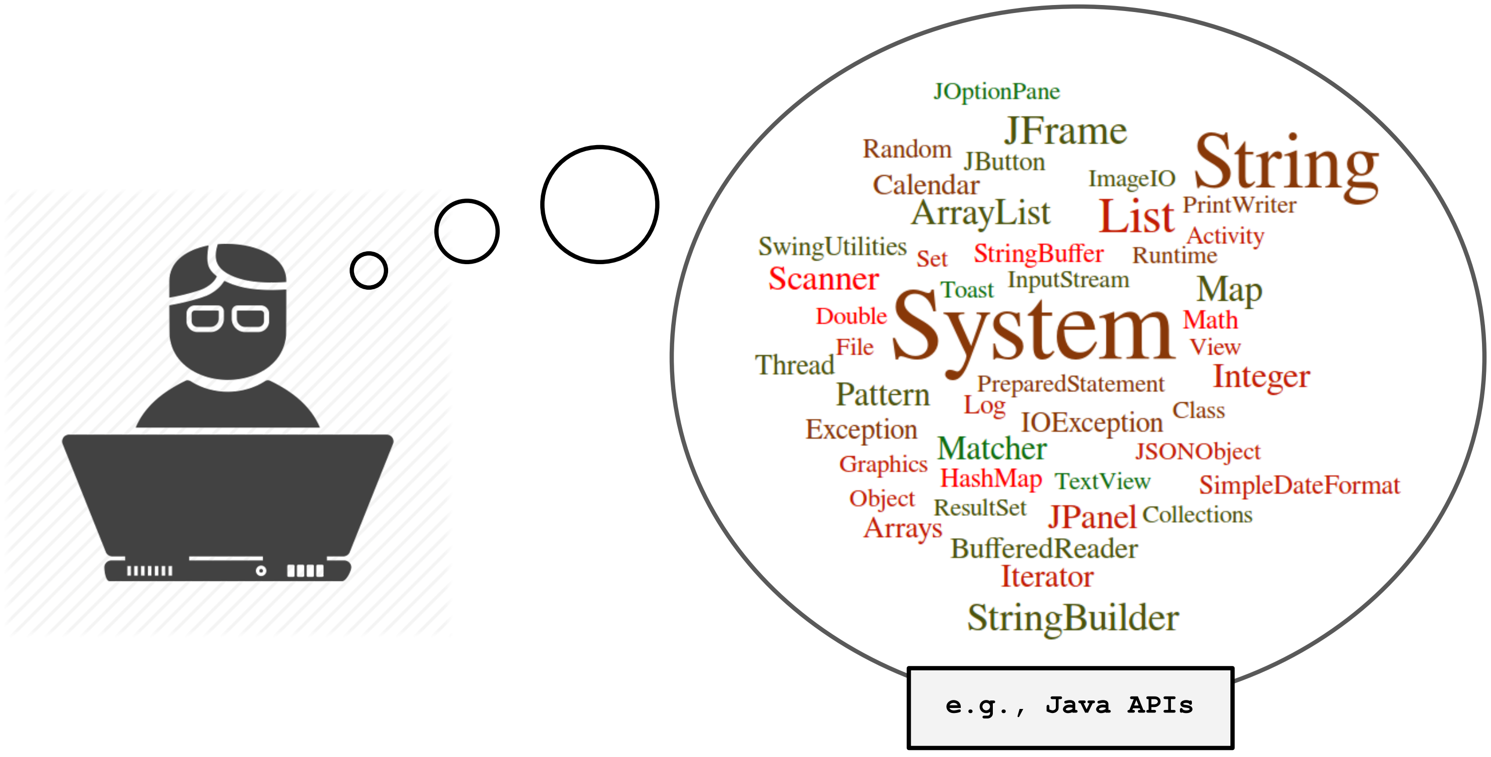
Visualizing API Usage Examples at Scale

Elena L. Glassman[1*], Tianyi Zhang[2*], Björn Hartmann[1], Miryung Kim[2]
[1]University of California, Berkeley
[2]University of California, Los Angeles

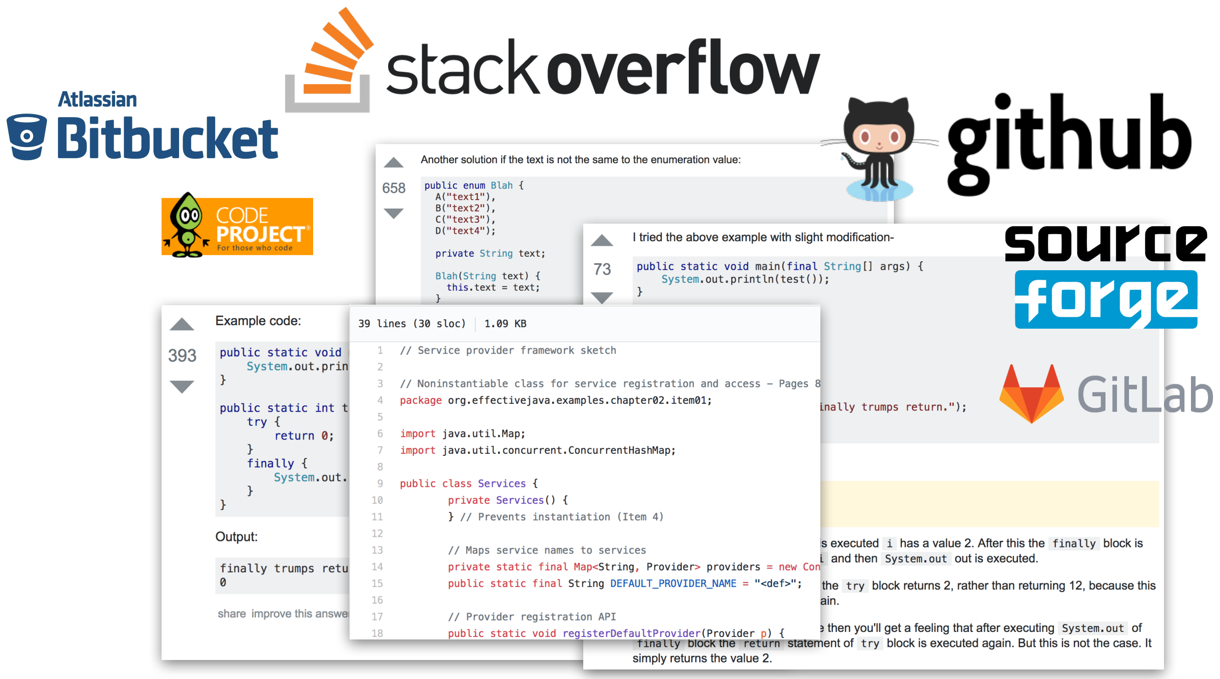
The two lead authors contributed equally to the work as part of an equal collaboration between both institutions.

Using APIs properly is a key challenge in programming



Status quo for answering “How have others used this API?”

Developers often search online for code examples to learn APIs [Sadowski et al., 2016]



Another solution if the text is not the same to the enumeration value:

658

```
public enum Blah {
  A("text1"),
  B("text2"),
  C("text3"),
  D("text4");

  private String text;

  Blah(String text) {
    this.text = text;
  }
```

I tried the above example with slight modification-

73

```
public static void main(final String[] args) {
    System.out.println(test());
}
```

```
inally trumps return.");
```

s executed `i` has a value 2. After this the `finally` block is
`i` and then `System.out` out is executed.

the `try` block returns 2, rather than returning 12, because this
ain.

e then you'll get a feeling that after executing `System.out` of
`finally` block the `return` statement of `try` block is executed again. But this is not the case. It
simply returns the value 2.

Example code:

393

```
public static void
    System.out.prin
}

public static int t
    try {
        return 0;
    }
    finally {
        System.out.
    }
}
```

Output:

```
finally trumps retu
0
```

share improve this answe

```
39 lines (30 sloc)  1.09 KB
```

```
// Service provider framework sketch

// Noninstantiable class for service registration and access - Pages 8
package org.effectivejava.examples.chapter02.item01;

import java.util.Map;
import java.util.concurrent.ConcurrentHashMap;

public class Services {
        private Services() {
        } // Prevents instantiation (Item 4)

        // Maps service names to services
        private static final Map<String, Provider> providers = new Con
        public static final String DEFAULT_PROVIDER_NAME = "<def>";

        // Provider registration API
        public static void registerDefaultProvider(Provider p) {
```

Status quo for answering “How have others used this API?”

- **Programmers only inspect a few of search results.**
  [Brandt 2009, Starke 2009, Duala-Ekoko & Robillard 2012]
- Individual code examples may suffer from
  - API usage violations [Zhang 2018]
  - insecure coding practices [Fischer 2017]
  - unchecked obsolete usage [Zhou & Walker 2016]
  - low readability [Treude & Robillard 2017]

How can we enable programmers to inspect more examples?

Code Demography:

How is **EVERYONE** using this API?

"How do other people create a `FileInputStream` object?"

API call of interest

```
new FileInputStream()
```

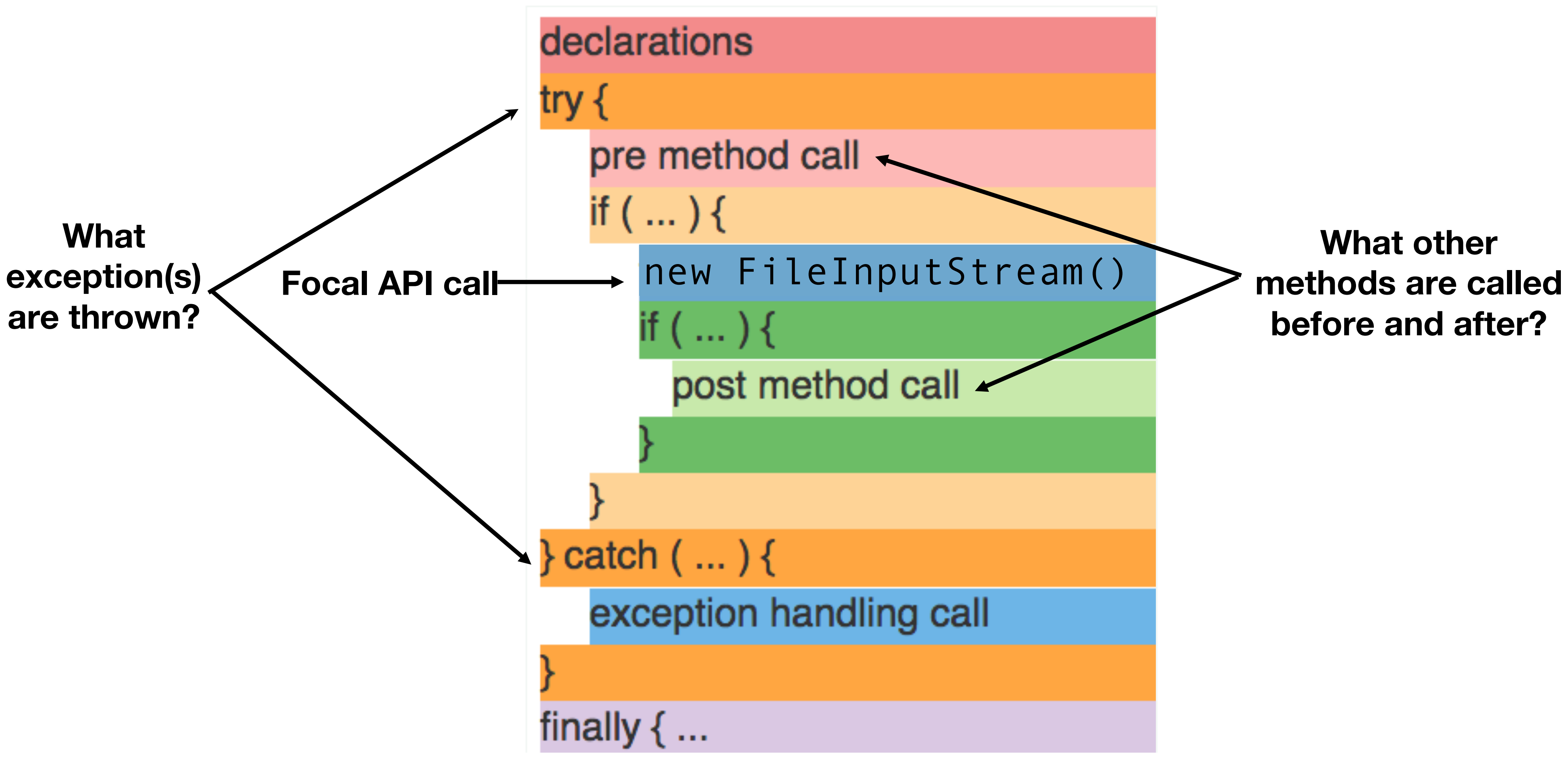
API Usage Questions:

- What arguments to pass into this call?
- How do I create these arguments?
- Do I need to check any pre-condition?
- What other methods to call together?
- What exception(s) does it throw?
- How do I handle the created object?

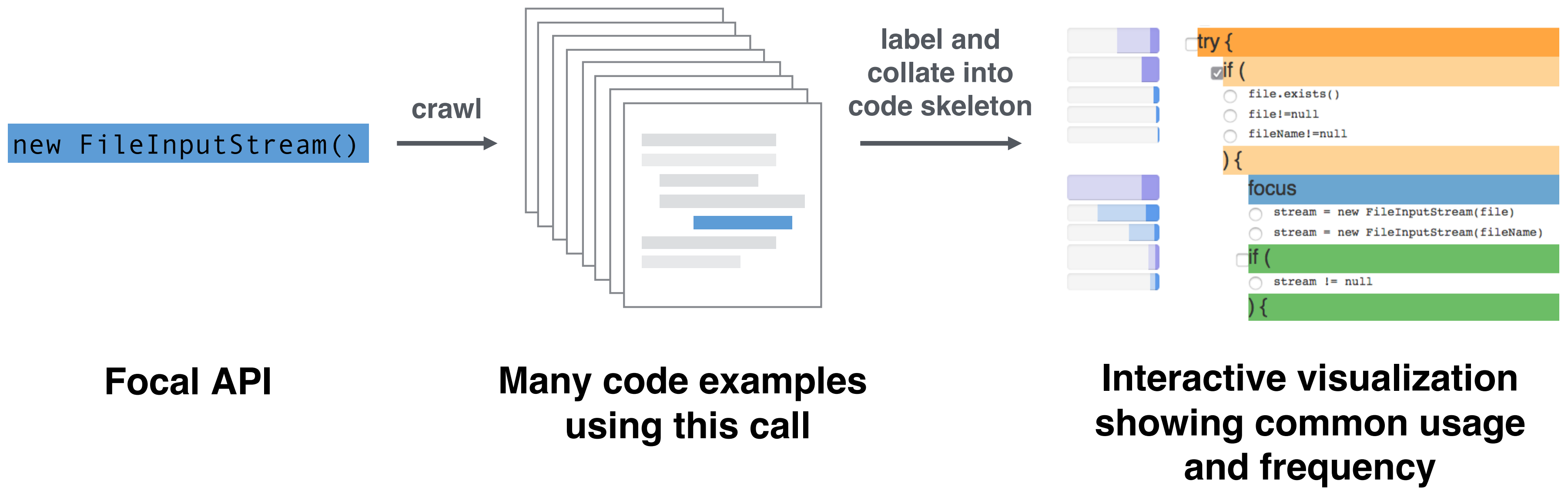
...

[Ko et al. 2004, Duala-Ekoko & Robillard 2012]

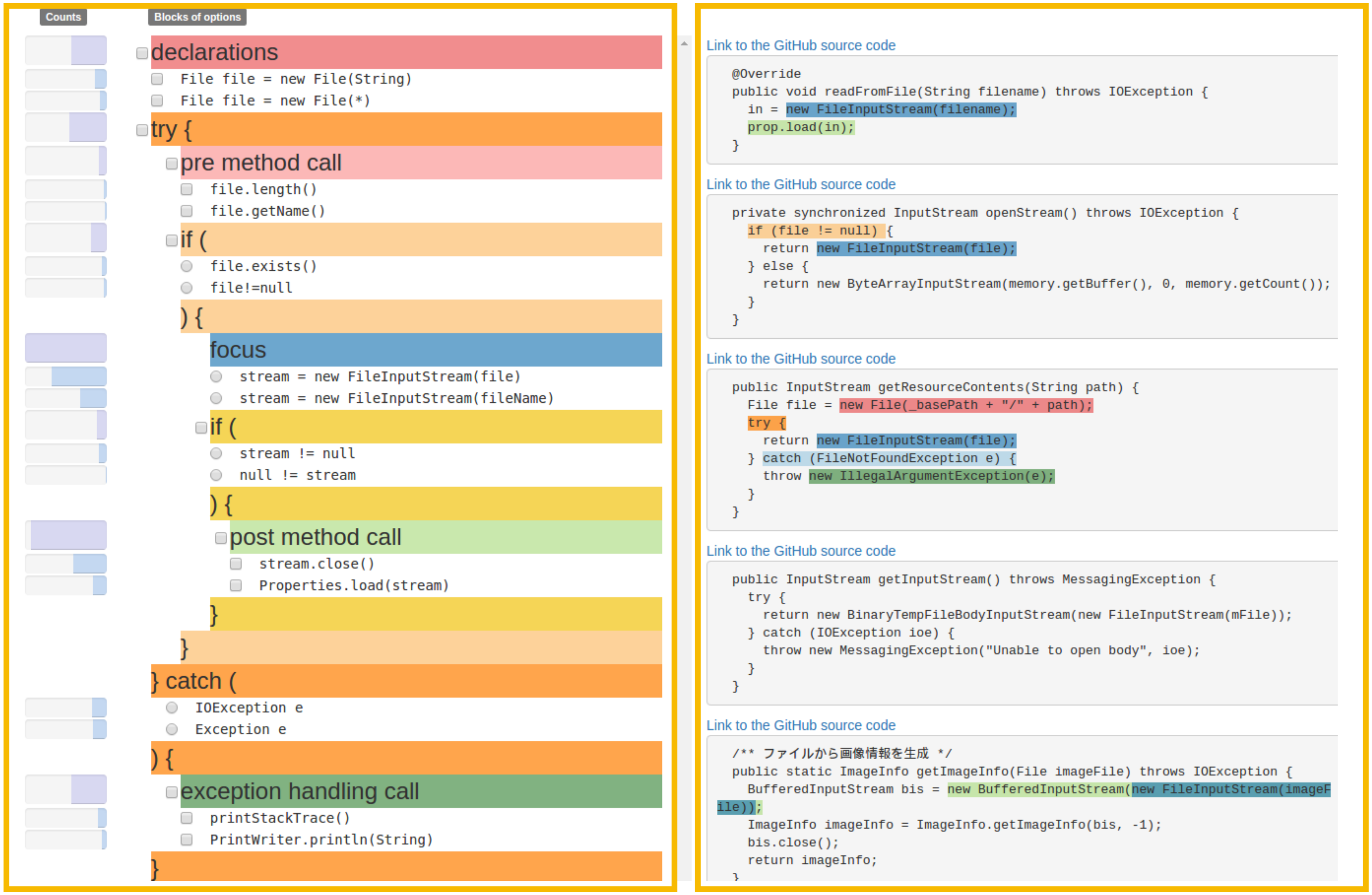
Designing an API Skeleton



Examplore: Visualizing Code Examples at Scale


new FileInputStream()
crawl
label and collate into code skeleton
Focal API
Many code examples using this call
Interactive visualization showing common usage and frequency

```
try {
  if (
    file.exists()
    file!=null
    fileName!=null
  ) {
    focus
      stream = new FileInputStream(file)
      stream = new FileInputStream(fileName)
    if (
      stream != null
    ) {
```



```
declarations
  File file = new File(String)
  File file = new File(*)
try {
  pre method call
    file.length()
    file.getName()
  if (
    file.exists()
    file!=null
  ) {
    focus
      stream = new FileInputStream(file)
      stream = new FileInputStream(fileName)
    if (
      stream != null
      null != stream
    ) {
      post method call
        stream.close()
        Properties.load(stream)
    }
  }
} catch (
  IOException e
  Exception e
) {
  exception handling call
    printStackTrace()
    PrintWriter.println(String)
}
```

Link to the GitHub source code

```
@Override
public void readFromFile(String filename) throws IOException {
  in = new FileInputStream(filename);
  prop.load(in);
}
```

Link to the GitHub source code

```
private synchronized InputStream openStream() throws IOException {
  if (file != null) {
    return new FileInputStream(file);
  } else {
    return new ByteArrayInputStream(memory.getBuffer(), 0, memory.getCount());
  }
}
```

Link to the GitHub source code

```
public InputStream getResourceContents(String path) {
  File file = new File(_basePath + "/" + path);
  try {
    return new FileInputStream(file);
  } catch (FileNotFoundException e) {
    throw new IllegalArgumentException(e);
  }
}
```

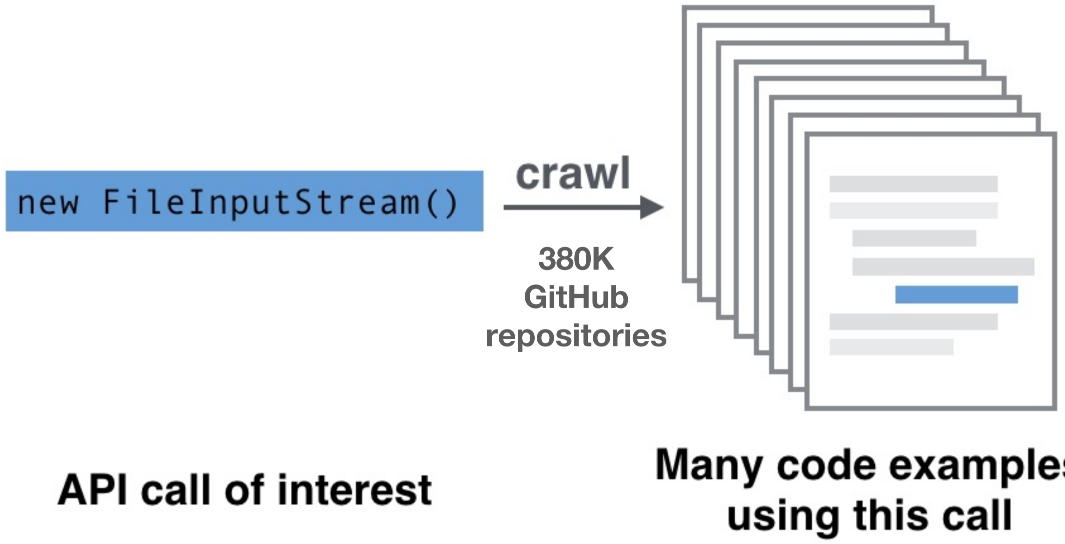
Link to the GitHub source code

```
public InputStream getInputStream() throws MessagingException {
  try {
    return new BinaryTempFileBodyInputStream(new FileInputStream(mFile));
  } catch (IOException ioe) {
    throw new MessagingException("Unable to open body", ioe);
  }
}
```

Link to the GitHub source code

```
/** ファイルから画像情報を生成 */
public static ImageInfo getImageInfo(File imageFile) throws IOException {
  BufferedInputStream bis = new BufferedInputStream(new FileInputStream(imageF
ile));
  ImageInfo imageInfo = ImageInfo.getImageInfo(bis, -1);
  bis.close();
  return imageInfo;
}
```

Mining API Usage from a Large Code Corpus

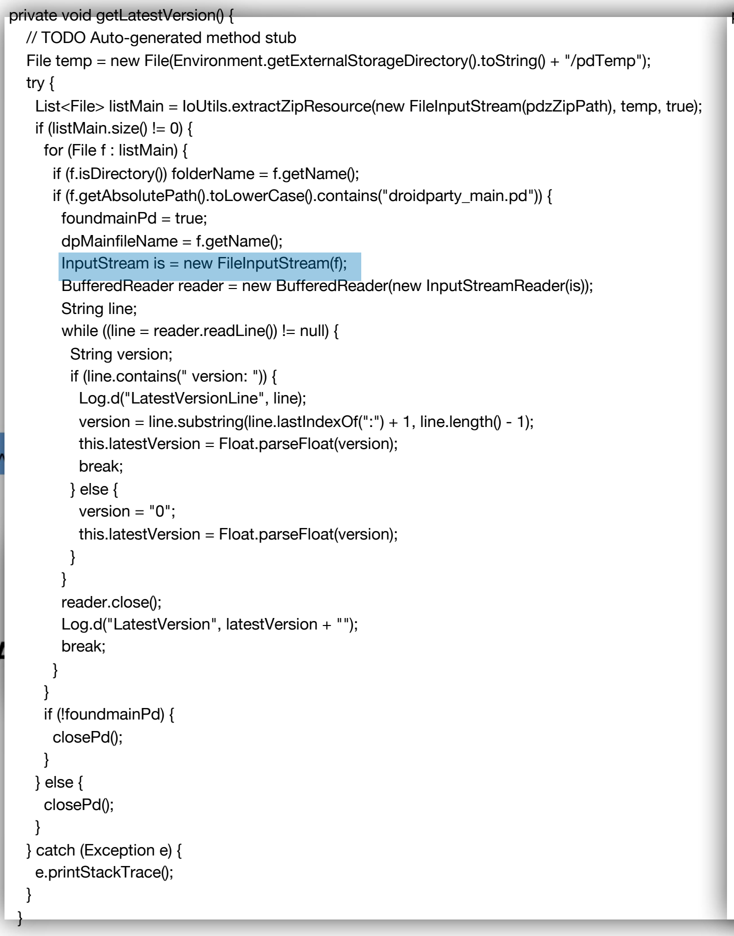


```
if (file != null) {
    return new FileInputStream(file);
} else {
    return new ByteArrayInputStream(…
}
```

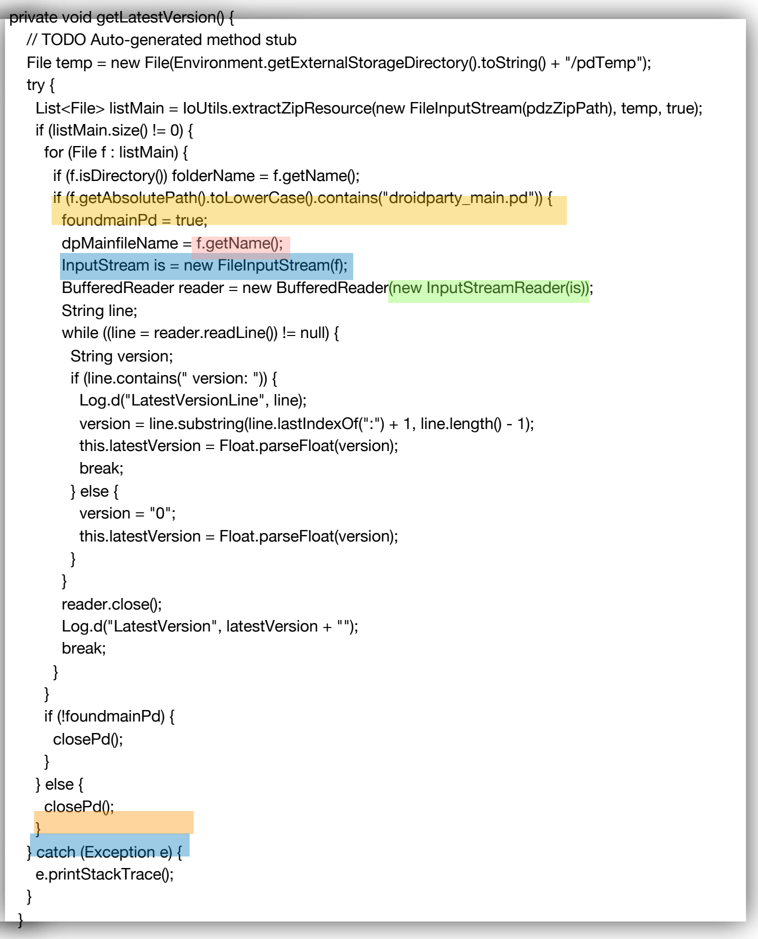
```
File file = new File(_basePath + "/" + path);
try {
    return new FileInputStream(file);
} catch (FileNotFoundException e) {
    throw new IllegalArgumentException(e);
}
```

```
File propertiesFile = getPropertiesFile();
try {
    InputStream in = new FileInputStream(propertiesFile);
    workspaceProperties.load(in);
} catch (IOException e) {
}
```

Program Slicing and Labeling



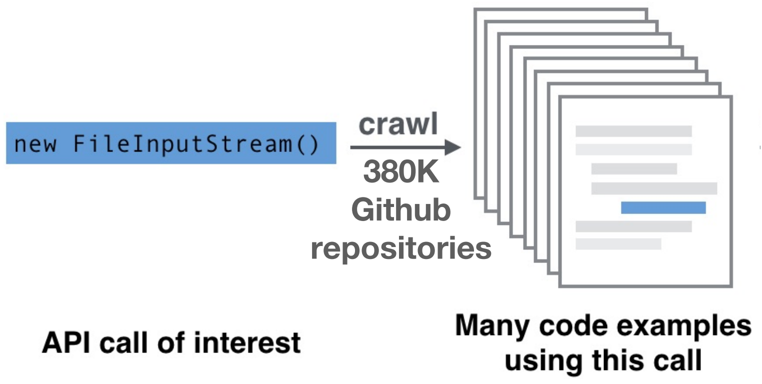
```
private void getLatestVersion() {
  // TODO Auto-generated method stub
  File temp = new File(Environment.getExternalStorageDirectory().toString() + "/pdTemp");
  try {
    List<File> listMain = IoUtils.extractZipResource(new FileInputStream(pdzZipPath), temp, true);
    if (listMain.size() != 0) {
      for (File f : listMain) {
        if (f.isDirectory()) folderName = f.getName();
        if (f.getAbsolutePath().toLowerCase().contains("droidparty_main.pd")) {
          foundmainPd = true;
          dpMainfileName = f.getName();
          InputStream is = new FileInputStream(f);
          BufferedReader reader = new BufferedReader(new InputStreamReader(is));
          String line;
          while ((line = reader.readLine()) != null) {
            String version;
            if (line.contains(" version: ")) {
              Log.d("LatestVersionLine", line);
              version = line.substring(line.lastIndexOf(":") + 1, line.length() - 1);
              this.latestVersion = Float.parseFloat(version);
              break;
            } else {
              version = "0";
              this.latestVersion = Float.parseFloat(version);
            }
          }
          reader.close();
          Log.d("LatestVersion", latestVersion + "");
          break;
        }
      }
      if (!foundmainPd) {
        closePd();
      }
    } else {
      closePd();
    }
  } catch (Exception e) {
    e.printStackTrace();
  }
}
```



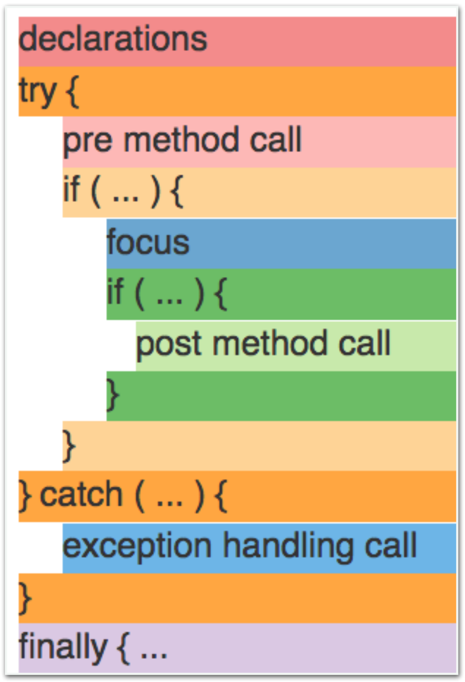
```
private void getLatestVersion() {
  // TODO Auto-generated method stub
  File temp = new File(Environment.getExternalStorageDirectory().toString() + "/pdTemp");
  try {
    List<File> listMain = IoUtils.extractZipResource(new FileInputStream(pdzZipPath), temp, true);
    if (listMain.size() != 0) {
      for (File f : listMain) {
        if (f.isDirectory()) folderName = f.getName();
        if (f.getAbsolutePath().toLowerCase().contains("droidparty_main.pd")) {
          foundmainPd = true;
          dpMainfileName = f.getName();
          InputStream is = new FileInputStream(f);
          BufferedReader reader = new BufferedReader(new InputStreamReader(is));
          String line;
          while ((line = reader.readLine()) != null) {
            String version;
            if (line.contains(" version: ")) {
              Log.d("LatestVersionLine", line);
              version = line.substring(line.lastIndexOf(":") + 1, line.length() - 1);
              this.latestVersion = Float.parseFloat(version);
              break;
            } else {
              version = "0";
              this.latestVersion = Float.parseFloat(version);
            }
          }
          reader.close();
          Log.d("LatestVersion", latestVersion + "");
          break;
        }
      }
      if (!foundmainPd) {
        closePd();
      }
    } else {
      closePd();
    }
  } catch (Exception e) {
    e.printStackTrace();
  }
}
```

Labeled Code Examples

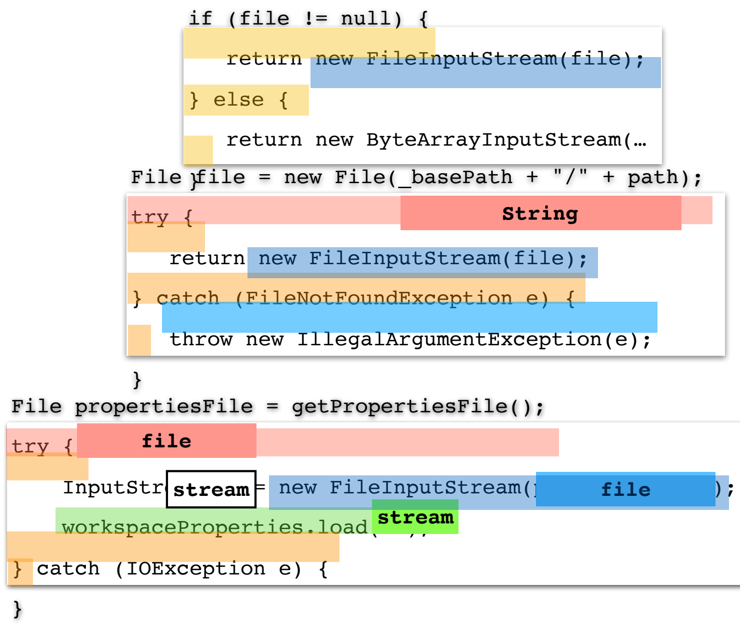
Code Canonicalization



Program Slicing and Labeling



Labeled Code Examples

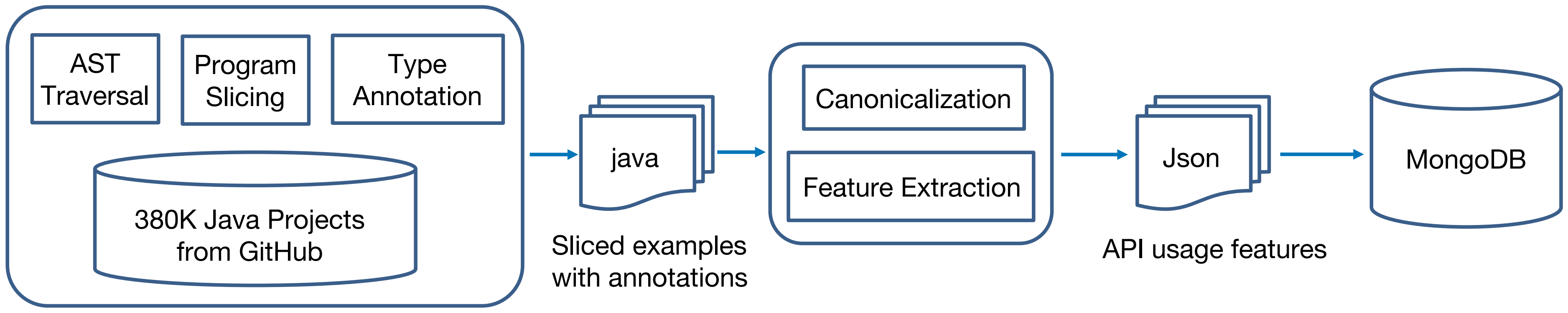


```
if (file != null) {
    return new FileInputStream(file);
} else {
    return new ByteArrayInputStream(…
```

```
File file = new File(_basePath + "/" + path);
try {                              String
    return new FileInputStream(file);
} catch (FileNotFoundException e) {
    throw new IllegalArgumentException(e);
}
```

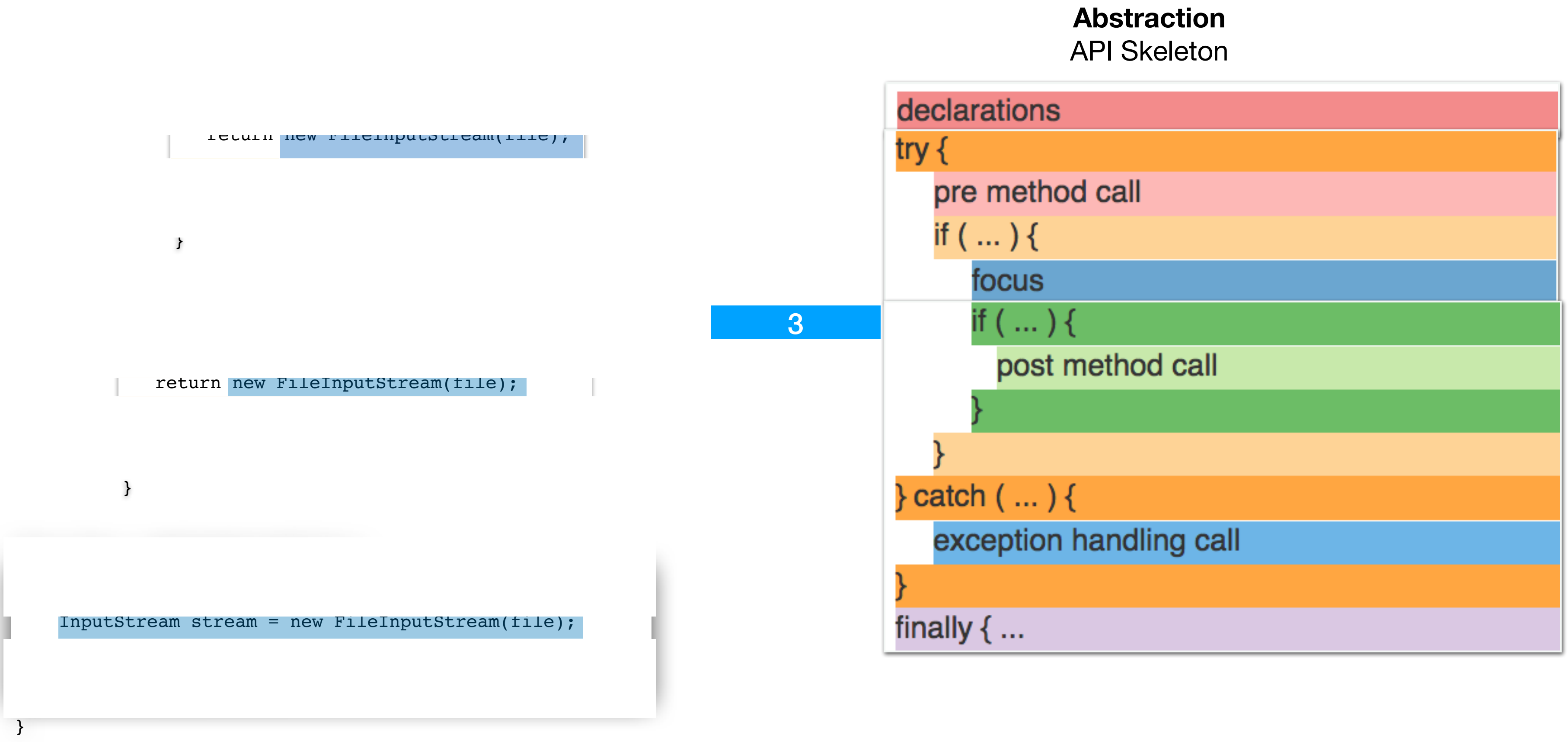
```
File propertiesFile = getPropertiesFile();
try {   file
    InputStr stream = new FileInputStream(   file   );
    workspaceProperties.load( stream );
} catch (IOException e) {
}
```

Back-end Architecture of Code Mining and Slicing



Examplore Interface

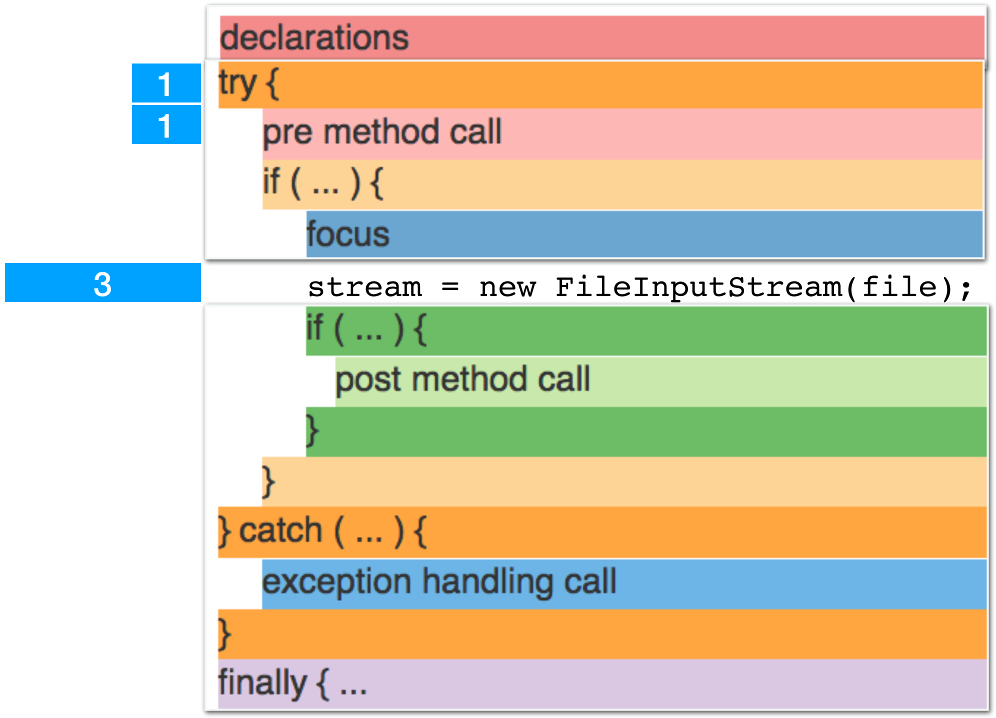
Mutual Alignment


```
            return new FileInputStream(file);

            }

        return new FileInputStream(file);

        }

    InputStream stream = new FileInputStream(file);

}
```

Examplore Interface

Mutual Alignment

Abstraction
API Skeleton



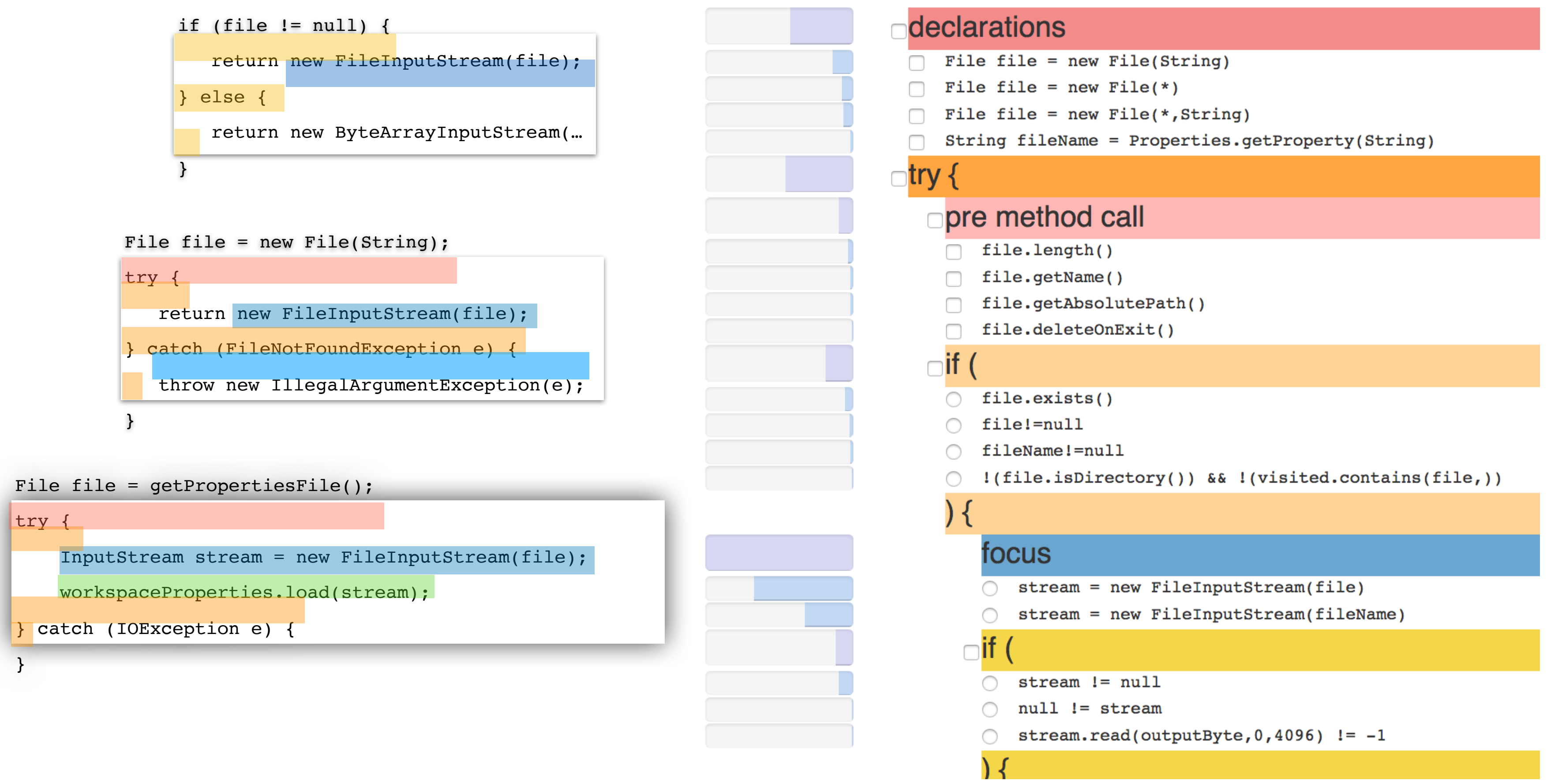
```
declarations
1  try {
1      pre method call
       if ( ... ) {
           focus
3          stream = new FileInputStream(file);
           if ( ... ) {
             post method call
           }
       }
   } catch ( ... ) {
       exception handling call
   }
   finally { ...
```

}

}

Examplore Interface

Mutual Alignment



```
if (file != null) {
    return new FileInputStream(file);
} else {
    return new ByteArrayInputStream(…
}
```

```
File file = new File(String);
try {
   return new FileInputStream(file);
} catch (FileNotFoundException e) {
   throw new IllegalArgumentException(e);
}
```

```
File file = getPropertiesFile();
try {
    InputStream stream = new FileInputStream(file);
    workspaceProperties.load(stream);
} catch (IOException e) {
}
```

Abstraction
API Skeleton

```
☐ declarations
  ☐ File file = new File(String)
  ☐ File file = new File(*)
  ☐ File file = new File(*,String)
  ☐ String fileName = Properties.getProperty(String)
☐ try {
  ☐ pre method call
    ☐ file.length()
    ☐ file.getName()
    ☐ file.getAbsolutePath()
    ☐ file.deleteOnExit()
  ☐ if (
    ○ file.exists()
    ○ file!=null
    ○ fileName!=null
    ○ !(file.isDirectory()) && !(visited.contains(file,))
  ) {
    focus
      ○ stream = new FileInputStream(file)
      ○ stream = new FileInputStream(fileName)
    ☐ if (
      ○ stream != null
      ○ null != stream
      ○ stream.read(outputByte,0,4096) != -1
    ) {
```

Live Demo

Theoretical Basis

Mutual alignment of contrasting examples

- Mutual alignment can **promote comprehension and abstraction**.
- Comparison brings **greater insight into the common structure**.
- Best results come from
  - **jointly interpreting examples**
  - **listing specific correspondences across examples**

[Kurtz et al. *Learning by Analogical Bootstrapping*, J. Learning Sciences, 2001]

Evaluation
Within-Subjects Lab Study on Answering API Usage Questions

- Recruited 16 CS students from UC Berkeley
- Picked out 3 APIs
  - 75% of participants had used `Map.get`
  - 38% had used `SQLiteDatabase.query`
  - 19% had used `Activity.findViewById`
- 50 min user study answering usage questions
  - 25 min block for API$_1$ [Baseline / Examplore]
  - 25 min block for API$_2$ [Examplore / Baseline]

Evaluation
Within-Subjects Lab Study on Answering API Usage Questions

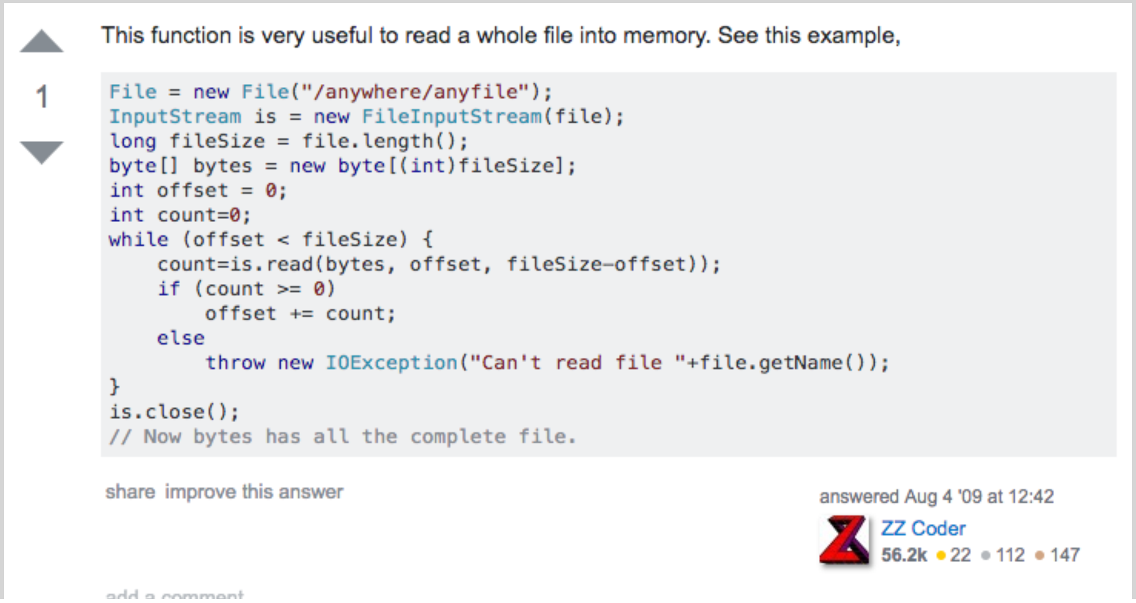
Sample of API Usage Questions

- Q2. How do I **create or initialize the arguments** so I can call this API method?
- Q6. How do programmers **handle the return value** of this API method?
- Q7. What are the **exceptions that programmers catch** and how do programmers **handle potential exceptions**?

Evaluation
Within-Subjects Lab Study on Answering API Usage Questions

Sample of API Usage Questions

- Q8. How might you **modify this code example** on Stack Overflow if you were going to copy and paste it into your own solution to the original prompt?



1

This function is very useful to read a whole file into memory. See this example,

```
File = new File("/anywhere/anyfile");
InputStream is = new FileInputStream(file);
long fileSize = file.length();
byte[] bytes = new byte[(int)fileSize];
int offset = 0;
int count=0;
while (offset < fileSize) {
    count=is.read(bytes, offset, fileSize-offset));
    if (count >= 0)
        offset += count;
    else
        throw new IOException("Can't read file "+file.getName());
}
is.close();
// Now bytes has all the complete file.
```

share improve this answer

answered Aug 4 '09 at 12:42

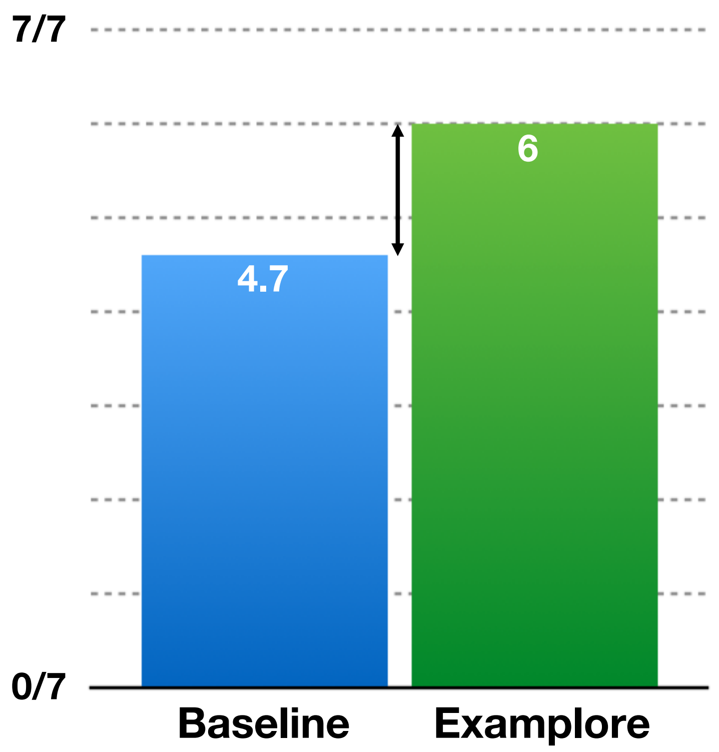
ZZ Coder

56.2k ● 22 ● 112 ● 147

Lab Study Results

- Examplore users investigated many relevant examples.
- Baseline users often answered based on one example or by guessing.

*Average # of correct answers on **Q1-7***



Mean difference is statistically significant

(paired t-test: t=3.02, df=15, p-value<0.01)

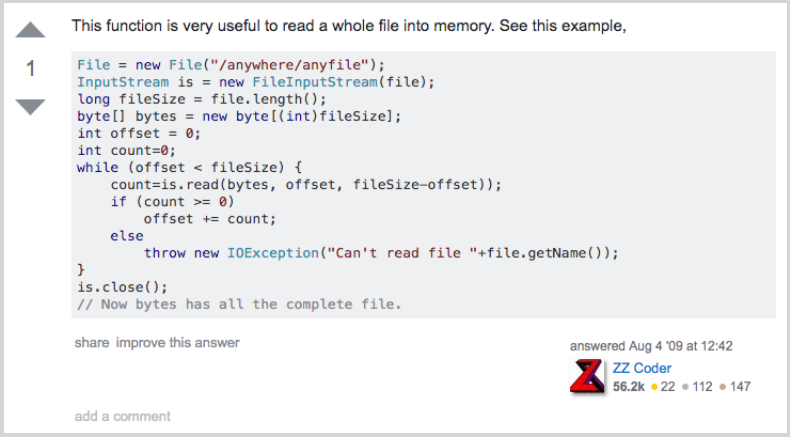
Lab Study Results

For **Q8**, 88% of participants gave valid comments about the StackOverflow answer.

The majority of participants' critiques…

- (Using the baseline) were about style and the mechanics of adaptation
- (Using Examplore) were about safety

- Q8. How might you **modify this code example on Stack Overflow** if you were going to copy and paste it into your own solution to the original prompt?



1

This function is very useful to read a whole file into memory. See this example,

```
File = new File("/anywhere/anyfile");
InputStream is = new FileInputStream(file);
long fileSize = file.length();
byte[] bytes = new byte[(int)fileSize];
int offset = 0;
int count=0;
while (offset < fileSize) {
    count=is.read(bytes, offset, fileSize-offset));
    if (count >= 0)
        offset += count;
    else
        throw new IOException("Can't read file "+file.getName());
}
is.close();
// Now bytes has all the complete file.
```

share improve this answer

answered Aug 4 '09 at 12:42

ZZ Coder

56.2k ● 22 ● 112 ● 147

add a comment

Lab Study Results

“[EXAMPLORE] provided structure to learning about the API.

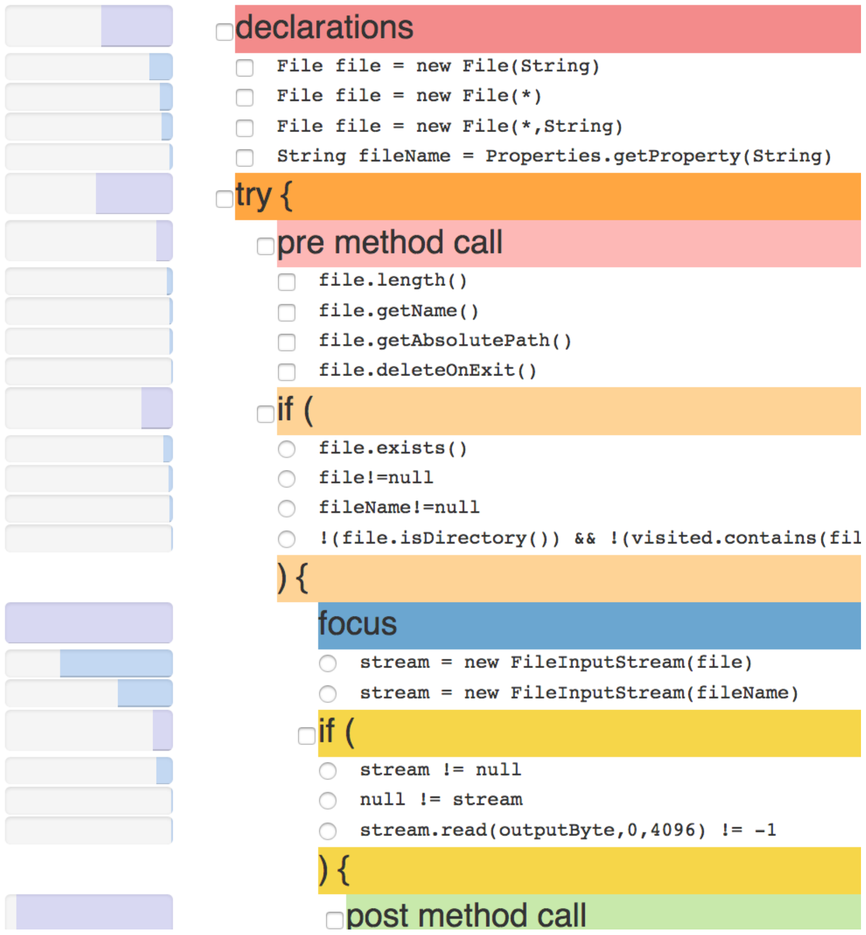
This structure guides functionality
while still showing variety of use.

The frequency of [each option] shows me if I am looking at a
random corner case or something commonly used.”

-P16

Future applications

- Release as a public resource based on Github
- For a specific codebase / organization
  - Code review
  - In-editor sidebar display
  - Corporate on-boarding
  - Data-driven library design and revision



declarations

```
File file = new File(String)
File file = new File(*)
File file = new File(*,String)
String fileName = Properties.getProperty(String)
```

try {

pre method call

```
file.length()
file.getName()
file.getAbsolutePath()
file.deleteOnExit()
```

if (

```
file.exists()
file!=null
fileName!=null
!(file.isDirectory()) && !(visited.contains(fil
```

) {

focus

```
stream = new FileInputStream(file)
stream = new FileInputStream(fileName)
```

if (

```
stream != null
null != stream
stream.read(outputByte,0,4096) != -1
```

) {

post method call

Summary

- The API skeleton is a key enabler for visualizing a large collection of API usage examples.
- The statistical distribution demonstrates the common and uncommon API usage in the community.
- By interacting with the skeleton, users can easily filter code examples and quickly drill down to those of interest.

Demo is available at https://eglassman.github.io/examplore/