Live API Documentation

Siddharth Subramanian

Laura Inozemtseva

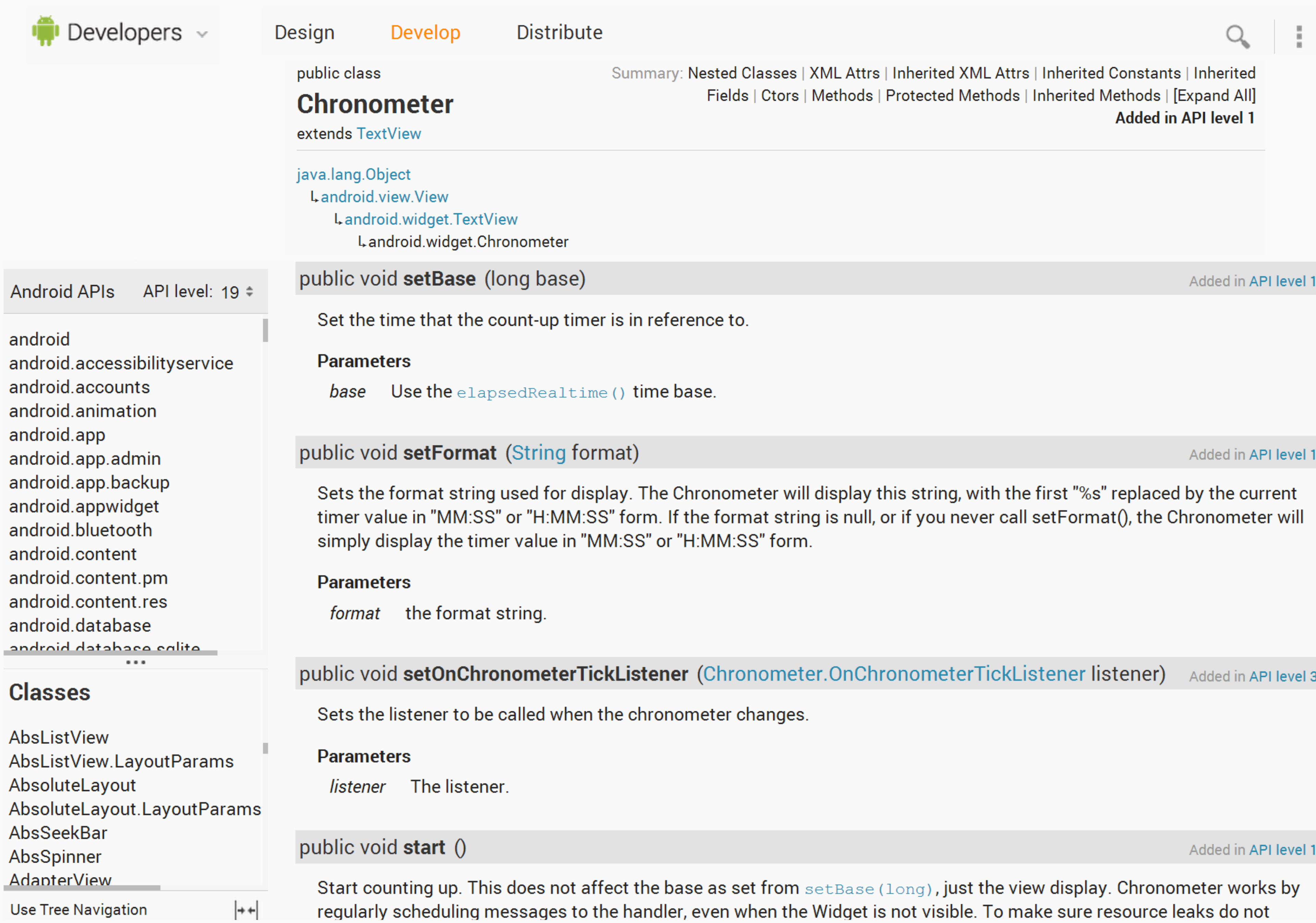
Reid Holmes

UNIVERSITY OF WATERLOO

- **3,463** Classes
- **30,420** Methods

- **21,696** Classes
- **179,408** Methods

- **307** Properties



public class

Chronometer

extends TextView

Summary: Nested Classes | XML Attrs | Inherited XML Attrs | Inherited Constants | Inherited Fields | Ctors | Methods | Protected Methods | Inherited Methods | [Expand All]

Added in API level 1

java.lang.Object
↳android.view.View
↳android.widget.TextView
↳android.widget.Chronometer

Android APIs API level: 19

android
android.accessibilityservice
android.accounts
android.animation
android.app
android.app.admin
android.app.backup
android.appwidget
android.bluetooth
android.content
android.content.pm
android.content.res
android.database
android.database.sqlite

Classes

AbsListView
AbsListView.LayoutParams
AbsoluteLayout
AbsoluteLayout.LayoutParams
AbsSeekBar
AbsSpinner
AdapterView

Use Tree Navigation

public void **setBase** (long base)

Added in API level 1

Set the time that the count-up timer is in reference to.

Parameters

base Use the `elapsedRealtime()` time base.

public void **setFormat** (String format)

Added in API level 1

Sets the format string used for display. The Chronometer will display this string, with the first "%s" replaced by the current timer value in "MM:SS" or "H:MM:SS" form. If the format string is null, or if you never call setFormat(), the Chronometer will simply display the timer value in "MM:SS" or "H:MM:SS" form.

Parameters

format the format string.

public void **setOnChronometerTickListener** (Chronometer.OnChronometerTickListener listener)

Added in API level 3

Sets the listener to be called when the chronometer changes.

Parameters

listener The listener.

public void **start** ()

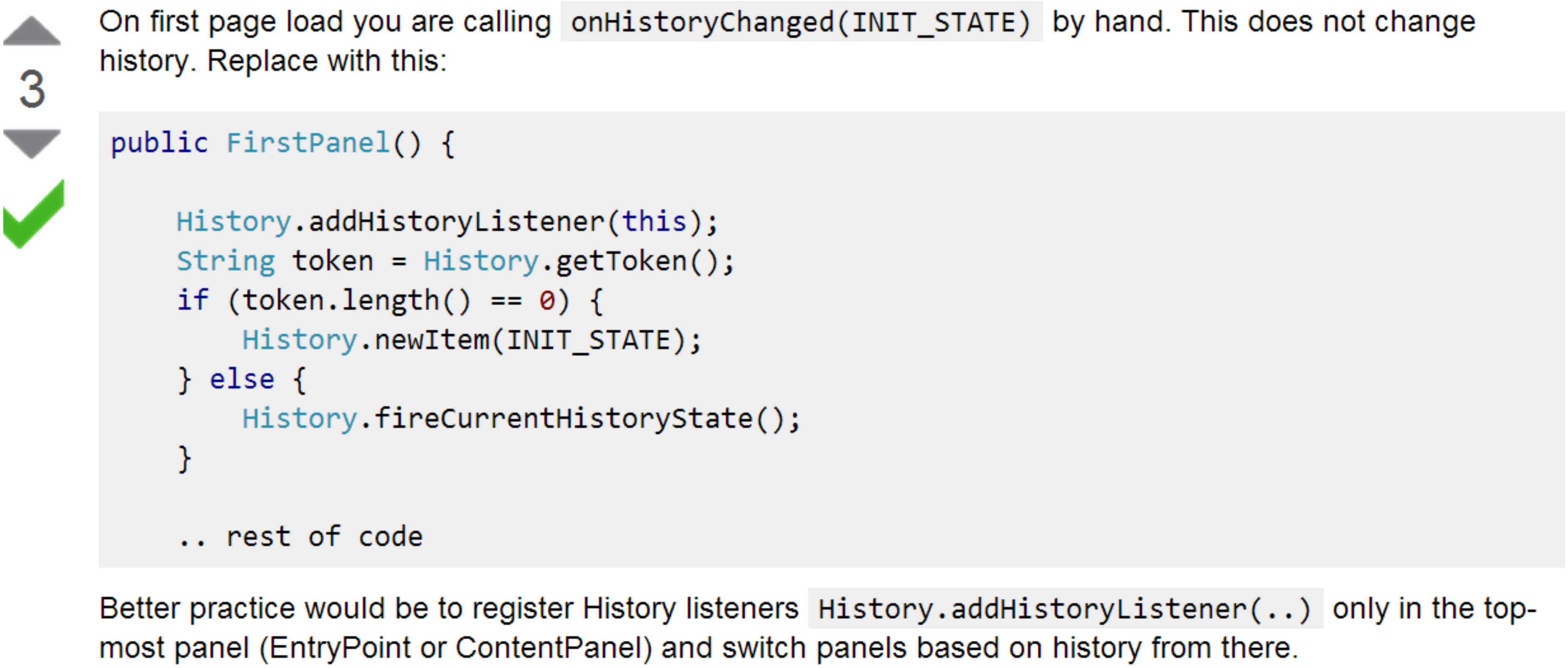
Added in API level 1

Start counting up. This does not affect the base as set from `setBase(long)`, just the view display. Chronometer works by regularly scheduling messages to the handler, even when the Widget is not visible. To make sure resource leaks do not

stackoverflow

- **1.5 million** users and **11 million** posts.
- Questions are answered in a median time of **11 minutes**.

  [Mamykina et al., CHI '11]

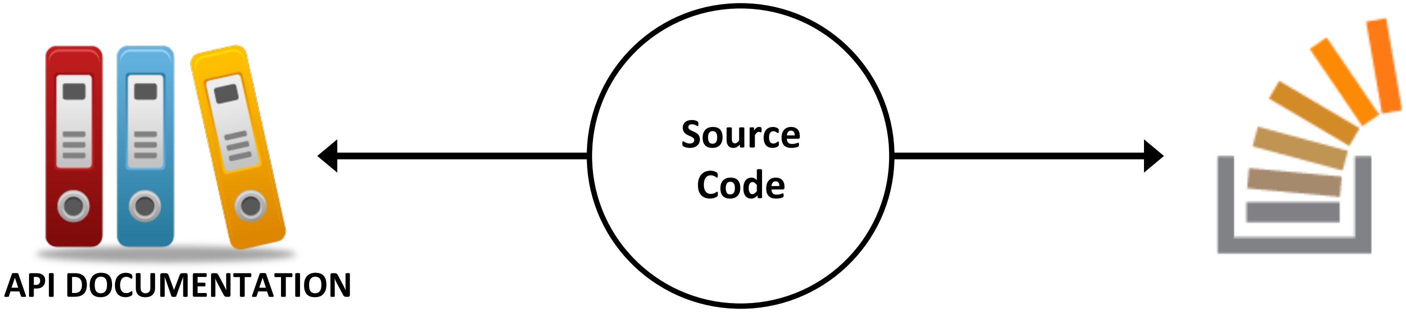


3

On first page load you are calling `onHistoryChanged(INIT_STATE)` by hand. This does not change history. Replace with this:

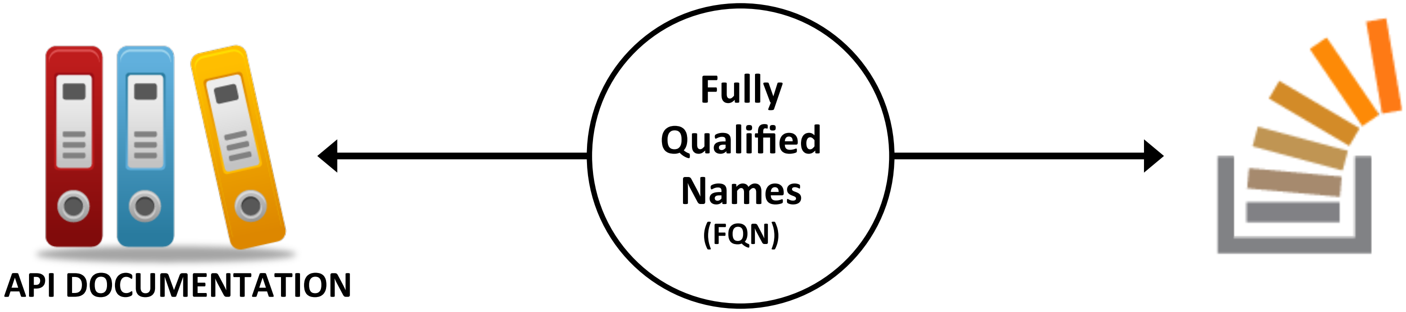
```
public FirstPanel() {

    History.addHistoryListener(this);
    String token = History.getToken();
    if (token.length() == 0) {
        History.newItem(INIT_STATE);
    } else {
        History.fireCurrentHistoryState();
    }

    .. rest of code
```

Better practice would be to register History listeners `History.addHistoryListener(..)` only in the top-most panel (EntryPoint or ContentPanel) and switch panels based on history from there.



API DOCUMENTATION

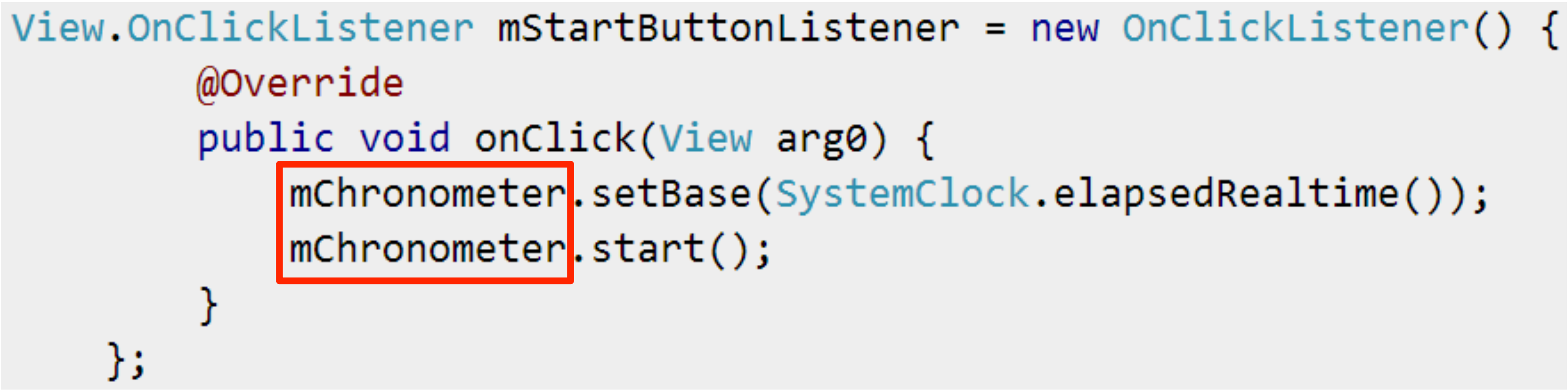
Source Code



API DOCUMENTATION

Fully Qualified Names (FQN)

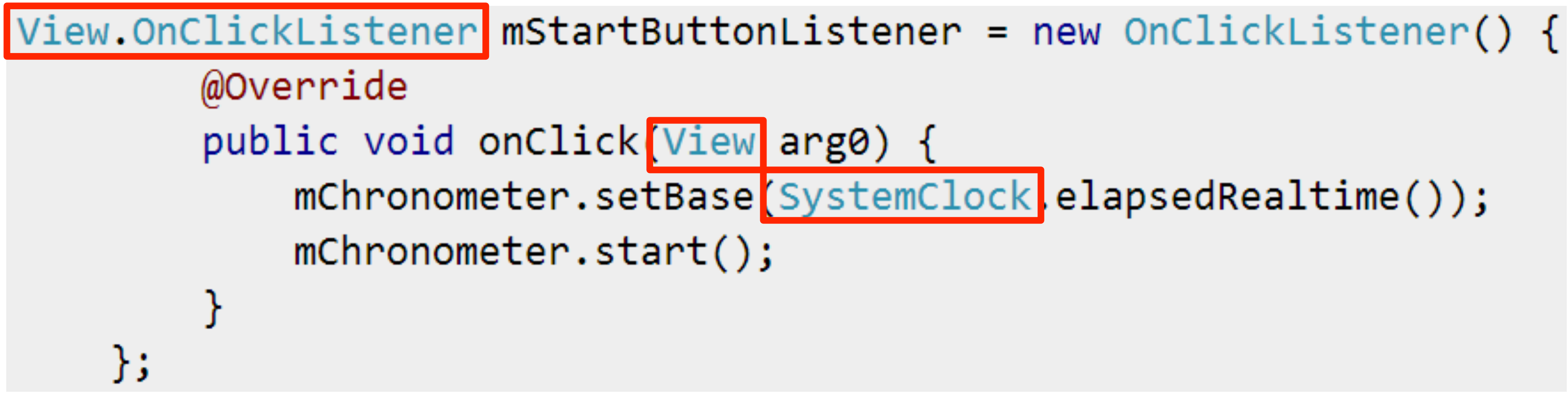
Declaration Ambiguity



```
View.OnClickListener mStartButtonListener = new OnClickListener() {
        @Override
        public void onClick(View arg0) {
            mChronometer.setBase(SystemClock.elapsedRealtime());
            mChronometer.start();
        }
    };
```

Missing declaration statement.

External Reference Ambiguity



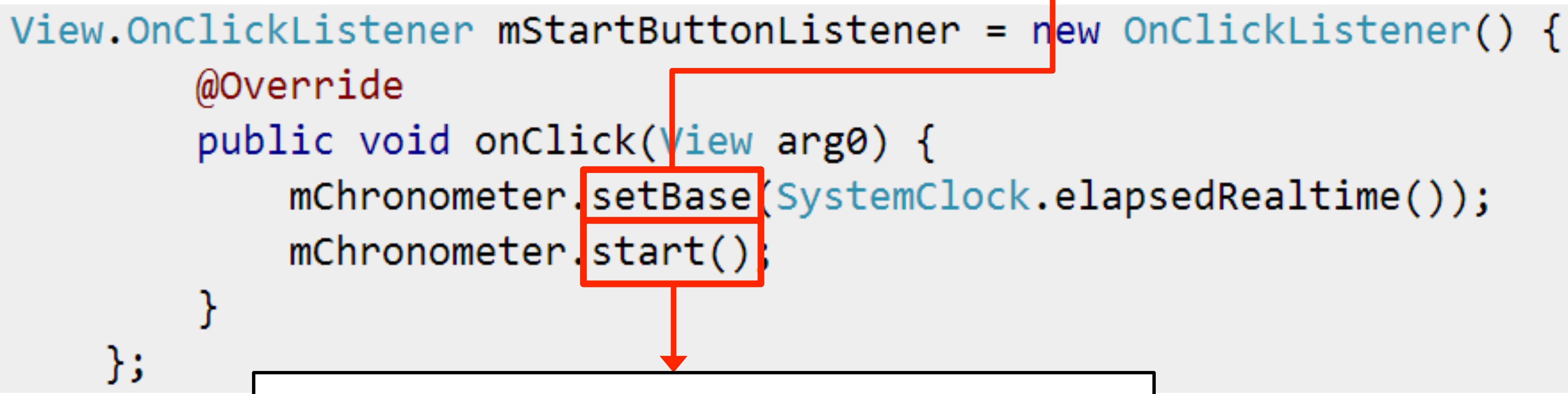
```
View.OnClickListener mStartButtonListener = new OnClickListener() {
        @Override
        public void onClick(View arg0) {
            mChronometer.setBase(SystemClock.elapsedRealtime());
            mChronometer.start();
        }
    };
```

Missing import statements.

Overcoming Ambiguity

895

- org.jfree.chart.axis.LogAxis.setBase(double)
- net.sf.jsog.client.UrlBuilder.setBase(java.lang.String)
- android.widget.Chronometer.setBase(long)
- …



```
View.OnClickListener mStartButtonListener = new OnClickListener() {
        @Override
        public void onClick(View arg0) {
            mChronometer.setBase(SystemClock.elapsedRealtime());
            mChronometer.start();
        }
    };
```

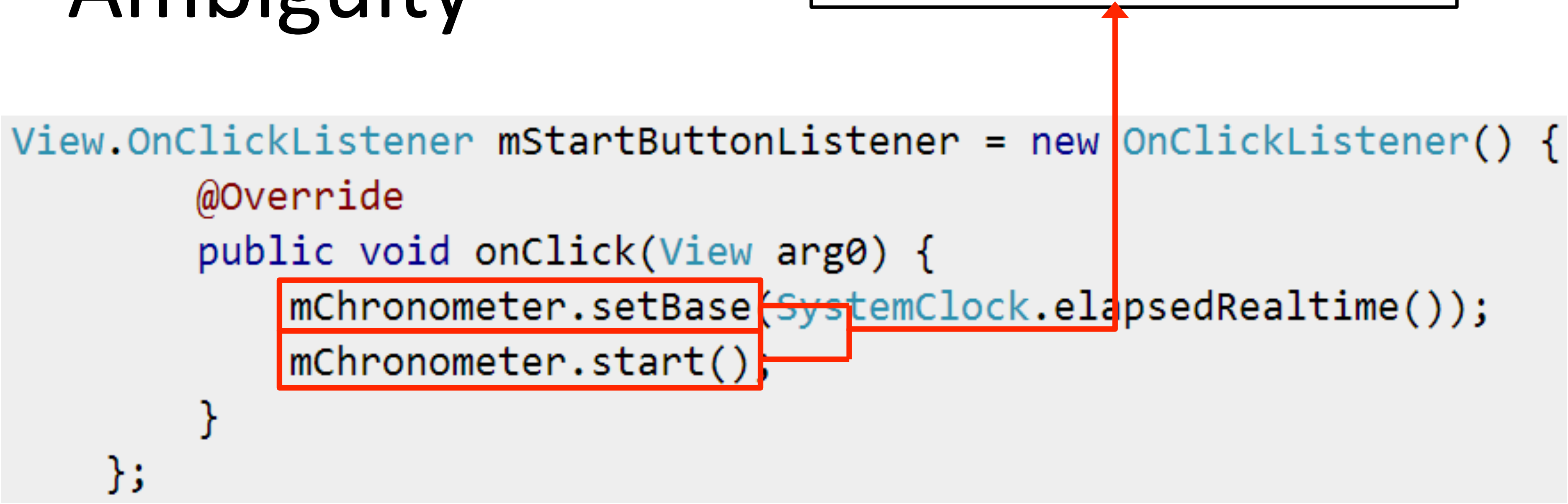
15,352

- java.lang.Thread.start()
- com.mysql.jdbc.util.ServerController.start()
- android.widget.Chronometer.start()
- com.jpattern.shared.util.Chronometer.start()
- …

Overcoming Ambiguity

1

- android.widget.Chronometer



```
View.OnClickListener mStartButtonListener = new OnClickListener() {
        @Override
        public void onClick(View arg0) {
            mChronometer.setBase(SystemClock.elapsedRealtime());
            mChronometer.start();
        }
    };
```

Java :

- Type signatures
- Method signatures
- Inheritance information

JavaScript :

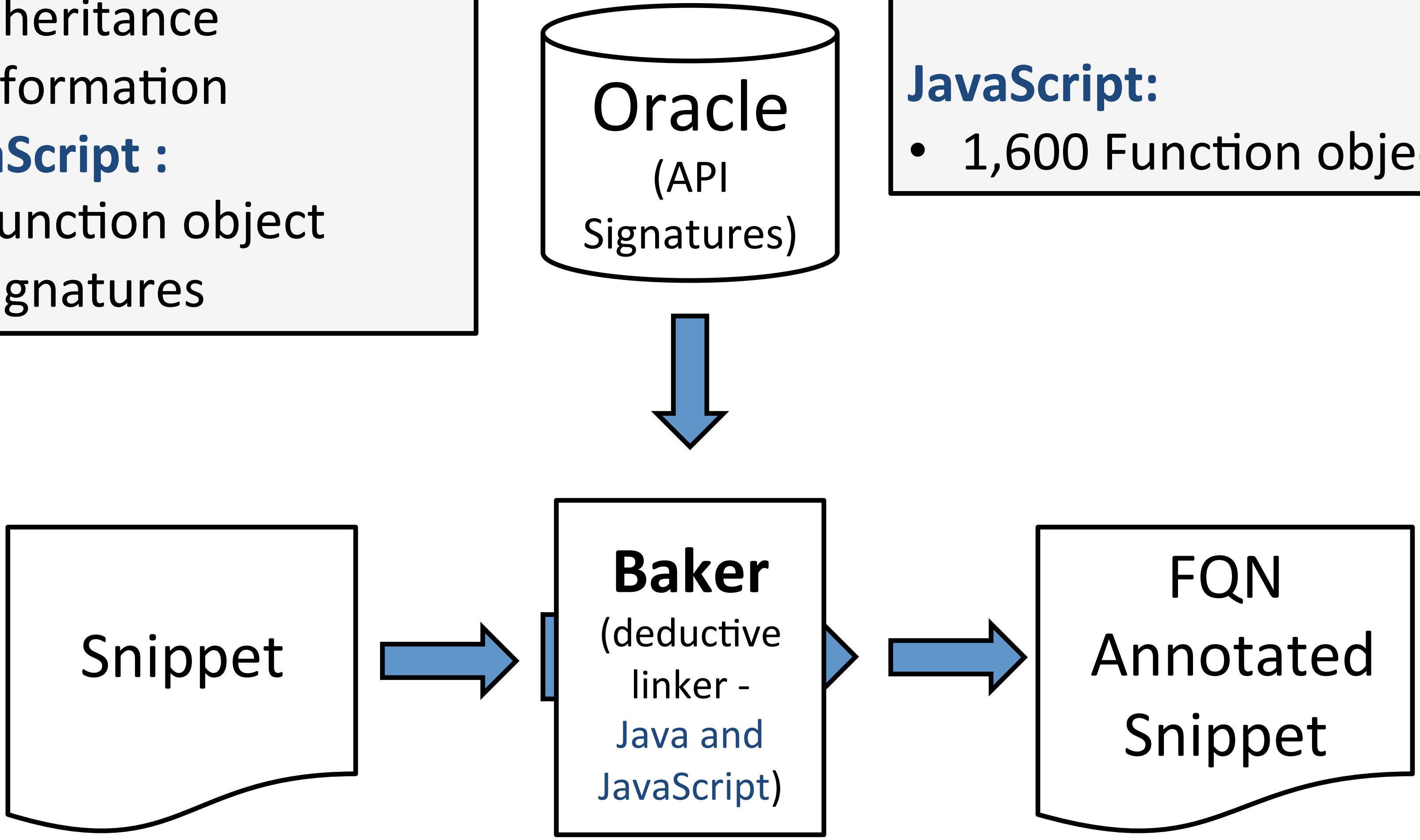
- Function object signatures

Java :

- 1.5 Million types
- 16 Million methods

JavaScript:

- 1,600 Function objects

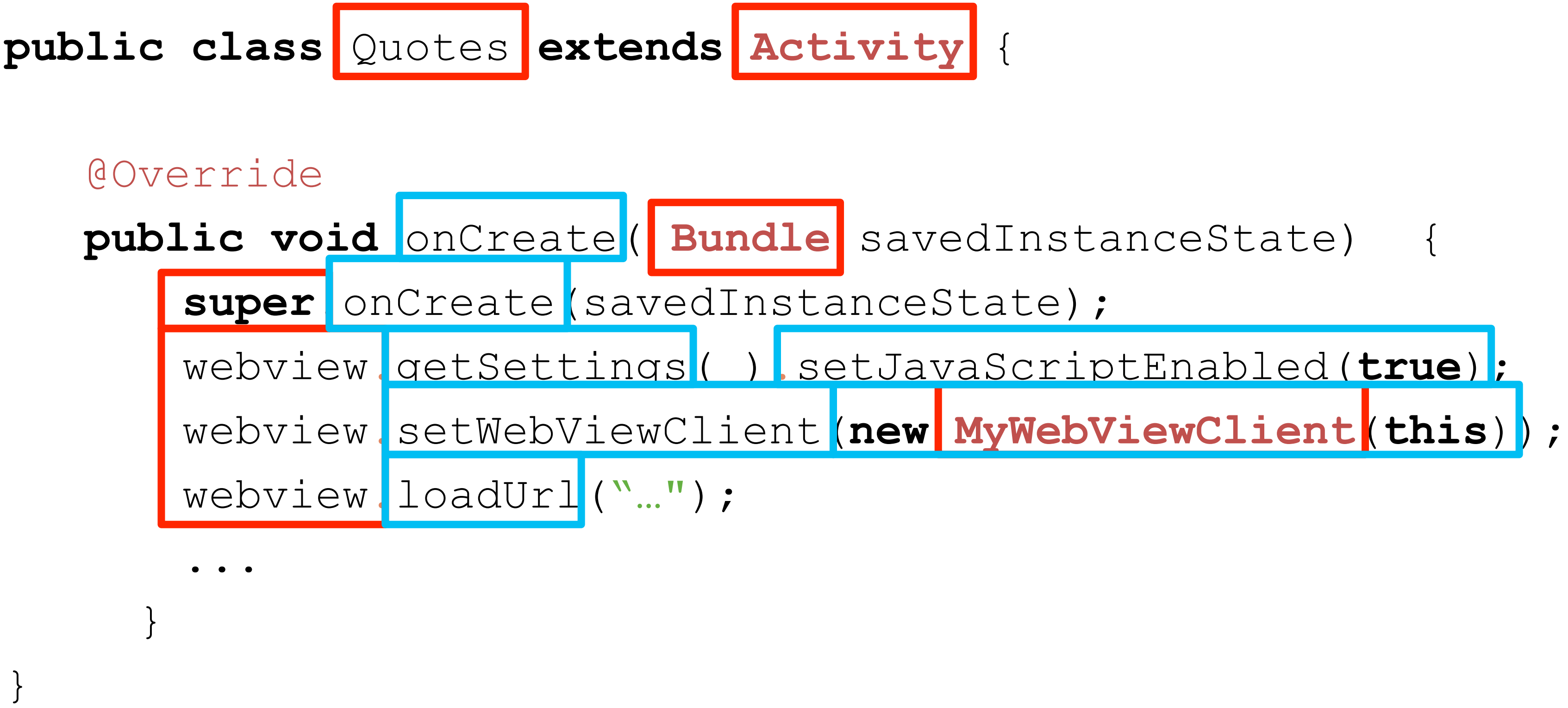


Oracle (API Signatures)

Snippet

Baker (deductive linker - Java and JavaScript)

FQN Annotated Snippet

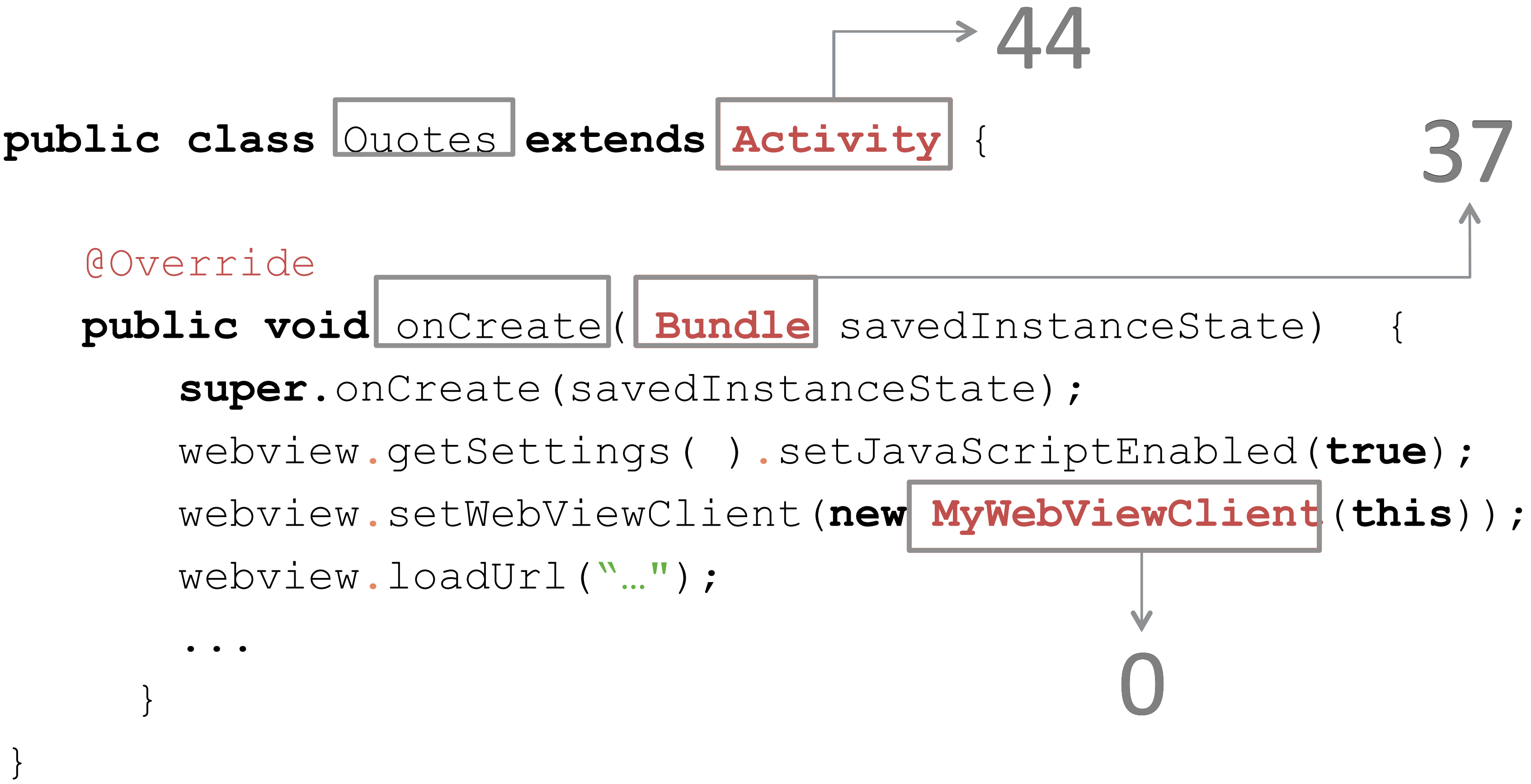


```
public class Quotes extends Activity {

    @Override
    public void onCreate( Bundle savedInstanceState)   {
        super.onCreate(savedInstanceState);
        webview.getSettings( ).setJavaScriptEnabled(true);
        webview.setWebViewClient(new MyWebViewClient(this));
        webview.loadUrl("…");
        ...
      }
}
```

Potential API references in the snippet.

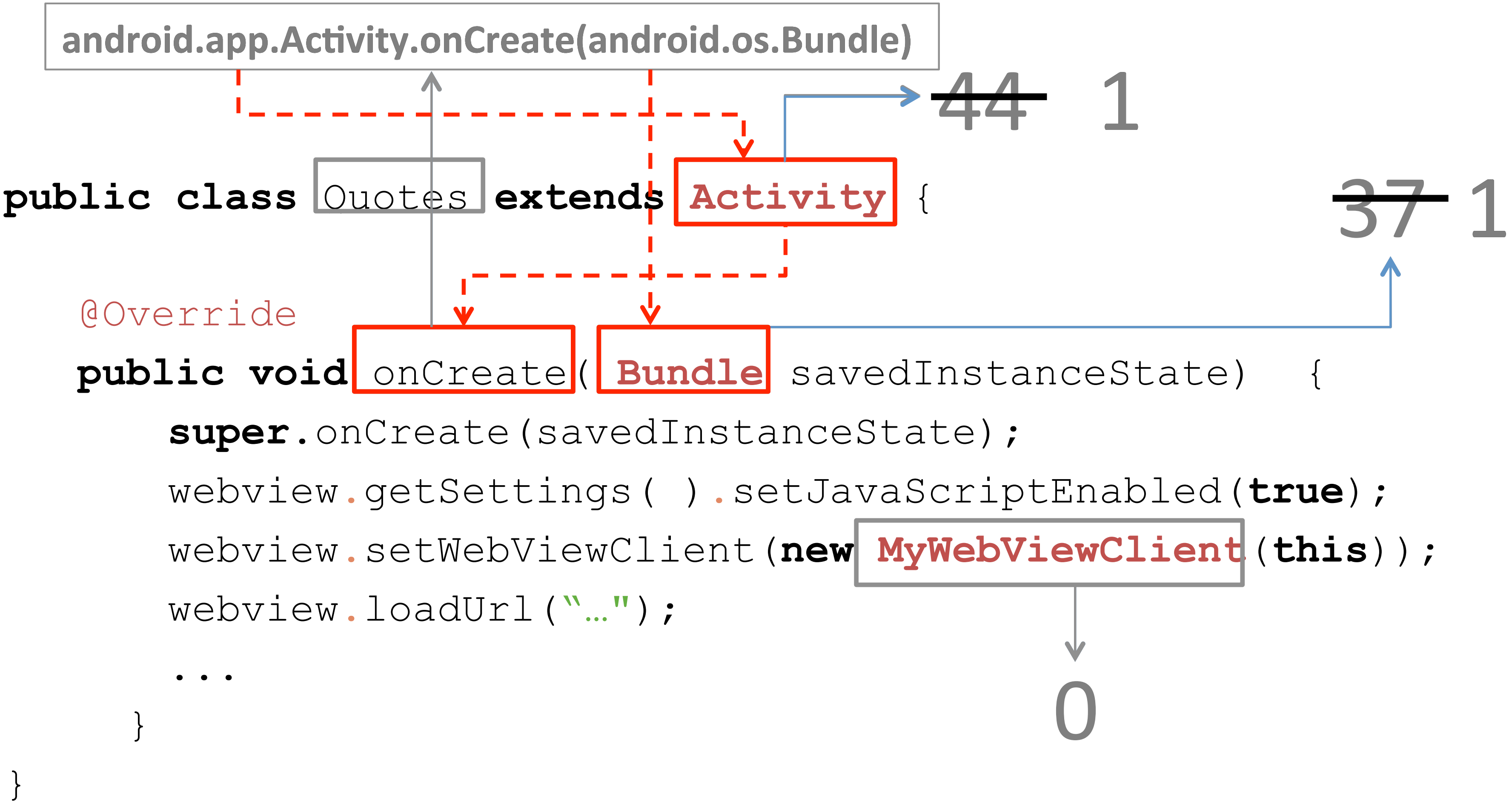
```
public class Quotes extends Activity {

    @Override
    public void onCreate( Bundle savedInstanceState)   {
        super.onCreate(savedInstanceState);
        webview.getSettings( ).setJavaScriptEnabled(true);
        webview.setWebViewClient(new MyWebViewClient(this));
        webview.loadUrl(“…");
        ...
      }
}
```

Identifying local declarations



```
public class Quotes extends Activity {

    @Override
    public void onCreate( Bundle savedInstanceState)   {
        super.onCreate(savedInstanceState);
        webview.getSettings( ).setJavaScriptEnabled(true);
        webview.setWebViewClient(new MyWebViewClient(this));
        webview.loadUrl("…");
        ...
    }
}
```

Inferring types



android.app.Activity.onCreate(android.os.Bundle)

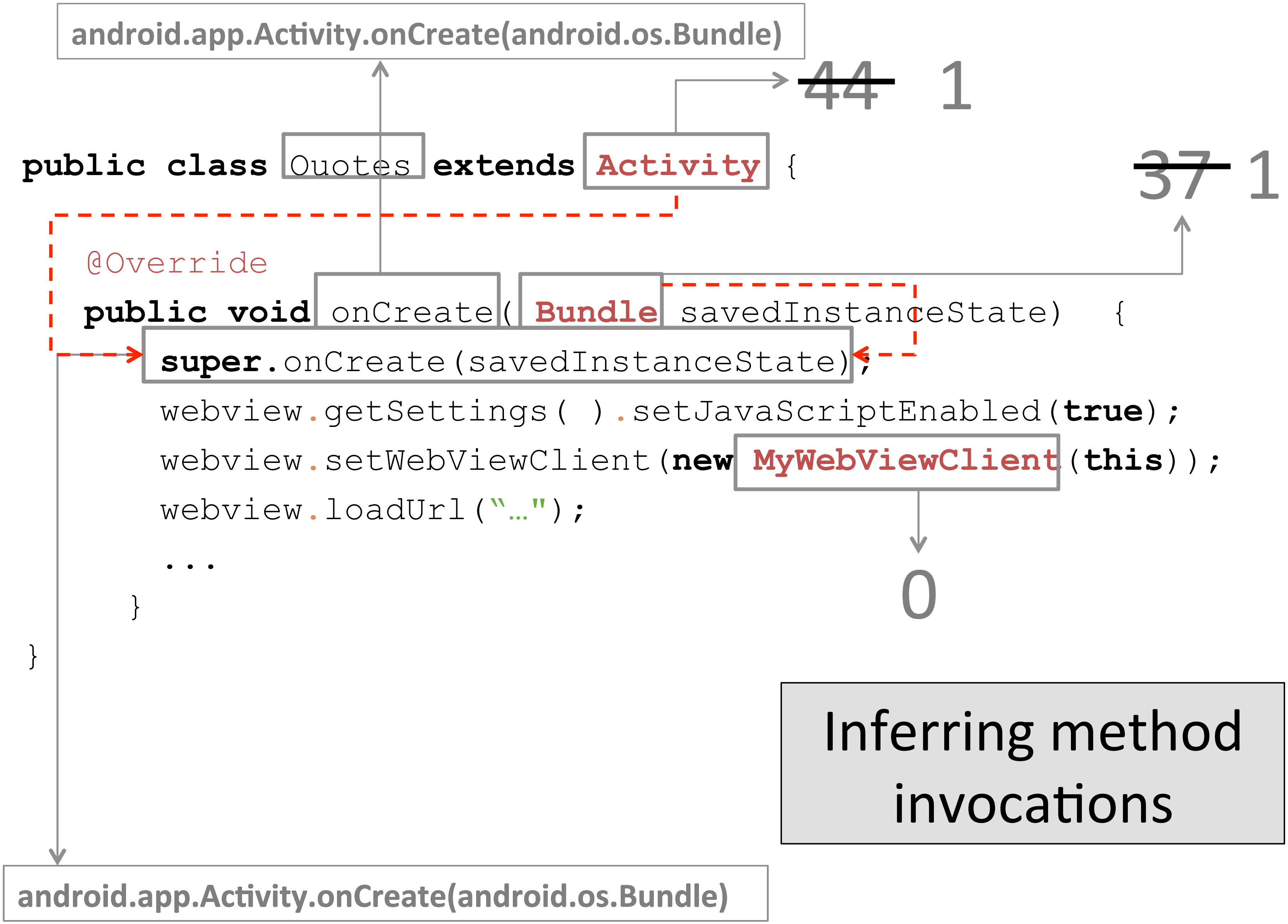
~~44~~ 1

~~37~~ 1

0

```
public class Quotes extends Activity {

   @Override
   public void onCreate( Bundle savedInstanceState)   {
       super.onCreate(savedInstanceState);
       webview.getSettings( ).setJavaScriptEnabled(true);
       webview.setWebViewClient(new MyWebViewClient(this));
       webview.loadUrl("…");
       ...
     }
}
```

Inferring overridden methods



android.app.Activity.onCreate(android.os.Bundle)

~~44~~ 1

~~37~~ 1

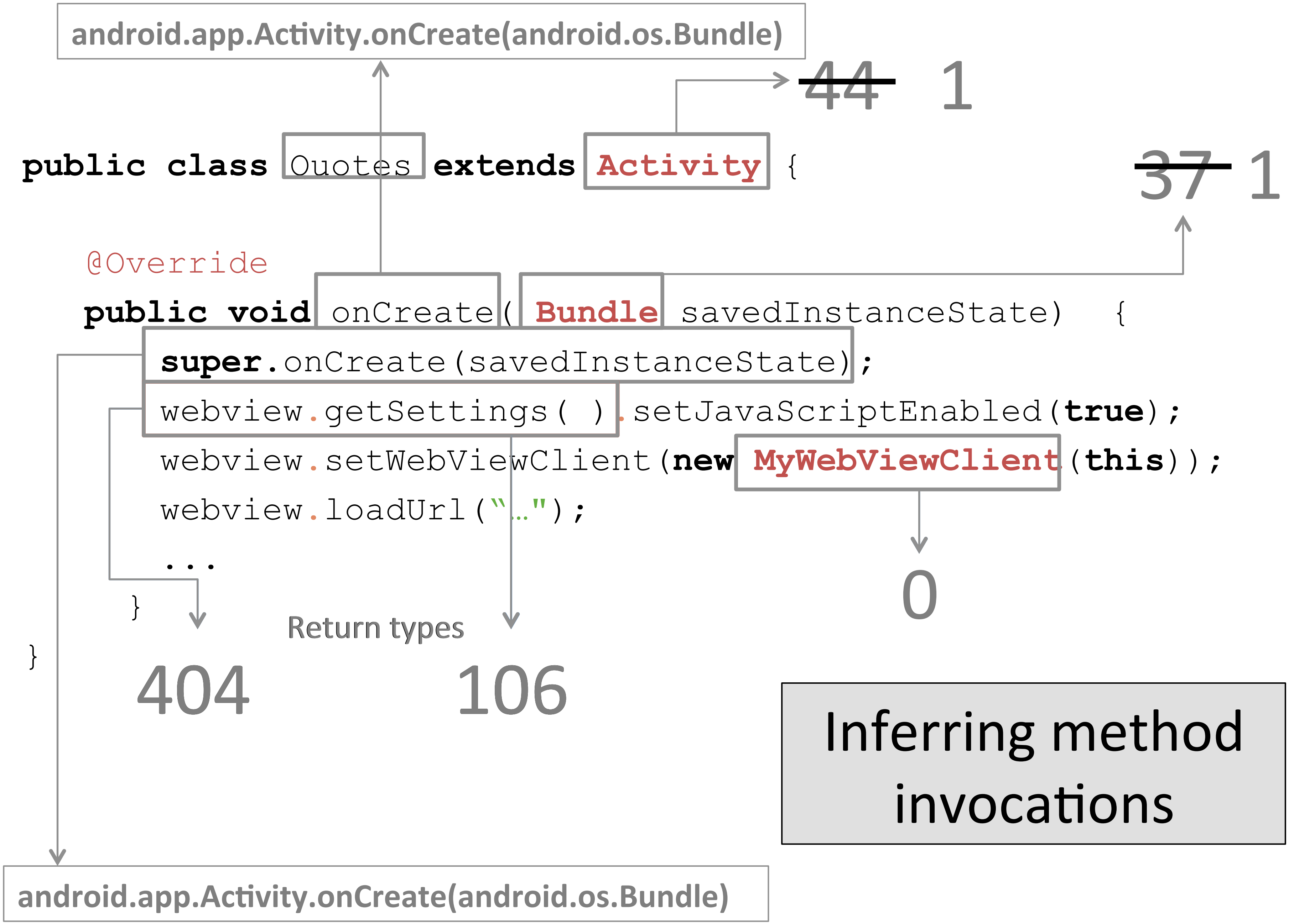
```
public class Quotes extends Activity {

  @Override
  public void onCreate( Bundle savedInstanceState)   {
    super.onCreate(savedInstanceState);
    webview.getSettings( ).setJavaScriptEnabled(true);
    webview.setWebViewClient(new MyWebViewClient(this));
    webview.loadUrl(“…");
    ...
  }
}
```

0

android.app.Activity.onCreate(android.os.Bundle)

Inferring method invocations

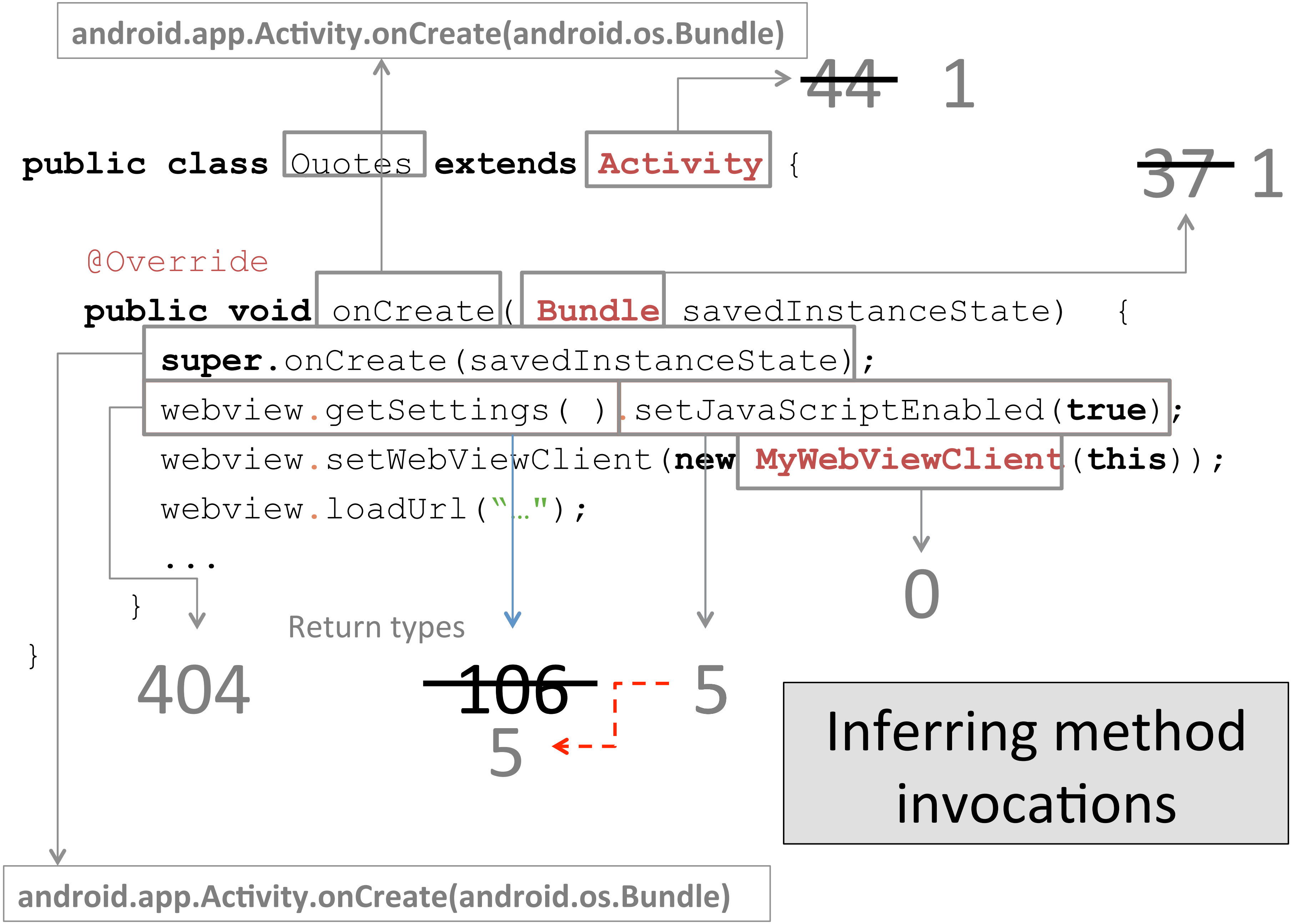
Inferring method invocations



```
public class Quotes extends Activity {

  @Override
  public void onCreate( Bundle savedInstanceState)   {
    super.onCreate(savedInstanceState);
    webview.getSettings( ).setJavaScriptEnabled(true);
    webview.setWebViewClient(new MyWebViewClient(this));
    webview.loadUrl("…");
    ...
  }
}
```

android.app.Activity.onCreate(android.os.Bundle)
~~44~~ 1
~~37~~ 1
0
Return types
404
106
android.app.Activity.onCreate(android.os.Bundle)

Inferring method invocations



android.app.Activity.onCreate(android.os.Bundle)

~~44~~ 1

~~37~~ 1

```
public class Quotes extends Activity {

  @Override
  public void onCreate( Bundle savedInstanceState)  {
    super.onCreate(savedInstanceState);
    webview.getSettings( ).setJavaScriptEnabled(true);
    webview.setWebViewClient(new MyWebViewClient(this));
    webview.loadUrl("…");
    ...
  }
}
```

0

Return types

404

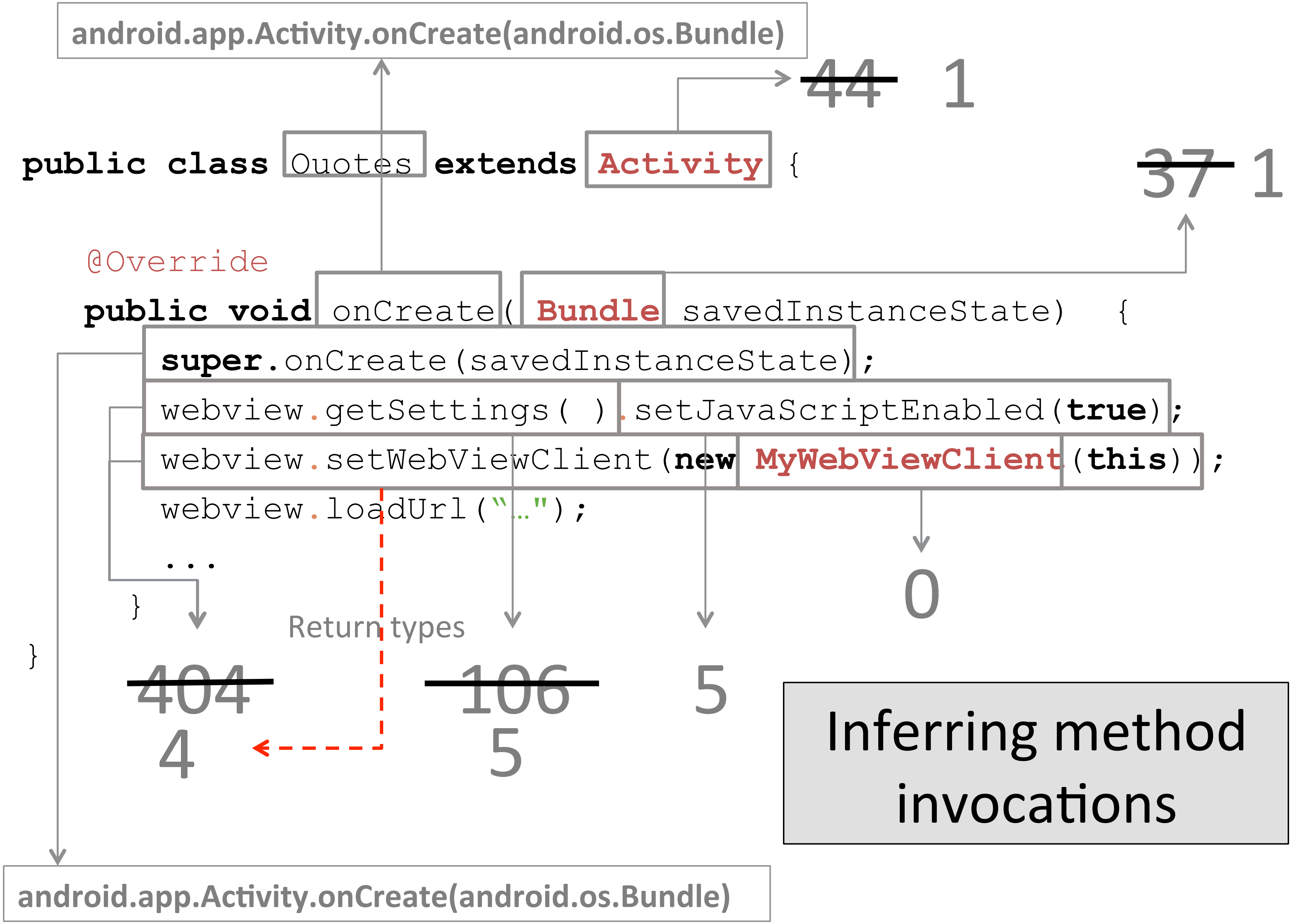
~~106~~

5

5

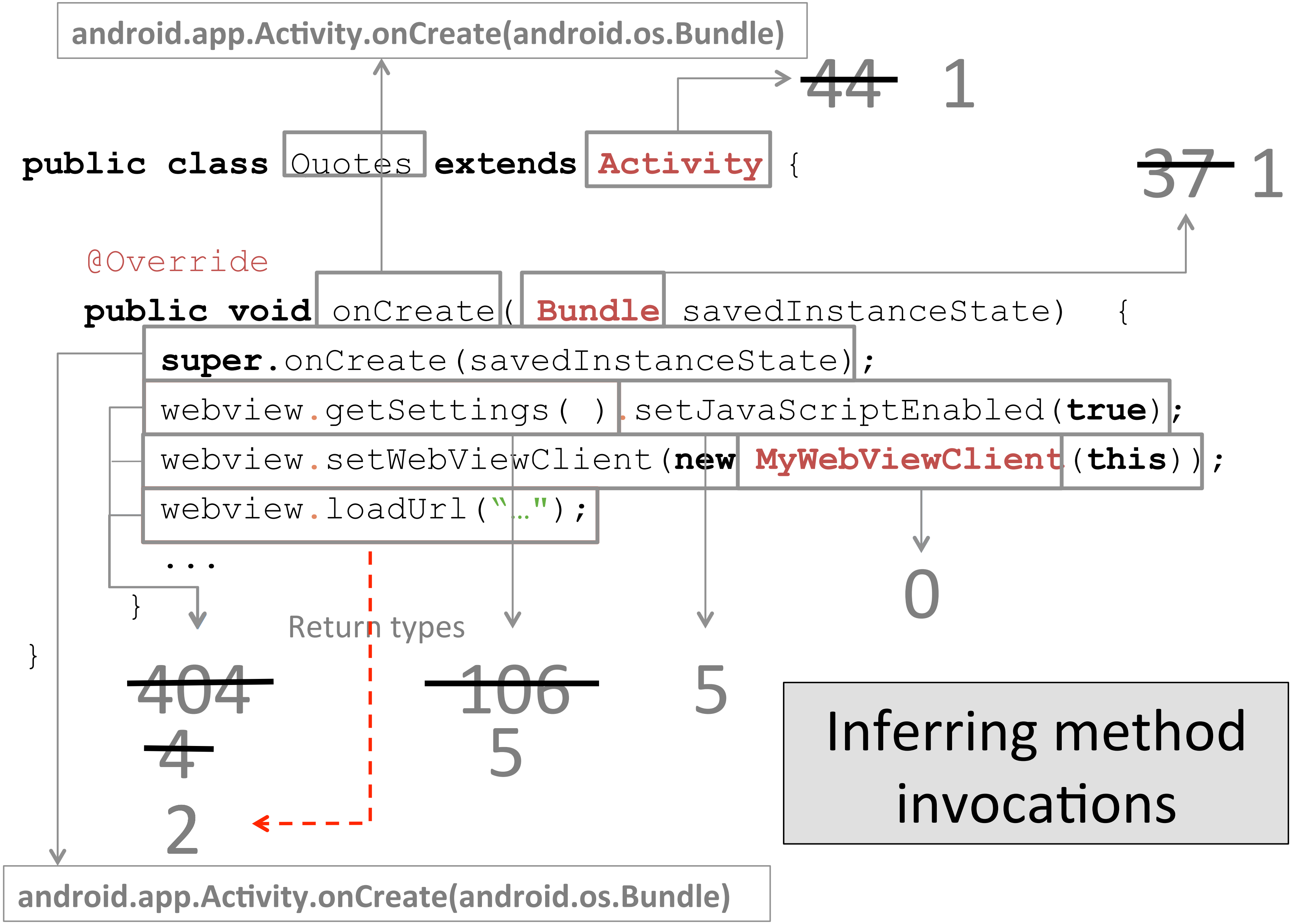
android.app.Activity.onCreate(android.os.Bundle)

Inferring method invocations



```
public class Quotes extends Activity {

  @Override
  public void onCreate( Bundle savedInstanceState)   {
    super.onCreate(savedInstanceState);
    webview.getSettings( ).setJavaScriptEnabled(true);
    webview.setWebViewClient(new MyWebViewClient(this));
    webview.loadUrl("…");
    ...
  }
}
```

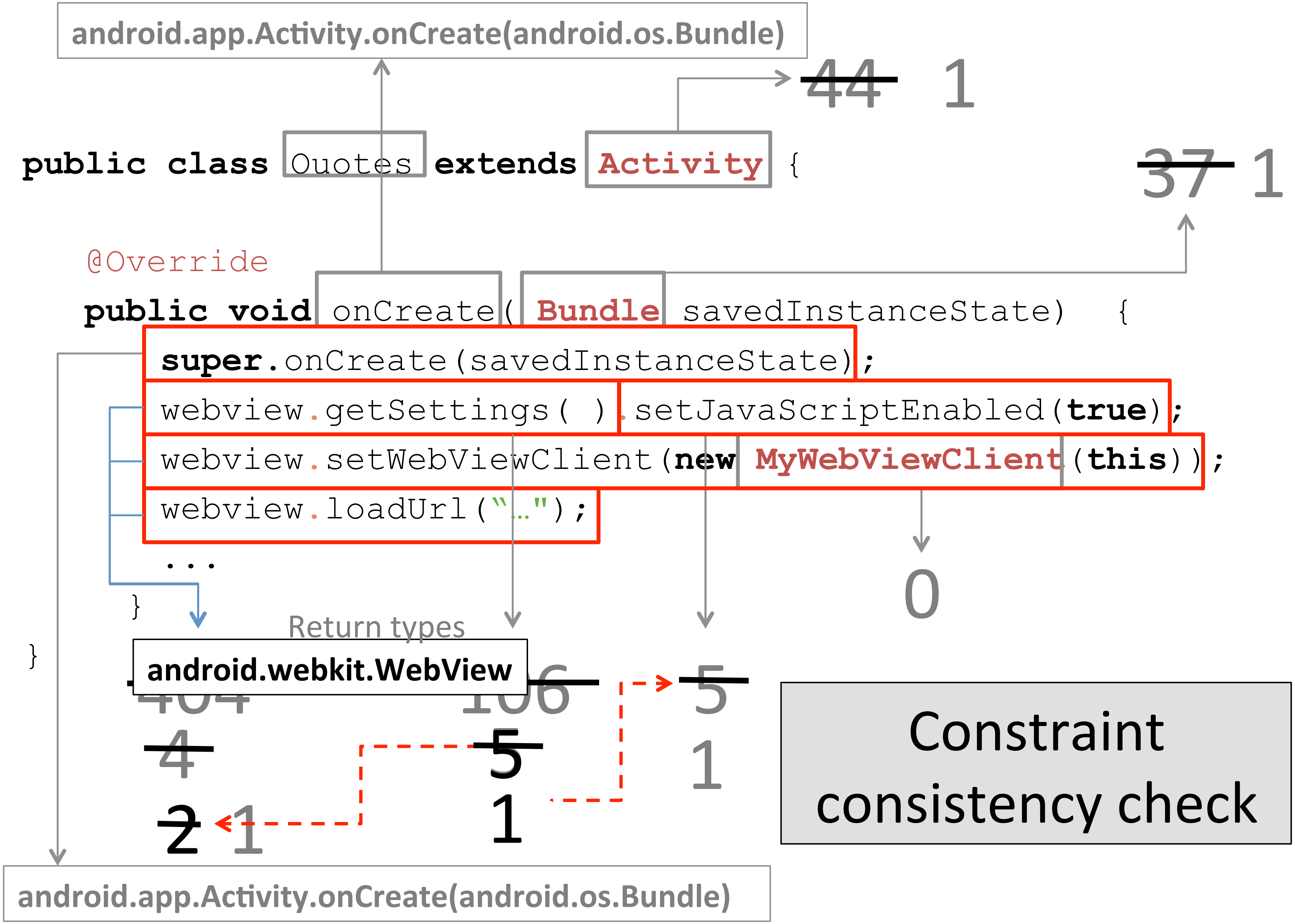
android.app.Activity.onCreate(android.os.Bundle)
~~44~~ 1
~~37~~ 1
0
Return types
~~404~~ 4
~~106~~ 5
5
android.app.Activity.onCreate(android.os.Bundle)



```
public class Quotes extends Activity {

    @Override
    public void onCreate( Bundle savedInstanceState)  {
        super.onCreate(savedInstanceState);
        webview.getSettings( ).setJavaScriptEnabled(true);
        webview.setWebViewClient(new MyWebViewClient(this));
        webview.loadUrl("…");
        ...
    }
}
```

android.app.Activity.onCreate(android.os.Bundle)
~~44~~ 1
~~37~~ 1
0
Return types
~~404~~
~~4~~
2
~~106~~
5
5
android.app.Activity.onCreate(android.os.Bundle)

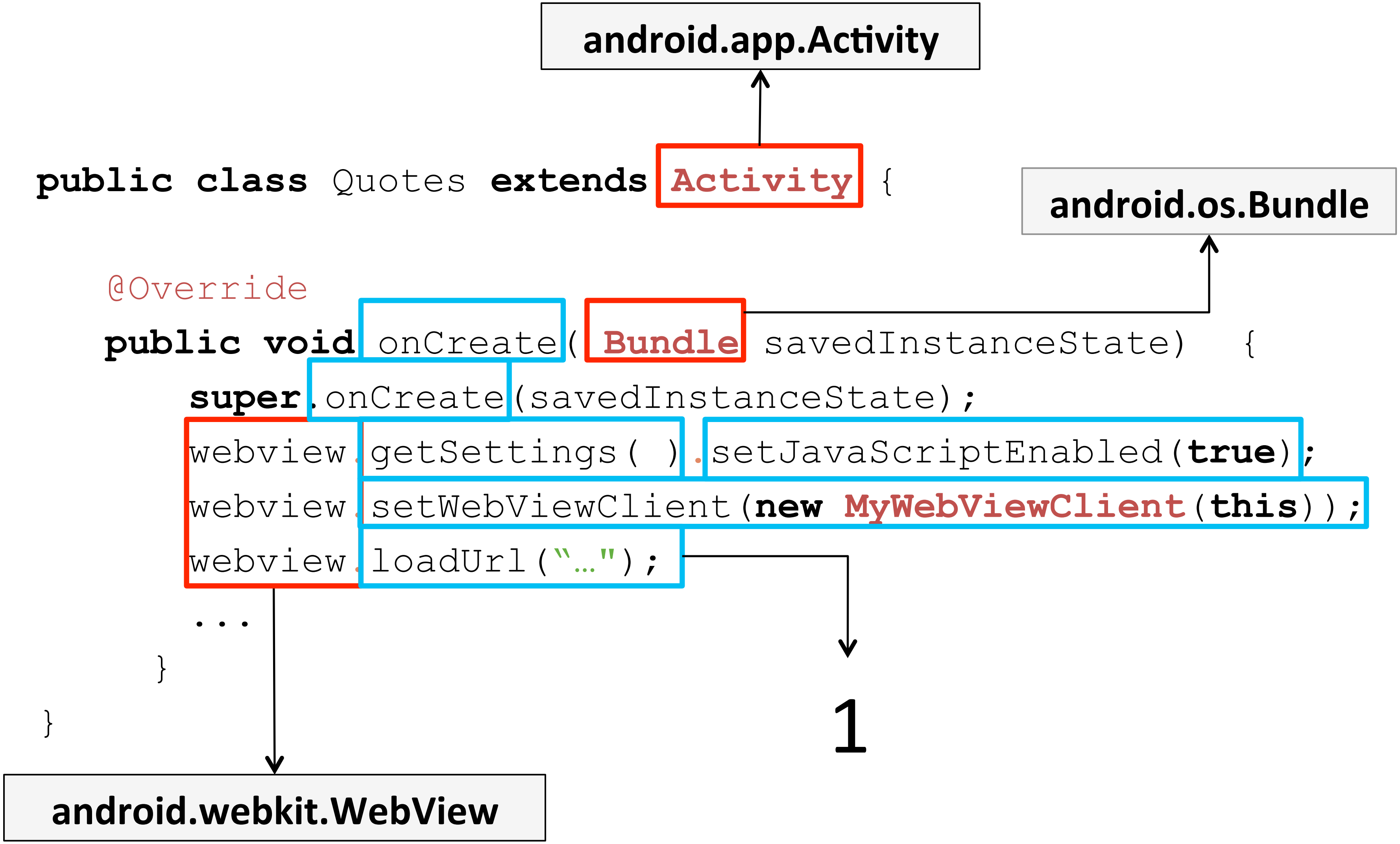
Inferring method invocations



```
public class Quotes extends Activity {

  @Override
  public void onCreate( Bundle savedInstanceState)   {
    super.onCreate(savedInstanceState);
    webview.getSettings( ).setJavaScriptEnabled(true);
    webview.setWebViewClient(new MyWebViewClient(this));
    webview.loadUrl("…");
    ...
  }
}
```

android.app.Activity.onCreate(android.os.Bundle)
~~44~~ 1
~~37~~ 1
0
Return types
android.webkit.WebView
~~404~~
~~4~~
~~2~~ 1
~~106~~
~~5~~
1
~~5~~
1
android.app.Activity.onCreate(android.os.Bundle)

Constraint consistency check



android.app.Activity

android.os.Bundle

```
public class Quotes extends Activity {

    @Override
    public void onCreate( Bundle savedInstanceState)   {
        super.onCreate(savedInstanceState);
        webview.getSettings( ).setJavaScriptEnabled(true);
        webview.setWebViewClient(new MyWebViewClient(this));
        webview.loadUrl("…");
        ...
    }
}
```

1

android.webkit.WebView

Baker Accuracy : Java

Precision: 0.98 Recall: 0.83

System	Unique			Relaxed
	True Positives	False Positives	False Negatives	False Negatives
Android	40	1	9	1
GWT	43	0	7	0
Hibernate	37	0	13	0
Joda Time	44	3	3	0
Xstream	40	0	10	0
Total	**204**	**4**	**42**	**1**

An API reference is a

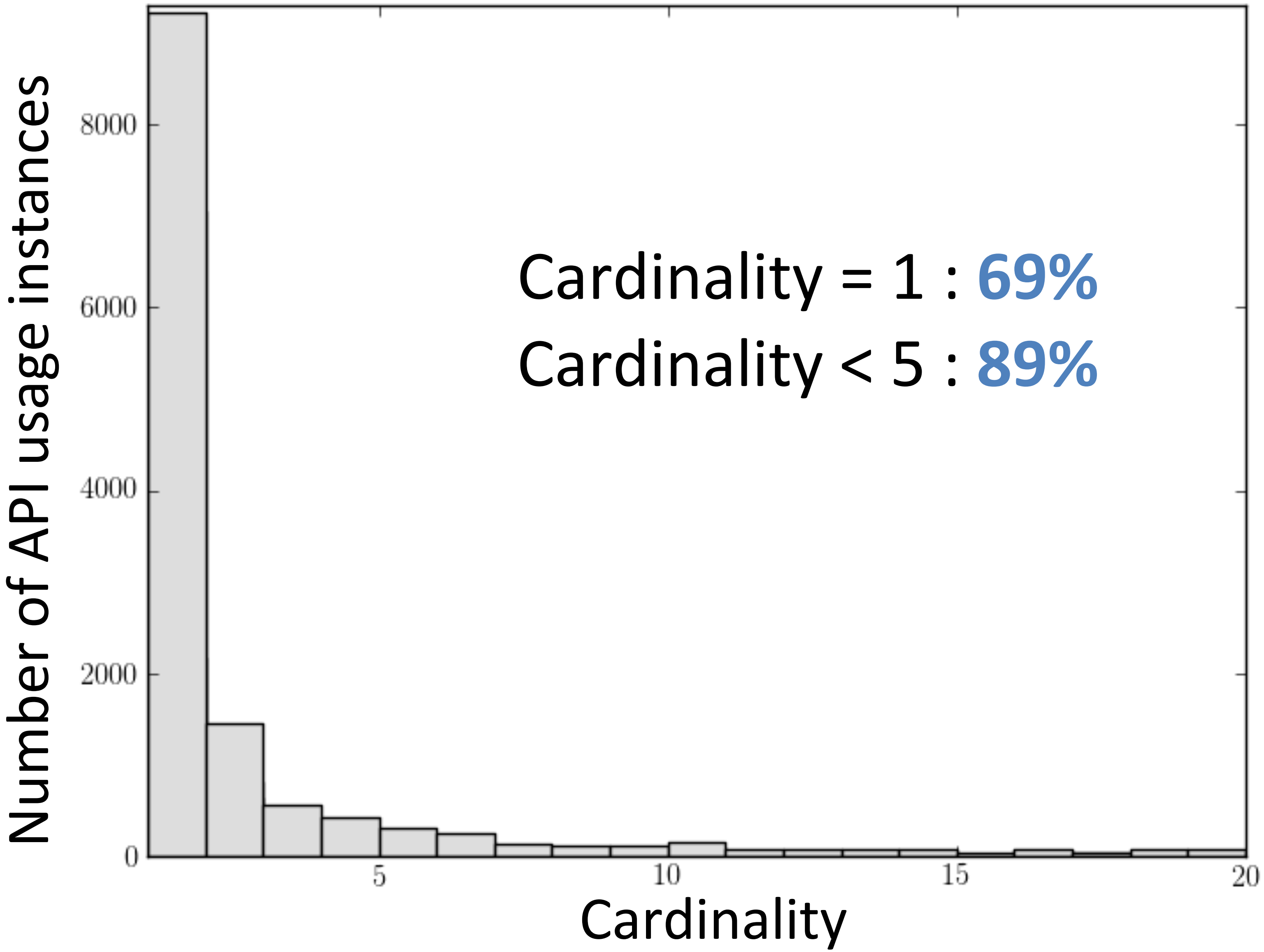
- Unique
  - True Positive: Baker correctly identified the target API.
  - False Positive: Baker incorrectly identified a target API.
  - False Negative: Baker either could not correctly identify the right target API or it produced an imprecise match.
- **Relaxed**
  - False Negative: Baker could not identify the target API even imprecisely.

Baker Accuracy : JavaScript

Precision: 0.97 Recall: 0.96

System	Unique			Relaxed
	True Positives	False Positives	False Negatives	False Negatives
JSCore/DOM	48	2	0	0
JQuery	47	2	1	0
Phonegap	46	2	2	0
Webworks	43	0	7	2
Total	**184**	**6**	**10**	**2**

Cardinality Distribution



Cardinality = 1 : **69%**

Cardinality < 5 : **89%**

Imprecise Matches

Imprecise matches in the `AbstractDocumentReader` class belonging to the xstream Java library:

- com.thoughtworks.xstream.io.xml.AbstractDocumentReader
- com.cloudbees.shaded.thoughtworks.xstream.io.xml.AbstractDocumentReader
- com.ovea.jetty.session.internal.xstream.io.xml.AbstractDocumentReader
- cucumber.runtime.xstream.io.xml.AbstractDocumentReader
- org.pitest.xstream.io.xml.AbstractDocumentReader

Enabling Live API Documentation (Demo)

Querying Baker

```
$  cat JavaSnippet.txt
   final Chronometer chrono =
         (Chronometer)findViewById(R.id.chronometer);
   chrono.setBase(SystemClock.elapsedRealTime());
   chrono.start();

$  curl --data "file = @JavaSnippet.txt"
   http://gadget.cs.uwaterloo.ca/Baker/getAPI.php
  {
        "api_elements" : [
                        ...
               {
                  "precision" : "1",
                  "name" : "setBase",
                  "line_number"  : "2",
                  "type" : "api_method",
                  "elements" : [
                      "android.widget.Chronometer.setBase(long)"
                    ]
                }
                        ...
     ]}
```

Summary

- **Deductive Linking** can identify API references in source code snippets with high precision.
- This helps enable "**Live Documentation**" by bridging interactive Stack Overflow posts with authoritative API documentation.
- Baker and related resources can be accessed at: https://cs.uwaterloo.ca/~rtholmes/baker

References

- **[1]** Mamykina, Lena and Manoim, Bella and Mittal, Manas and Hripcsak, George and Hartmann, Bjorn. Design Lessons from the Fastest Q&A Site in the West.

Additional Info

stackoverflow

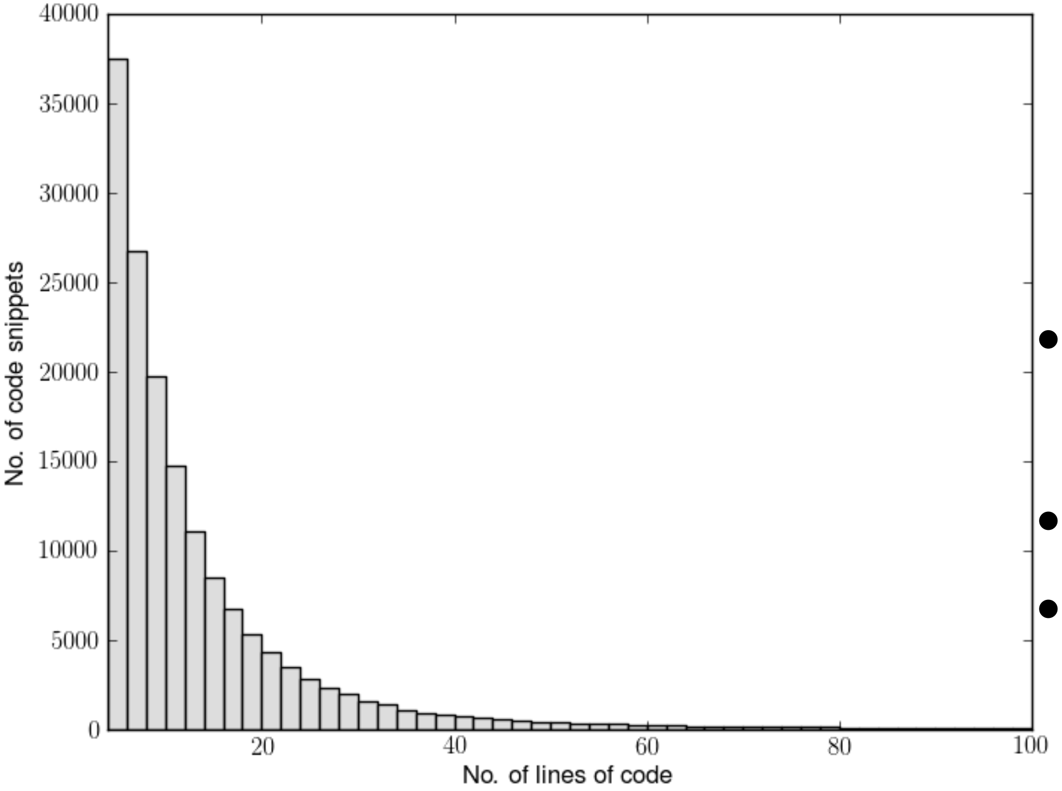
Java

- 277 K posts
- 110 K accepted answers
- 69 K code snippets (in accepted answers)

JS

- 241 K posts
- 121 K accepted answers
- 89 K code snippets (in accepted answers)

Challenges



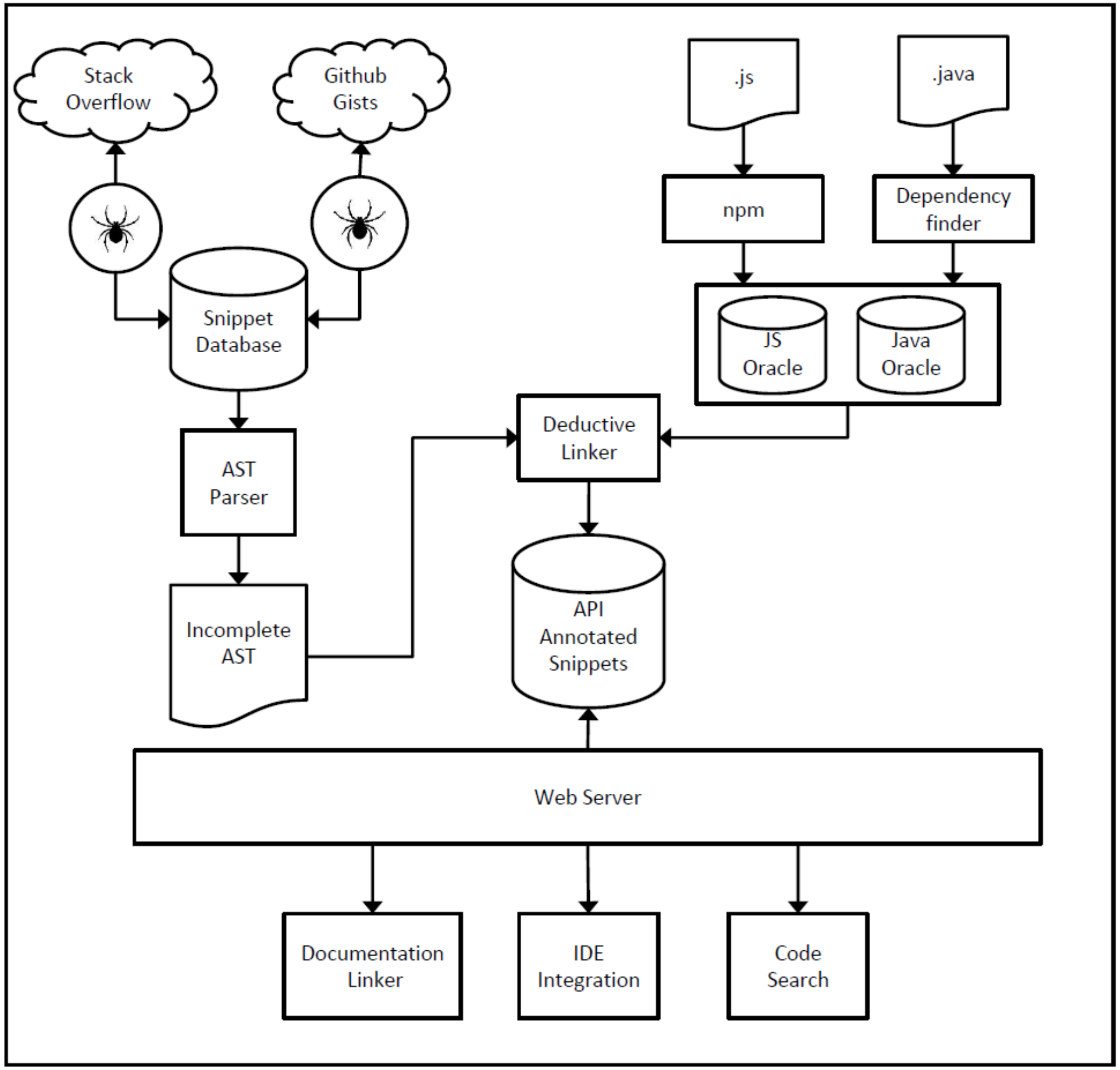
In **accepted answers** tagged Java/JavaScript:

- No. of snippets > 3 LOC : **159, 531**
- Median size : **9 LOC**
- Mean size : **14.33 LOC**

Method Name	No. of Collisions	No. of Packages
onCreate	32	15
show	17	2
getResources	9	6
getContentResolver	4	2
getItemId	7	2
getAction	6	3
findViewById	5	3
getString	34	12
getData	12	9
close	137	36
Average	**26.3**	**9.0**

Naming Collisions among the 10 most-used Android methods.

Architecture



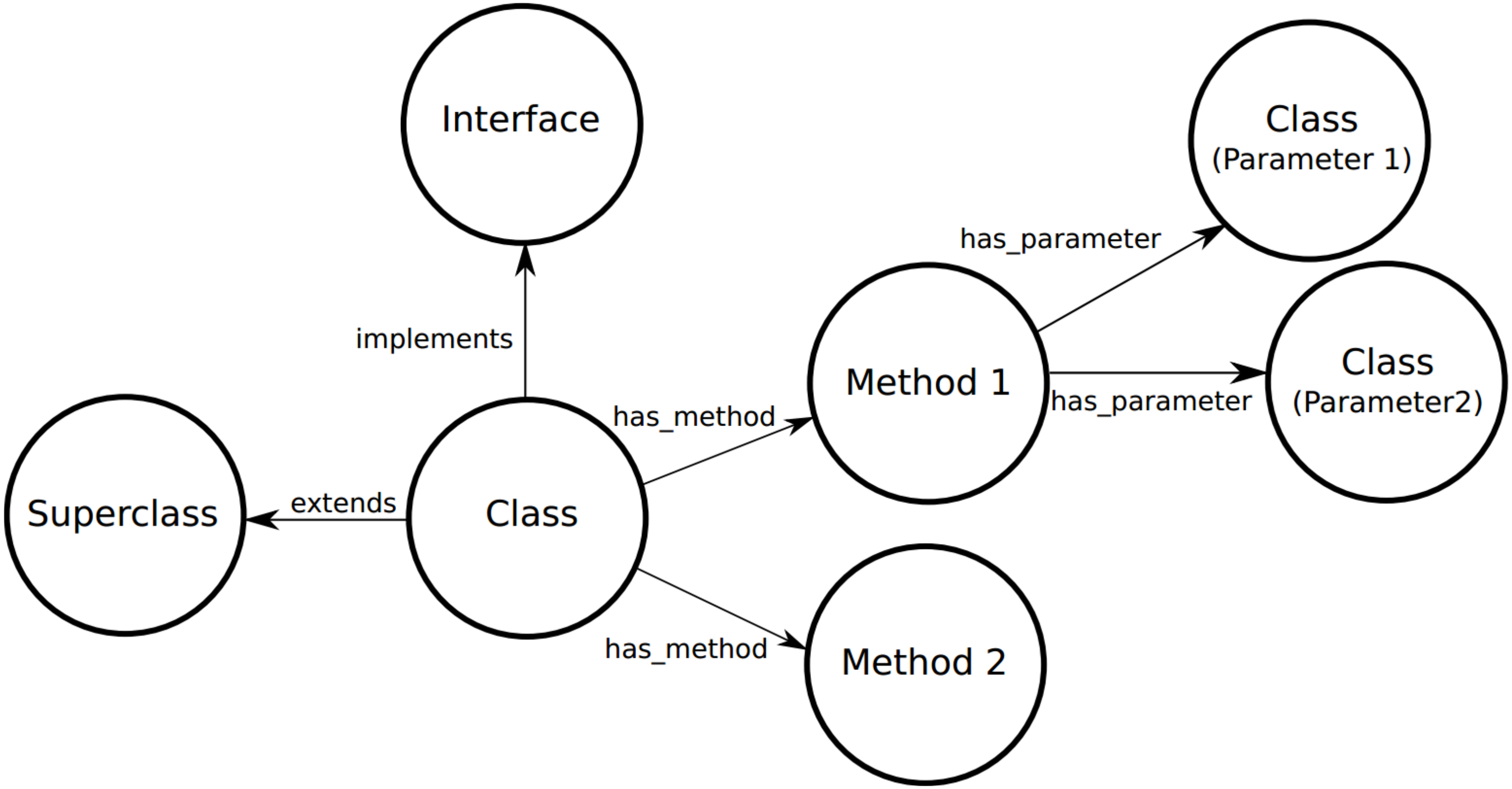
Java :

- 1.5 Million Classes
- 16 Million Methods

Oracle

JavaScript:

- 1,600 Properties



Evaluation

- **RQ1:** Does Baker accurately identify fully-qualified API references in code snippets?
- **RQ2:** Is Baker robust and able to resolve references on a variety of libraries?

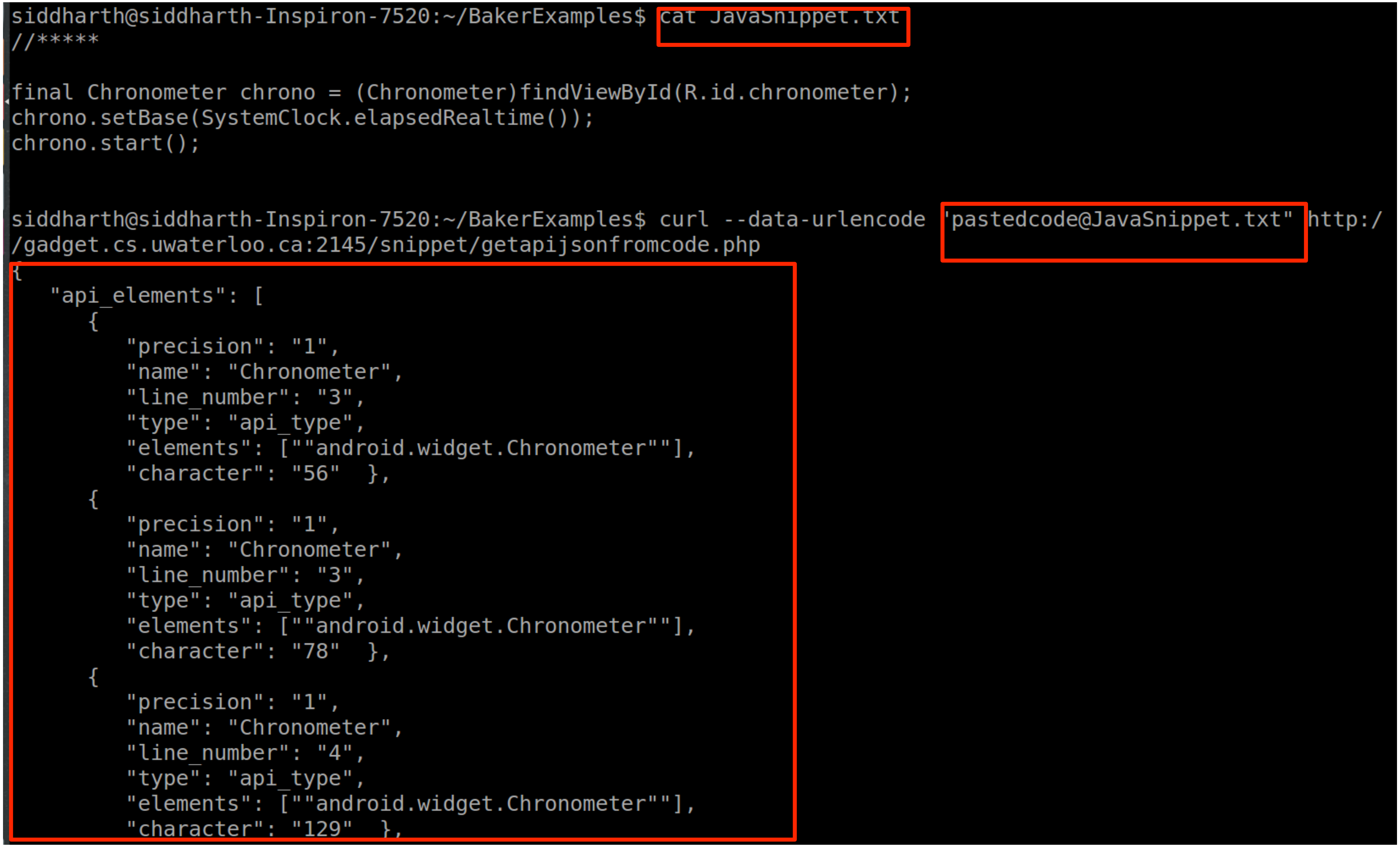
Example Diversity : Java

System	No. of Classes (Total/Unique)	No. of Methods (Total/Unique)
Android	272 / 64	175 / 104
Apache	178 / 79	108 / 97
Eclipse	104 / 41	53 / 45
GWT	149 / 47	122 / 69
Hibernate	389 / 133	378 / 199
JDK	14,252 / 632	7,483 / 1,981
Other	5,956 / 487	1,339 / 747
Total	**21,300 / 1,483**	**9,658 / 3,242**

Example Diversity : JavaScript

System	No. of Properties (Total/Unique)
JSCore/DOM	6,467/107
JQuery	1,793/96
Phonegap	126/27
Webworks	244/52
Other	1,297/300
Total	**9,927/582**

The Baker Framework



```
siddharth@siddharth-Inspiron-7520:~/BakerExamples$ cat JavaSnippet.txt
//*****

final Chronometer chrono = (Chronometer)findViewById(R.id.chronometer);
chrono.setBase(SystemClock.elapsedRealtime());
chrono.start();


siddharth@siddharth-Inspiron-7520:~/BakerExamples$ curl --data-urlencode 'pastedcode@JavaSnippet.txt" http:/
/gadget.cs.uwaterloo.ca:2145/snippet/getapijsonfromcode.php
{
   "api_elements": [
      {
         "precision": "1",
         "name": "Chronometer",
         "line_number": "3",
         "type": "api_type",
         "elements": [""android.widget.Chronometer""],
         "character": "56"  },
      {
         "precision": "1",
         "name": "Chronometer",
         "line_number": "3",
         "type": "api_type",
         "elements": [""android.widget.Chronometer""],
         "character": "78"  },
      {
         "precision": "1",
         "name": "Chronometer",
         "line_number": "4",
         "type": "api_type",
         "elements": [""android.widget.Chronometer""],
         "character": "129"  },
```

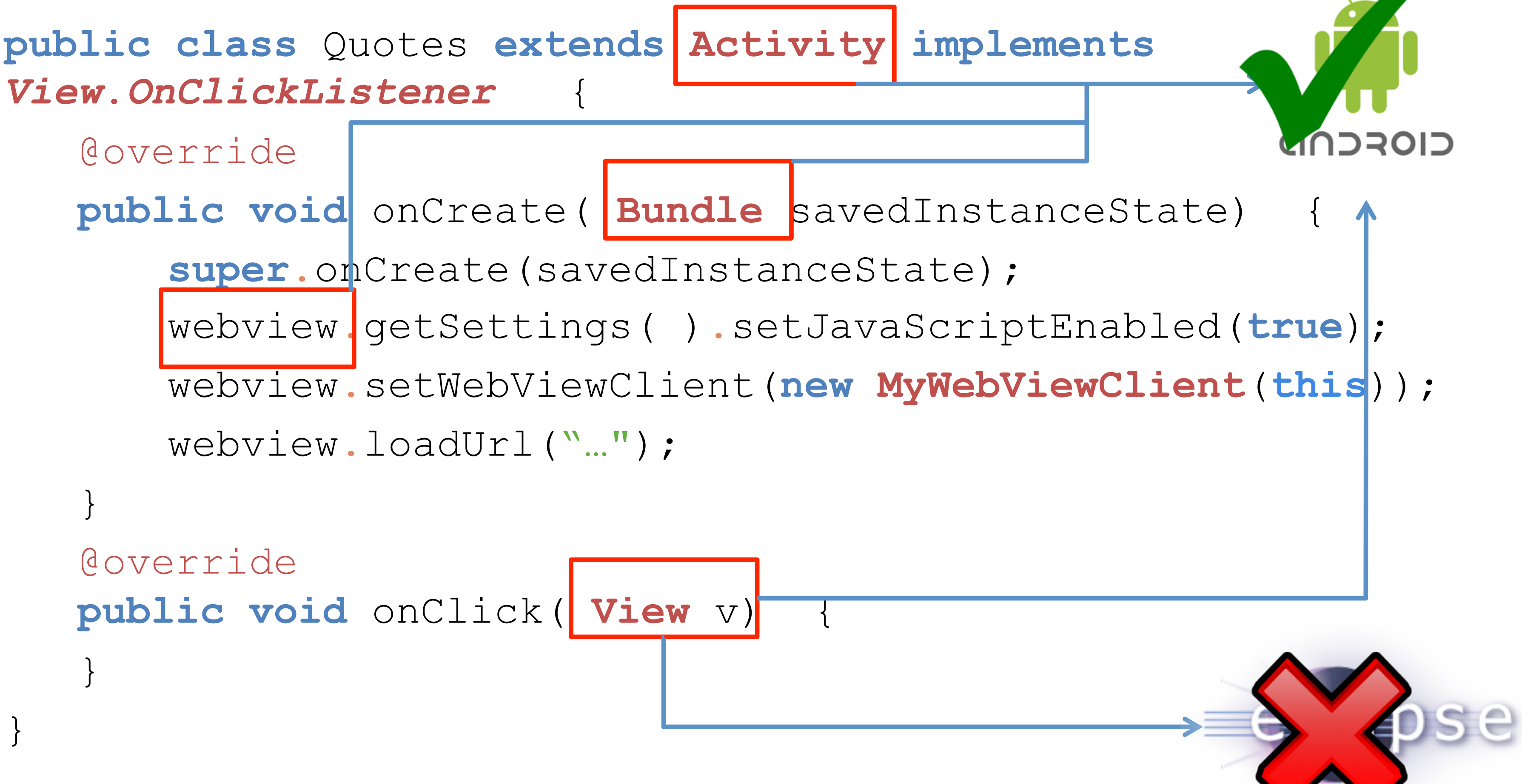
Quantifying Naming Collisions in the Java Oracle

	Types	Methods	Fields	Total	Average
FQN	1,646,650	14,206,944	3,149,206	19,002,800	-
PQN	-	9,455,644	2,571,384	12,027,028	-
UN	1,121,887	1,600,053	1,115,099	3,837,039	-
% ambiguous PQN	-	33%	37%	-	37%
% ambiguous UN	32%	89%	65%	-	80%

FQN : Fully Qualified Name (e.g., `android.widget.Chronometer.start()`)
PQN : Partially Qualified Name (e.g., `Chronometer.start()`)
UN : Unqualified Name (e.g., `start()`)

Extensions

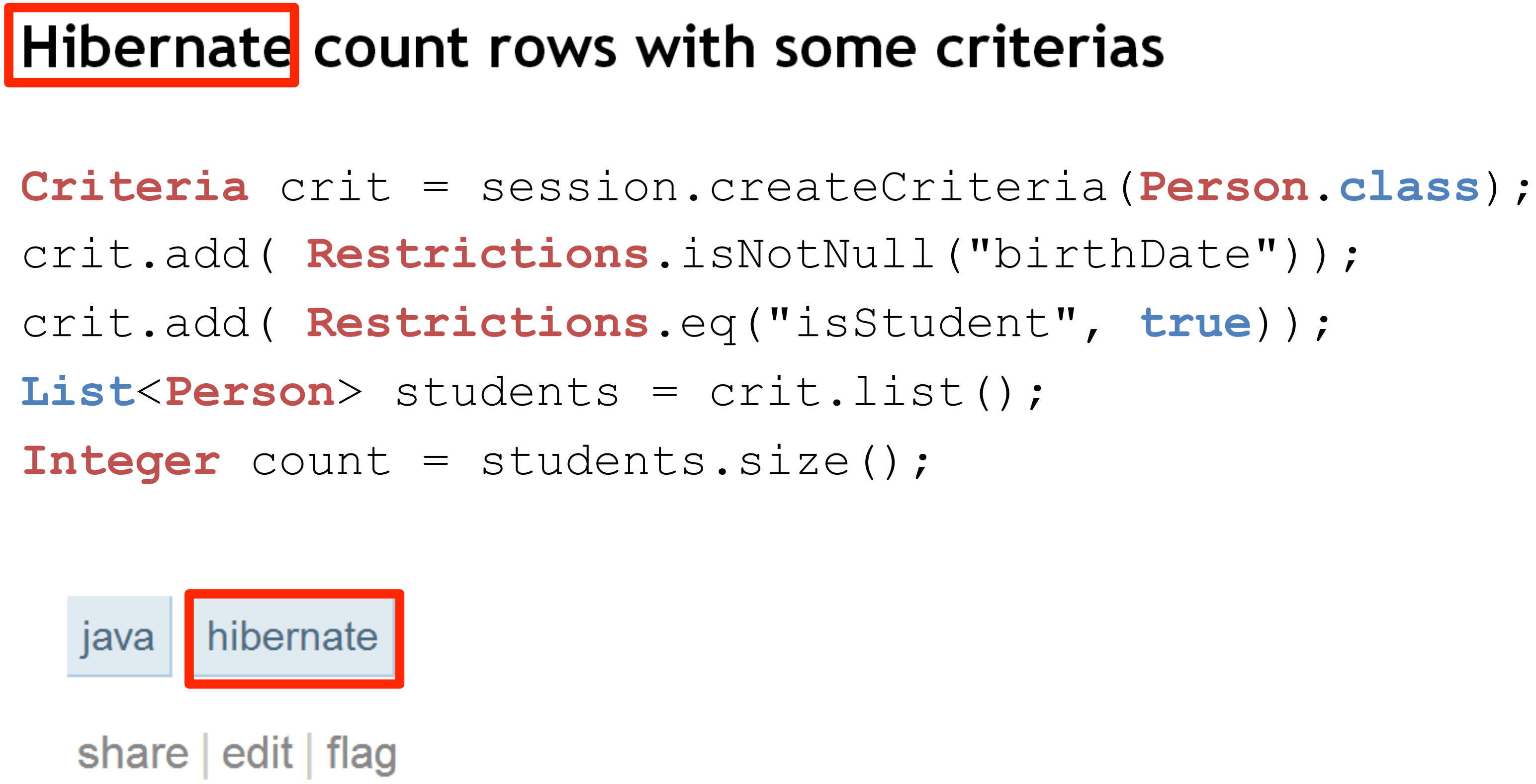
- Heuristics to predict libraries imported.



```
public class Quotes extends Activity implements
View.OnClickListener    {
    @override
    public void onCreate( Bundle savedInstanceState)   {
        super.onCreate(savedInstanceState);
        webview.getSettings( ).setJavaScriptEnabled(true);
        webview.setWebViewClient(new MyWebViewClient(this));
        webview.loadUrl("…");
    }
    @override
    public void onClick( View v)    {
    }
}
```

Extensions

- Utilizing metadata wherever available.

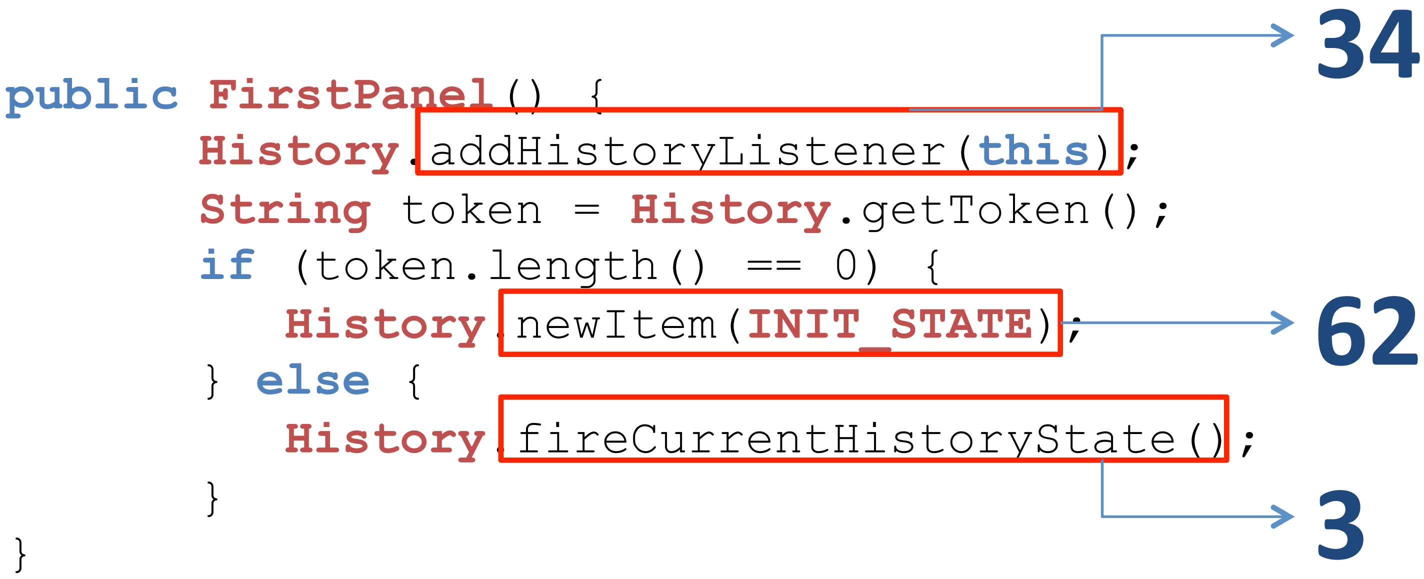


Hibernate count rows with some criterias

```
Criteria crit = session.createCriteria(Person.class);
crit.add( Restrictions.isNotNull("birthDate"));
crit.add( Restrictions.eq("isStudent", true));
List<Person> students = crit.list();
Integer count = students.size();
```

Extensions

- Pre-processing to re-order constraints and reduce running time.

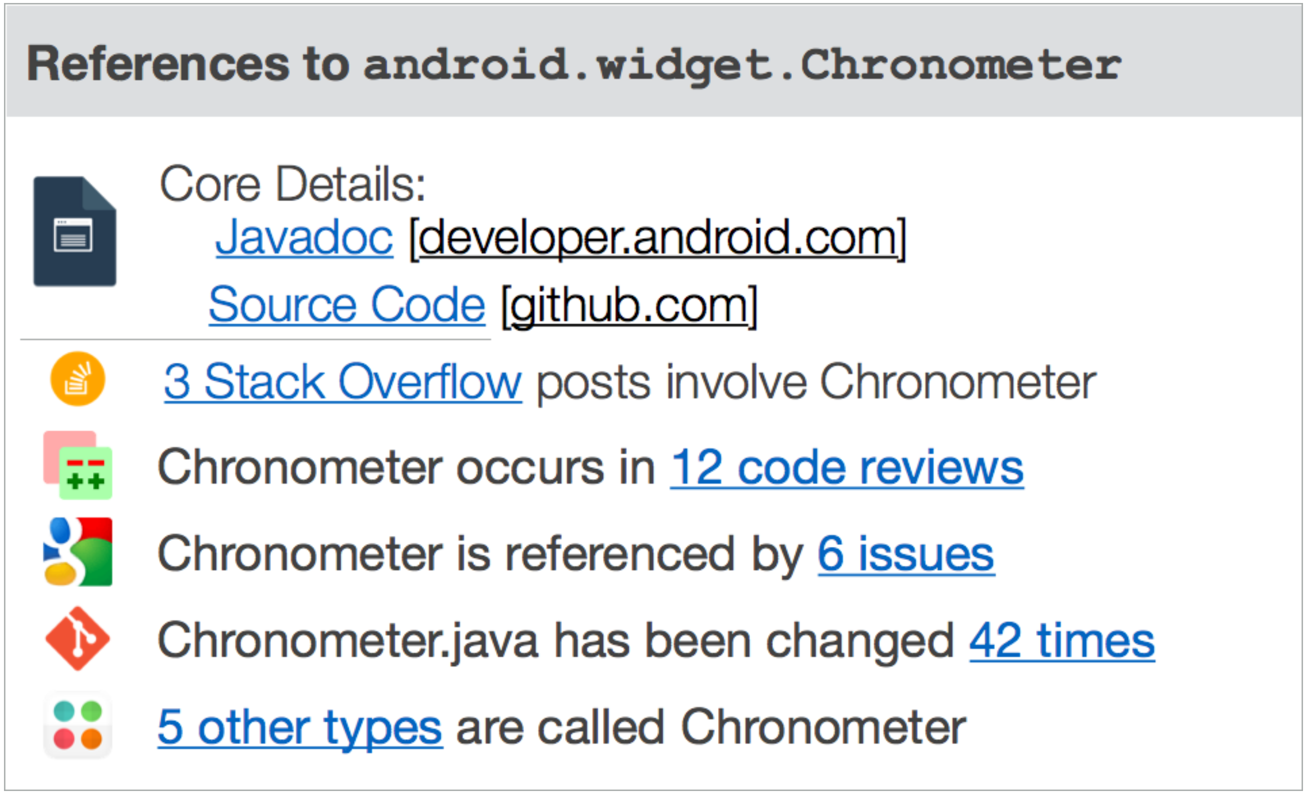


```
public FirstPanel() {
        History.addHistoryListener(this);
        String token = History.getToken();
        if (token.length() == 0) {
            History.newItem(INIT_STATE);
        } else {
            History.fireCurrentHistoryState();
        }
}
```

Related Work

- **ACE**
  - Does not need an oracle.
  - Cannot identify FQNs (only PQNs).
- **RecoDoc**
  - Specific target libraries.
  - Cannot resolve external references.

Traceability Link Recovery



References to `android.widget.Chronometer`

Core Details:
Javadoc [developer.android.com]
Source Code [github.com]

3 Stack Overflow posts involve Chronometer

Chronometer occurs in 12 code reviews

Chronometer is referenced by 6 issues

Chronometer.java has been changed 42 times

5 other types are called Chronometer

A mock-up of what could be achieved with an effective traceability linking strategy across a number of information sources.