(">>> " + stmtclass + " added to list apis for current instance of InvokeStmt");
        apis.add(stmtclass.toString());
}
```

As a whole:

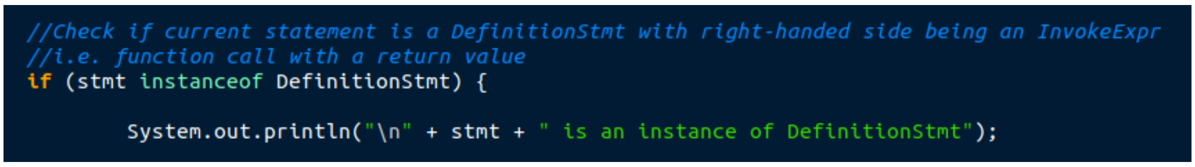


```
//Check if current statement is an InvokeStmt
//i.e. function call without a return value
if (stmt.containsInvokeExpr()) {


        System.out.println("\n" + stmt + " contains an InvokeExpr:");

        //Check if the declaring class is defined in the Application under analysis

        //We need to get delcaring class
        //First get the InvokeExpression
        InvokeExpr stmtexpr = stmt.getInvokeExpr();
        System.out.println(">>> " + stmtexpr + " is the InvokeExpr for current instance of InvokeStmt");

        //Second, get the methoid that calls it.
        SootMethod stmtmethod = stmtexpr.getMethod();
        System.out.println(">>> " + stmtmethod + " is the containing Method for current instance of InvokeStmt");


        //Third, get the declaring class containing the method
        SootClass stmtclass = stmtmethod.getDeclaringClass();
        System.out.println(">>> " + stmtclass + " is the declaring Class for current instance of InvokeStmt");

        if(stmtclass.isApplicationClass()){
                //do nothing
        }
        else{
                //Add to apis list
                System.out.println(">>> " + stmtclass + " added to list apis for current instance of InvokeStmt");
                apis.add(stmtclass.toString());
        }
}
```

Part 2:

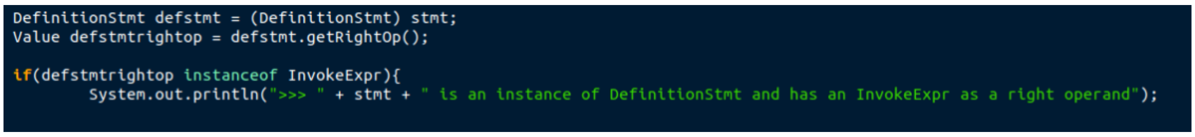
(a) Check if current statement is a `DefinitionStmt`, with the right operand being an `InvokeExpr`

I did this using the `instanceof` operator, by checking if the statement is an instance of `DefinitionStmt` in soot.



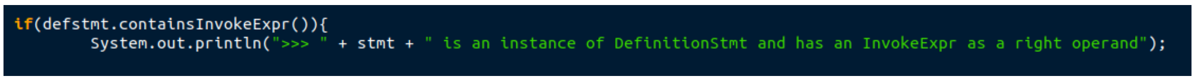
```java
//Check if current statement is a DefinitionStmt with right-handed side being an InvokeExpr
//i.e. function call with a return value
if (stmt instanceof DefinitionStmt) {

        System.out.println("\n" + stmt + " is an instance of DefinitionStmt");
```

I checked whether the right operand is an InvokeExpr by getting the right operand with `getRightOp()` and checking if it was an instance of `InvokeExpr`.



```java
DefinitionStmt defstmt = (DefinitionStmt) stmt;
Value defstmtrightop = defstmt.getRightOp();

if(defstmtrightop instanceof InvokeExpr){
        System.out.println(">>> " + stmt + " is an instance of DefinitionStmt and has an InvokeExpr as a right operand");
```

We could also do this using the `Stmt.containsInvokeExpr()` method if we do not need to care about which operand (left or right) has it. This is because we cannot use `Stmt.containsInvokeExpr()` on a single side operand of `DefinitionStmt`. Output was identical to that of using the `instanceof` operator.



```java
if(defstmt.containsInvokeExpr()){
        System.out.println(">>> " + stmt + " is an instance of DefinitionStmt and has an InvokeExpr as a right operand");
```

(b) Check if the declaring class is defined in the Application under analysis. If not, method `m` is an API

Again, as before, we can get the declaring class using `SootMethod.getDeclaringClass()`. But first, we need to get the `SootMethod` of the `InvokeExpr`.

Thus, first we needed to get the `InvokeExpr`. I did this using the `InvokeStmt.getInvokeExpr()` method. All these methods were found using the soot API documentation.

Then, using the `InvokeExpr`, we could get the calling Method. This was done by using the `SootMethod.getMethod()` method.

Then, finally, we could get the Declaring Class using the `SootMethod.getDeclaringClass()` method which returns a `SootClass`.

Finally, once we have the required `SootClass`, we add it to our list of `apis`.

```java
DefinitionStmt defstmt = (DefinitionStmt) stmt;
Value defstmtrightop = defstmt.getRightOp();

if(defstmtrightop instanceof InvokeExpr){
        System.out.println(">>> " + stmt + " is an instance of DefinitionStmt and has an InvokeExpr as a right operand");


        //Check if the declaring class is defined in the Application under analysis
        //If not, method m is an API

        //We need to get declaring class
        //First get the InvokeExpression
        InvokeExpr stmtexpr = stmt.getInvokeExpr();
        System.out.println(">>> " + stmtexpr + " is the InvokeExpr for current instance of DefinitionStmt");

        //Second, get the methoid that calls it.
        SootMethod stmtmethod = stmtexpr.getMethod();
        System.out.println(">>> " + stmtmethod + " is the containing Method for current instance of DefinitionStmt");


        //Third, get the declaring class containing the method
        SootClass stmtclass = stmtmethod.getDeclaringClass();
        System.out.println(">>> " + stmtclass + " is the declaring Class for current instance of DefinitionStmt");

        if(stmtclass.isApplicationClass()){
                //do nothing
        }
        else{
                //Add to apis list
                System.out.println(">>> " + stmtclass + " added to list apis for current instance of DefinitionStmt\n");
                apis.add(stmtclass.toString());
        }

}
```

As a whole:

```java
//Check if current statement is a DefinitionStmt with right-handed side being an InvokeExpr
//i.e. function call with a return value
if (stmt instanceof DefinitionStmt) {

        System.out.println("\n" + stmt + " is an instance of DefinitionStmt");

        DefinitionStmt defstmt = (DefinitionStmt) stmt;
        Value defstmtrightop = defstmt.getRightOp();

        if(defstmtrightop instanceof InvokeExpr){
                System.out.println(">>> " + stmt + " is an instance of DefinitionStmt and has an InvokeExpr as a right operand");


                //Check if the declaring class is defined in the Application under analysis
                //If not, method m is an API

                //We need to get declaring class
                //First get the InvokeExpression
                InvokeExpr stmtexpr = stmt.getInvokeExpr();
                System.out.println(">>> " + stmtexpr + " is the InvokeExpr for current instance of DefinitionStmt");

                //Second, get the methoid that calls it.
                SootMethod stmtmethod = stmtexpr.getMethod();
                System.out.println(">>> " + stmtmethod + " is the containing Method for current instance of DefinitionStmt");


                //Third, get the declaring class containing the method
                SootClass stmtclass = stmtmethod.getDeclaringClass();
                System.out.println(">>> " + stmtclass + " is the declaring Class for current instance of DefinitionStmt");

                if(stmtclass.isApplicationClass()){
                        //do nothing
                }
                else{
                        //Add to apis list
                        System.out.println(">>> " + stmtclass + " added to list apis for current instance of DefinitionStmt\n");
                        apis.add(stmtclass.toString());
                }

        }
}
```

Output:

Raw Output:

```
Processing class unknown under ./unknown/, output directory:output_unknown
looking up APIs...
Soot started on Sun Mar 06 19:34:43 MST 2022

Analyzing method: <test.Unknown: void <init>(java.lang.String,java.lang.String)>



r0 := @this: test.Unknown is an instance of DefinitionStmt
```

```
r1 := @parameter0: java.lang.String is an instance of DefinitionStmt

r2 := @parameter1: java.lang.String is an instance of DefinitionStmt

specialinvoke r0.<android.app.Activity: void <init>()>() is an instance of
InvokeStmt
>>> specialinvoke r0.<android.app.Activity: void <init>()>() is the InvokeExpr
for current instance of InvokeStmt
>>> <android.app.Activity: void <init>()> is the containing Method for current
instance of InvokeStmt
>>> android.app.Activity is the declaring Class for current instance of
InvokeStmt
>>> android.app.Activity added to list apis for current instance of InvokeStmt

r0.<test.Unknown: java.lang.String phoneNo> = r1 is an instance of DefinitionStmt

r0.<test.Unknown: java.lang.String sms> = r2 is an instance of DefinitionStmt

Analyzing method: <test.Unknown: void aaaa()>


r0 := @this: test.Unknown is an instance of DefinitionStmt

r8 = staticinvoke <android.telephony.SmsManager: android.telephony.SmsManager
getDefault()>() is an instance of DefinitionStmt
>>> r8 = staticinvoke <android.telephony.SmsManager: android.telephony.SmsManager
getDefault()>() is an instance of DefinitionStmt and has an InvokeExpr as a right
operand
>>> staticinvoke <android.telephony.SmsManager: android.telephony.SmsManager
getDefault()>() is the InvokeExpr for current instance of DefinitionStmt
>>> <android.telephony.SmsManager: android.telephony.SmsManager getDefault()> is
the containing Method for current instance of DefinitionStmt
>>> android.telephony.SmsManager is the declaring Class for current instance of
DefinitionStmt
>>> android.telephony.SmsManager added to list apis for current instance of
DefinitionStmt


$r2 = r0.<test.Unknown: java.lang.String phoneNo> is an instance of
DefinitionStmt

$r1 = r0.<test.Unknown: java.lang.String sms> is an instance of DefinitionStmt

virtualinvoke r8.<android.telephony.SmsManager: void
sendTextMessage(java.lang.String,java.lang.String,java.lang.String,android.app.Pe
ndingIntent,android.app.PendingIntent)>($r2, null, $r1, null, null) is an
instance of InvokeStmt
>>> virtualinvoke r8.<android.telephony.SmsManager: void
sendTextMessage(java.lang.String,java.lang.String,java.lang.String,android.app.Pe
ndingIntent,android.app.PendingIntent)>($r2, null, $r1, null, null) is the
InvokeExpr for current instance of InvokeStmt
>>> <android.telephony.SmsManager: void
sendTextMessage(java.lang.String,java.lang.String,java.lang.String,android.app.Pe
ndingIntent,android.app.PendingIntent)> is the containing Method for current
instance of InvokeStmt
>>> android.telephony.SmsManager is the declaring Class for current instance of
InvokeStmt
```

```
>>> android.telephony.SmsManager added to list apis for current instance of
InvokeStmt

$r3 = virtualinvoke r0.<test.Unknown: android.content.Context
getApplicationContext()>() is an instance of DefinitionStmt
>>> $r3 = virtualinvoke r0.<test.Unknown: android.content.Context
getApplicationContext()>() is an instance of DefinitionStmt and has an InvokeExpr
as a right operand
>>> virtualinvoke r0.<test.Unknown: android.content.Context
getApplicationContext()>() is the InvokeExpr for current instance of
DefinitionStmt
>>> <android.content.ContextWrapper: android.content.Context
getApplicationContext()> is the containing Method for current instance of
DefinitionStmt
>>> android.content.ContextWrapper is the declaring Class for current instance of
DefinitionStmt
>>> android.content.ContextWrapper added to list apis for current instance of
DefinitionStmt


$r4 = staticinvoke <android.widget.Toast: android.widget.Toast
makeText(android.content.Context,java.lang.CharSequence,int)>($r3, "SMS Sent!",
1) is an instance of DefinitionStmt
>>> $r4 = staticinvoke <android.widget.Toast: android.widget.Toast
makeText(android.content.Context,java.lang.CharSequence,int)>($r3, "SMS Sent!",
1) is an instance of DefinitionStmt and has an InvokeExpr as a right operand
>>> staticinvoke <android.widget.Toast: android.widget.Toast
makeText(android.content.Context,java.lang.CharSequence,int)>($r3, "SMS Sent!",
1) is the InvokeExpr for current instance of DefinitionStmt
>>> <android.widget.Toast: android.widget.Toast
makeText(android.content.Context,java.lang.CharSequence,int)> is the containing
Method for current instance of DefinitionStmt
>>> android.widget.Toast is the declaring Class for current instance of
DefinitionStmt
>>> android.widget.Toast added to list apis for current instance of
DefinitionStmt


virtualinvoke $r4.<android.widget.Toast: void show()>() is an instance of
InvokeStmt
>>> virtualinvoke $r4.<android.widget.Toast: void show()>() is the InvokeExpr for
current instance of InvokeStmt
>>> <android.widget.Toast: void show()> is the containing Method for current
instance of InvokeStmt
>>> android.widget.Toast is the declaring Class for current instance of
InvokeStmt
>>> android.widget.Toast added to list apis for current instance of InvokeStmt

$r5 := @caughtexception is an instance of DefinitionStmt

$r6 = virtualinvoke r0.<test.Unknown: android.content.Context
getApplicationContext()>() is an instance of DefinitionStmt
>>> $r6 = virtualinvoke r0.<test.Unknown: android.content.Context
getApplicationContext()>() is an instance of DefinitionStmt and has an InvokeExpr
as a right operand
>>> virtualinvoke r0.<test.Unknown: android.content.Context
getApplicationContext()>() is the InvokeExpr for current instance of
DefinitionStmt
```

```
>>> <android.content.ContextWrapper: android.content.Context
getApplicationContext()> is the containing Method for current instance of
DefinitionStmt
>>> android.content.ContextWrapper is the declaring Class for current instance of
DefinitionStmt
>>> android.content.ContextWrapper added to list apis for current instance of
DefinitionStmt


$r7 = staticinvoke <android.widget.Toast: android.widget.Toast
makeText(android.content.Context,java.lang.CharSequence,int)>($r6, "SMS faild,
please try again later!", 1) is an instance of DefinitionStmt
>>> $r7 = staticinvoke <android.widget.Toast: android.widget.Toast
makeText(android.content.Context,java.lang.CharSequence,int)>($r6, "SMS faild,
please try again later!", 1) is an instance of DefinitionStmt and has an
InvokeExpr as a right operand
>>> staticinvoke <android.widget.Toast: android.widget.Toast
makeText(android.content.Context,java.lang.CharSequence,int)>($r6, "SMS faild,
please try again later!", 1) is the InvokeExpr for current instance of
DefinitionStmt
>>> <android.widget.Toast: android.widget.Toast
makeText(android.content.Context,java.lang.CharSequence,int)> is the containing
Method for current instance of DefinitionStmt
>>> android.widget.Toast is the declaring Class for current instance of
DefinitionStmt
>>> android.widget.Toast added to list apis for current instance of
DefinitionStmt


virtualinvoke $r7.<android.widget.Toast: void show()>() is an instance of
InvokeStmt
>>> virtualinvoke $r7.<android.widget.Toast: void show()>() is the InvokeExpr for
current instance of InvokeStmt
>>> <android.widget.Toast: void show()> is the containing Method for current
instance of InvokeStmt
>>> android.widget.Toast is the declaring Class for current instance of
InvokeStmt
>>> android.widget.Toast added to list apis for current instance of InvokeStmt

virtualinvoke $r5.<java.lang.Exception: void printStackTrace()>() is an instance
of InvokeStmt
>>> virtualinvoke $r5.<java.lang.Exception: void printStackTrace()>() is the
InvokeExpr for current instance of InvokeStmt
>>> <java.lang.Throwable: void printStackTrace()> is the containing Method for
current instance of InvokeStmt
>>> java.lang.Throwable is the declaring Class for current instance of InvokeStmt
>>> java.lang.Throwable added to list apis for current instance of InvokeStmt

Analyzing method: <test.Unknown: void bbbb()>


r0 := @this: test.Unknown is an instance of DefinitionStmt

$r3 = virtualinvoke r0.<test.Unknown: java.lang.Object
getSystemService(java.lang.String)>("location") is an instance of DefinitionStmt
>>> $r3 = virtualinvoke r0.<test.Unknown: java.lang.Object
getSystemService(java.lang.String)>("location") is an instance of DefinitionStmt
and has an InvokeExpr as a right operand
```

```
>>> virtualinvoke r0.<test.Unknown: java.lang.Object
getSystemService(java.lang.String)>("location") is the InvokeExpr for current
instance of DefinitionStmt
>>> <android.app.Activity: java.lang.Object getSystemService(java.lang.String)>
is the containing Method for current instance of DefinitionStmt
>>> android.app.Activity is the declaring Class for current instance of
DefinitionStmt
>>> android.app.Activity added to list apis for current instance of
DefinitionStmt


r1 = (android.location.LocationManager) $r3 is an instance of DefinitionStmt

r2 = virtualinvoke r1.<android.location.LocationManager:
android.location.Location getLastKnownLocation(java.lang.String)>("gps") is an
instance of DefinitionStmt
>>> r2 = virtualinvoke r1.<android.location.LocationManager:
android.location.Location getLastKnownLocation(java.lang.String)>("gps") is an
instance of DefinitionStmt and has an InvokeExpr as a right operand
>>> virtualinvoke r1.<android.location.LocationManager: android.location.Location
getLastKnownLocation(java.lang.String)>("gps") is the InvokeExpr for current
instance of DefinitionStmt
>>> <android.location.LocationManager: android.location.Location
getLastKnownLocation(java.lang.String)> is the containing Method for current
instance of DefinitionStmt
>>> android.location.LocationManager is the declaring Class for current instance
of DefinitionStmt
>>> android.location.LocationManager added to list apis for current instance of
DefinitionStmt


virtualinvoke r2.<android.location.Location: double getLatitude()>() is an
instance of InvokeStmt
>>> virtualinvoke r2.<android.location.Location: double getLatitude()>() is the
InvokeExpr for current instance of InvokeStmt
>>> <android.location.Location: double getLatitude()> is the containing Method
for current instance of InvokeStmt
>>> android.location.Location is the declaring Class for current instance of
InvokeStmt
>>> android.location.Location added to list apis for current instance of
InvokeStmt

virtualinvoke r2.<android.location.Location: double getLongitude()>() is an
instance of InvokeStmt
>>> virtualinvoke r2.<android.location.Location: double getLongitude()>() is the
InvokeExpr for current instance of InvokeStmt
>>> <android.location.Location: double getLongitude()> is the containing Method
for current instance of InvokeStmt
>>> android.location.Location is the declaring Class for current instance of
InvokeStmt
>>> android.location.Location added to list apis for current instance of
InvokeStmt
Soot finished on Sun Mar 06 19:34:43 MST 2022
Soot has run for 0 min. 0 sec.
<test.Unknown: void <init>(java.lang.String,java.lang.String)> calls APIs:
android.app.Activity

<test.Unknown: void aaaa()> calls APIs:
```

```
android.telephony.SmsManager
android.telephony.SmsManager
android.content.ContextWrapper
android.widget.Toast
android.widget.Toast
android.content.ContextWrapper
android.widget.Toast
android.widget.Toast
java.lang.Throwable

<test.Unknown: void bbbb()> calls APIs:
android.app.Activity
android.location.LocationManager
android.location.Location
android.location.Location
```

Output Explained:

As we can see, we get the following list of APIs in our result:

```
<test.Unknown: void aaaa()> calls APIs:
android.telephony.SmsManager
android.telephony.SmsManager
android.content.ContextWrapper
android.widget.Toast
android.widget.Toast
android.content.ContextWrapper
android.widget.Toast
android.widget.Toast
java.lang.Throwable

<test.Unknown: void bbbb()> calls APIs:
android.app.Activity
android.location.LocationManager
android.location.Location
android.location.Location
```

If we use contains `Stmt.containsInvokeExpr()` instead of `instanceof InvokeStmt` however, we get the following result:

```
<test.Unknown: void aaaa()> calls APIs:
android.telephony.SmsManager
android.telephony.SmsManager
android.telephony.SmsManager
android.content.ContextWrapper
android.content.ContextWrapper
android.widget.Toast
android.widget.Toast
android.widget.Toast
android.content.ContextWrapper
android.content.ContextWrapper
android.widget.Toast
android.widget.Toast
android.widget.Toast
java.lang.Throwable
```

```
<test.Unknown: void bbbb()> calls APIs:
android.app.Activity
android.app.Activity
android.location.LocationManager
android.location.LocationManager
android.location.Location
android.location.Location
```

As we can see, we get a few more API calls (albeit duplicates of the ones already caught the first time) here.