



## Compiler project document

### **Group 10**

#### **Group members:**

**Ali Sedaghi**

**Mohammad Yarmoghadam**

**Amir Masoud Golestane**

**Hasti Karamdel**

|  |    |
|--|----|
| Introduction .....                                       | 3  |
| What is Understand? .....                                | 3  |
| This project.....  | 3  |
| Our sections .....                                       | 4  |
| Project division .....                                   | 5  |
| What is expected? .....                                  | 6  |
| Implementation .....                                     | 9  |
| Initializing .....                                       | 9  |
| Import/Importby .....                                    | 13 |
| Modify (Deref) Partial and Modifyby (Deref) Partial..... | 19 |
| Counting the number of statements.....                   | 25 |
| Cyclomatic.....  | 31 |
| Tests and results.....                                   | 47 |
| Procedure and Challenges.....                            | 62 |
| Conclusions and Recommendations .....                    | 66 |
| Additional Sources .....                                 | 67 |

# Introduction

## What is Understand?

Understand is a static analysis tool focused on source code comprehension, metrics, and standards testing. It is designed to help maintain and understand large amounts of legacy or newly created source code. It provides a cross-platform, multi-language, maintenance-oriented IDE (interactive development environment).

Understand uses more than 50 different graphs to help you visualize exactly what your code is doing and how it is built. Browse call trees, explore dependencies, verify UML structures or design your own graphs with the API.

Understand has architecture features that help you create hierarchical aggregations of source code units. You can name these units and manipulate them in various ways to create interesting hierarchies for analysis.

## This project

Unfortunately, the Understand API source code is not publicly available, making it difficult to change, customize, and reuse in new activities and environments which appears in academic researches.

This project aims to provide an open-source implementation of the Understand Python API to analyze the source codes. We primarily focus on implementing the API for Java programs using Python programming languages and compiler tools such as ANTLR. To develop an open-source implementation of Understand Python API, we look at the structures used by Understand for analyzing source codes.

At phase two of this project the goal is to implement an extended version of the Sci-tools Understand APIs for computing source metrics. The following

APIs are used to commute source code metrics at different abstraction levels:

- [understand.Db.html](#)
- [understand.Ent.html](#)

The computation of source code metrics consists of two steps. First, developing the required APIs for querying the database created in Phase one of the project. Second, querying the database using the developed API to find the appropriate entities and reference kinds involved in computing metrics according to the metric definitions. You can find more references and details in this link:

[Core source code metrics development](#)

## Our sections

Most of the data captured by Understand involves Entities and References.

Entity: An Entity is anything in the code that Understand captures information on: i.e., A file, a class, a variable, a function, etc. In the Perl API, Entities are represented with the `Understand::Ent` class. In Python, it is the `Understand.Ent` class.

Reference: A specific place where an entity appears in the code. A reference is always defined as a relationship between two entities. e.g., function Bar is called on line 14 of function Foo. In the Perl API, References are represented with the `Understand::Ref` class. In Python, it is the `Understand.Ref` class.

Every entity and reference have a unique set of attributes that can be queried by the API. A few of the attributes you can view for an entity would be its name, its type, any associated comments, what kind of entity it is, and if it has them: its parent entity and its parameters. On the other hand, a reference would have both of the entities associated with it as well as the file, line, and column where the reference occurs and what kind of reference it is.

This project should support different reference kinds that we can have in our java code, each of these references can also use different entities. This is the table of the reference kinds which are each implemented by a different group.

You can see the reference kinds in this table:

[https://m-zakeri.github.io/OpenUnderstand/reference\\_kinds/](https://m-zakeri.github.io/OpenUnderstand/reference_kinds/)

You can see the entity kinds in this table:

[https://m-zakeri.github.io/OpenUnderstand/entity\\_kinds/](https://m-zakeri.github.io/OpenUnderstand/entity_kinds/)

Our group has tried to implement a code for the entities of Import/Import by & Modify (Deref) Partial and Modifyby (Deref) Partial. To recognize these entities, we have to use different entity kinds to Analyze our java code and find the references of each time they are used.

In the second phase we implemented CountStmt, CountStmtDecl, CountStmtExe and Cyclomatic. Source code metrics are essential components in the software measurement process. They are extracted from the source code of the software, and their values allow us to reach conclusions about the quality attributes measured by the metrics. OpenUnderstand supports the following source code metrics for the Java programming language:

[Source code metrics](#)

## Project division

We decided it is best to divide the project into different parts we could each do separately and also a part we could all participate together. Therefore two of us worked on import/import by and the two others worked on modify/modify by. In the end we all gathered to study and test our codes and write this documentation.

Also in the second phase we took the same approach. Therefore, we were divided into two groups of two. Although in this phase a lot more discussing and sharing was needed and the unity was much more tight than the first phase.

## What is expected?

We need to implement a standalone python code which scans through a java project. Then it must be able to detect the reference kind we want (import or modify) and fill our database with the attributes we need for each kind.

Import/ImportBy:

Java Import Demand indicates a file has an on-demand import statement for a package or class. For example, if a file named file.java has imported a class, we would store the following data in our table:

- File id
- Imported class
- Entity kind id
- Line and Col in which import is used
- And also the responding entity which is imported by the importing entity

Reference:

[https://m-zakeri.github.io/OpenUnderstand/reference\\_kinds/#java-import-and-importby](https://m-zakeri.github.io/OpenUnderstand/reference_kinds/#java-import-and-importby)

## Modify (Deref) Partial and Modifyby (Deref) Partial:

Java Modify and Modifyby indicates that a variable's value is modified or both read and set, as with the increment (++), decrement (--), and assignment/operator combinations (\*=, /=, ...). This reference is like Modify and Modifyby. But, it is used when an entity modifies some elements of a variable of collection type. Same as import we also need to detect in which scope a variable is being modified, for example if it has been modified in a function, we would add the following data to our table:

- File id
- Scope id
- Entity kind id
- Line and Col in which import is used

- And also the responding entity which is modified by the modifying entity

Reference:

[https://m-zakeri.github.io/OpenUnderstand/reference\\_kinds/#java-modify-and-modifyby](https://m-zakeri.github.io/OpenUnderstand/reference_kinds/#java-modify-and-modifyby)

### CountStmt, CountStmtDecl & CountStmtExe:

Number of declarative plus executable statements, number of declarative statements and number of executable statements.

For Java we count:

Project, File, Class, Interface, Method

|                             | Executable   | Declarative (Function) | Empty          |           |
|-----------------------------|--------------|------------------------|----------------|-----------|
| int main(){                 | 0            | 0                      | 0              |           |
| for(int i=0;                | 0            | 1                      | 0              |           |
| i<10;                       | 1            | 0                      | 0              |           |
| i++)                        | 1            | 0                      | 0              |           |
| ;                           | 0            | 0                      | 1              |           |
|                             | 0            | 0                      | 0              |           |
| int j = func();             | 0            | 1                      | 0              |           |
| int k = 0;                  | 0            | 1                      | 0              |           |
| int l = 1;                  | 0            | 1                      | 0              |           |
|                             | 0            | 0                      | 0              |           |
| int m                       | 0            | 1                      | 0              |           |
| = func();                   | 0            | 0                      | 0              |           |
| int n;                      | 0            | 1                      | 0              |           |
| n = 1;                      | 1            | 0                      | 0              |           |
|                             | 0            | 0                      | 0              |           |
| #ifdef A_VERY_NICE_VARIABLE | 0            | 0                      | 0              |           |
|                             | 0            | 0                      | 0              |           |
| int j=0;                    | 0            | 0                      | 0              |           |
| cout<<"coendi";             | 0            | 0                      | 0              |           |
| #endif                      | 0            | 0                      | 0              |           |
|                             | 0            | 0                      | 0              |           |
| return 0;                   | 1            | 0                      | 0              |           |
| }                           | 0            | 0                      | 0              |           |
|                             | 4            | 6                      | 1              | = 11      |
|                             | CountStmtExe | CountStmtDecl          | CountStmtEmpty | CountStmt |

## Cyclomatic:

The cyclomatic complexity of any structured program with only one entrance point and one exit point is equal to the number of decision points contained in that program plus one. Understand counts the keywords for decision points (FOR, WHILE, etc) and then adds 1. For a switch statement, each 'case' is counted as one.

For Java we count:

Project, Method

| Cyclomatic                                     |   |   |   |   |   |   |   |   |         |    |
|--|---|---|---|---|---|---|---|---|---------|----|
| Formula: case, catch, do, for, if, ?, while +1 |   |   |   |   |   |   |   |   |         |    |
| Result (for function cyclomaticDemo()): 10     |   |   |   |   |   |   |   |   |         |    |
| void cyclomaticDemo(){                         | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0       |    |
| bool a=true,b=true,c=true;                     | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0       |    |
| if(a    (b && c)){                             | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0       |    |
| while(a? b : c){                               | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0       |    |
| for(int i=0; i < 10; i++){                     | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0       |    |
| switch(i){                                     | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0       |    |
| case 1:  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1       |    |
| case 2:  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1       |    |
| cout<<endl;                                    | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0       |    |
| break;   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0       |    |
| case 5:  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1       |    |
| break;   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0       |    |
| default:                                       | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0       |    |
| cout<<i<<endl;                                 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0       |    |
| }  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0       |    |
| }  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0       |    |
| }  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0       |    |
| } else{  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0       |    |
| try{   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0       |    |
| do{  | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0       |    |
| cout<<a<<b<<c<<endl;                           | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0       |    |
| }while(a);                                     | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0       |    |
| }  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0       |    |
| catch(...){                                    | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0       |    |
| }  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0       |    |
| }  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0       |    |
| }  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0       |    |
|  | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 3 + 1 = | 10 |



# Implementation

## Initializing

This section of the code is repeated in both of our files, because we implemented a standalone python code for each of the commands.

### Imports:

First, we need to import antlr, Java Lexer, Parser and Listener. Then to connect and use the database we need to import the API and functions needed to fill the tables, the models for entity kinds and etc.

```
import os
from antlr4 import *
from pathlib import Path
from gen.javaLabeled.JavaLexer import JavaLexer
from gen.javaLabeled.JavaParserLabeled import JavaParserLabeled
from gen.javaLabeled.JavaParserLabeledListener import JavaParserLabeledListener
from oudb.fill import main as db_fill
from oudb.api import create_db, open as db_open
from oudb.models import KindModel, EntityModel, ReferenceModel
```

### Benchmark settings:

To run and test each of the benchmark projects we need to set a project index and add the names of the projects that we need to be tested. We also need to add a path so that the project files can be read and tested from there, to handle this matter we have made a similar path for each of the projects and the only thing that differs each path is the project name itself; resulting an array for the paths of each project. We also need to set the path of our database file which includes the final results and tables.

```
PRJ_INDEX = 3
REF_NAME = "import"

def get_project_info(index, ref_name):
    project_names = [
        'calculator_app',
        'JSON',
```

```

        'testing_legacy_code',
        'jhotdraw-develop',
        'xerces2j',
        'jvlt-1.3.2',
        'jfreechart',
        'ganttproject',
        '105_freemind',
    ]
    project_name = project_names[index]
    db_path = f"../../databases/{ref_name}/{project_name}"
    if ref_name == "origin":
        db_path = db_path + ".udb"
    else:
        db_path = db_path + ".oudb"
    project_path = f"../../benchmarks/{project_name}"

    db_path = os.path.abspath(db_path)
    project_path = os.path.abspath(project_path)

    return {
        'PROJECT_NAME': project_name,
        'DB_PATH': db_path,
        'PROJECT_PATH': project_path,
    }

```

### Init:

In the beginning of the class, we set the database name, project name and directory and also the two arrays with the names and paths of the files.

```

def __init__(self, db_name, project_dir, project_name=None):
    self.db_name = db_name
    self.project_dir = project_dir
    self.project_name = project_name
    self.files = []

```

### Initdb:

We used createdb to create our database then used the fill function to add

the models, the database is added to the path we gave it in the beginning and is ready to use.

```
def init_db(self):
    create_db(self.db_name, self.project_dir, self.project_name)
    db_fill()
    db_open(self.db_name)
```

### Get Java Files:

The project directories that we have include many files, that is why we have to find the .java files in order to continue. For each java file we add the name and path to the arrays we have.

```
def get_java_files(self):
    for dir_path, _, file_names in os.walk(self.project_dir):
        for file in file_names:
            if '.java' in str(file):
                path = os.path.join(dir_path, file)
                path = path.replace("/", "\\")
                path = os.path.abspath(path)
                self.files.append((file, path))
                add_java_file_entity(path, file)
```

### Get Parent:

This function checks our database for the parent entity if there is a parent entity with the id of the “java file” entity kind and also the name and the path given, it will return its object.

```
def get_parent(parent_file_name, files):
    file_names, file_paths = zip(*files)
    parent_file_index = file_names.index(parent_file_name)
    parent_file_path = file_paths[parent_file_index]
    parent_entity = EntityModel.get_or_none(
        _kind=KindModel.get_or_none(_name="Java File").get_id(),
        _name=parent_file_name,
        _longname=parent_file_path,
    )
```

```
return parent_entity, parent_file_path
```

### Add Java file entity:

This function gets each java file name and path from the arrays and sets its entity in the entity kinds table, or if there already is an id for that entity, it will return the object.

```
def add_java_file_entity(file_path, file_name):
    kind_id = KindModel.get_or_none(_name="Java File").get_id()
    obj, _ = EntityModel.get_or_create(
        _kind=kind_id,
        _name=file_name,
        _longname=file_path,
        _contents=FileStream(file_path, encoding="utf-8"),
    )
    return obj
```

### Get parse tree:

This function gets the file from the file path given(using FileStream) then creates a java lexer for that file. From that lexer we can tokenize it and make our parser.

```
def get_parse_tree(file_path):
    file = FileStream(file_path, encoding="utf-8")
    lexer = JavaLexer(file)
    tokens = CommonTokenStream(lexer)
    parser = JavaParserLabeled(tokens)
    return parser.compilationUnit()
```

## Import/Importby

After getting all of the java files and adding the entity kinds to our database table, we need to focus on finding the import references in our files.

### Import Listener:

The goal of this class is to create a list of objects for each “import” in our file. The object should have the attributes needed to reference an import. At first we get the Imported classes name and check if it already is in the array of names we created before, because if it isn’t that means our imported class is a java built-in class type. Then with the help of our listener we get the line and column where the import is used in our file. After setting all the attributes we shape our object for that import and add it to the list.

```
class ImportListener(JavaParserLabeledListener):
    def __init__(self, files):
        self.repository = []
        self.files = files

    def enterImportDeclaration(self, ctx: JavaParserLabeled.ImportDeclaration):
        imported_class_longname = ctx.qualifiedName().getText()
        imported_class_name = imported_class_longname.split('.')[-1]

        is_built_in = False
        imported_class_file_name = imported_class_name + ".java"
        if imported_class_file_name not in [file[0] for file in self.files]:
            is_built_in = True
            imported_class_file_name = None

        line = ctx.children[0].symbol.line
        col = ctx.children[0].symbol.column

        self.repository.append({
            'imported_class_name': imported_class_name,
            'imported_class_longname': imported_class_longname,
            'is_built_in': is_built_in,
```

```

        'imported_class_file_name': imported_class_file_name,
        'line': line,
        'column': col,
    })

```

### Imported Entity Listener:

Same as the previous class this listener indicates the kind of our imported entity.

```

class ImportedEntityListener(JavaParserLabeledListener):
    def __init__(self, name):
        self.body = None
        self.branches = None
        self.type = None
        self.name = name

    def enterClassDeclaration(self, ctx: JavaParserLabeled.ClassDeclarationContext):
        if self.name == ctx.IDENTIFIER().getText():
            self.body = ctx.getText()
            self.branches = ctx.parentCtx.children

    def enterInterfaceDeclaration(self, ctx:
JavaParserLabeled.InterfaceDeclarationContext):
        if self.name == ctx.IDENTIFIER().getText():
            self.body = ctx.getText()
            self.branches = ctx.parentCtx.children

    def enterEnumDeclaration(self, ctx: JavaParserLabeled.EnumDeclarationContext):
        if self.name == ctx.IDENTIFIER().getText():
            self.body = ctx.getText()
            self.branches = ctx.parentCtx.children

```

### Add Imported Entity:

At first we will check if the entity is a built in java class the kindid of it should be set to the id of "Java Unknown Class Type Member", also this kind of class doesn't have a parent. If the entity isn't built in we will set its kind using get kind method. We will also get the parent of this entity from the

functions explained. At last we will get the class body to add the contents of that class to our table.

```
def add_imported_entity(i, files):
    if i['is_built_in']:
        imported_entity, _ = EntityModel.get_or_create(
            _kind=KindModel.get_or_none(_name="Java Unknown Class Type
Member").get_id(),
            _parent=None,
            _name=i['imported_class_name'],
            _longname=i['imported_class_longname'],
        )
    else:
        parent_entity, parent_file_path = get_parent(i['imported_class_file_name'], files)
        prefixes, class_body, kind = get_imported_entity(parent_file_path)
        entity_kind = get_kind_name(prefixes, kind)
        imported_entity, _ = EntityModel.get_or_create(
            _kind=KindModel.get_or_none(_name=entity_kind).get_id(),
            _parent=parent_entity.get_id(),
            _name=i['imported_class_name'],
            _longname=i['imported_class_longname'],
            _contents=class_body,
        )
    return imported_entity
```

### Get Imported Entity:

This method returns the imported entity attributes. At first we get the parse tree of the path we have, and we also pass the file path to the imported entity listener explained before. After getting our listener, we can start walking in the tree. As we walk in the tree we check if the kind of our branch is class, interface or enum. Therefore set the corresponding attributes and return it.

```
def get_imported_entity(file_path):
    tree = get_parse_tree(file_path)
    listener = ImportedEntityListener(Path(file_path).stem)
    walker = ParseTreeWalker()
```

```

walker.walk(listener=listener, t=tree)

prefixes = ""
kind = ""
for branch in listener.branches:
    if type(branch) == JavaParserLabeled.ClassDeclarationContext:
        kind = "Class"
        break
    elif type(branch) == JavaParserLabeled.InterfaceDeclarationContext:
        kind = "Interface"
        break
    elif type(branch) == JavaParserLabeled.EnumDeclarationContext:
        kind = "Enum Class"
        break
    prefixes += branch.getText() + " "
return prefixes, listener.body, kind

```

### Kind Type:

This method helps us figure out the exact kind name for each entity. After checking the prefixes we can assemble them in one string(in the right order) as the final name.

```

def get_kind_name(prefixes, kind):
    p_static = ""
    p_abstract = ""
    p_generic = ""
    p_type = "Type"
    p_visibility = "Default"
    p_member = "Member"

    if "static" in prefixes:
        p_static = "Static"

    if "generic" in prefixes:
        p_generic = "Generic"

    if "abstract" in prefixes:
        p_abstract = "Abstract"

```



```

elif "final" in prefixes:
    p_abstract = "Final"

if "private" in prefixes:
    p_visibility = "Private"
elif "public" in prefixes:
    p_visibility = "Public"
elif "protected" in prefixes:
    p_visibility = "Protected"

if kind == "Interface":
    p_member = ""

if kind == "Method":
    p_type = ""

s = f"Java {p_static} {p_abstract} {p_generic} {kind} {p_type} {p_visibility}
{p_member}"
s = " ".join(s.split())
return s

```

### Add references:

This function gets the imported and importing entity and adds them in the final table of our database.

```

def add_references(importing_ent, imported_ent, ref_dict):
    ref, _ = ReferenceModel.get_or_create(
        _kind=KindModel.get_or_none(_name="Java Import").get_id(),
        _file=importing_ent.get_id(),
        _line=ref_dict['line'],
        _column=ref_dict['column'],
        _ent=imported_ent.get_id(),
        _scope=importing_ent.get_id(),
    )
    inverse_ref, _ = ReferenceModel.get_or_create(
        _kind=KindModel.get_or_none(_name="Java Importby").get_id(),
        _file=importing_ent.get_id(),
        _line=ref_dict['line'],

```

```

        _column=ref_dict['column'],
        _ent=importing_ent.get_id(),
        _scope=imported_ent.get_id(),
    )

```

### Main:

First we make an instance of the project and call `init_db`. After setting our arrays with the java files found, we need to match the corresponding name and path of each file, then for each file we find the importing entity as explained before. We need to use `get_parse_tree` to get our tree, then use the listener we made for import before so we can walk in our tree. Now we have our repository which is the list with all of our import objects. For each item in that list we need to find the imported entity using the `add_imported_entity` method explained before. At last we have our importing and imported entity and also the matching repository object in the listener so we pass them to `add_references` which adds them to the final references table in our database.

```

def main():
    info = get_project_info(PRJ_INDEX, REF_NAME)
    p = Project(info['DB_PATH'], info['PROJECT_PATH'], info['PROJECT_NAME'])
    p.init_db()
    p.get_java_files()

    for file_name, file_path in p.files:
        importing_entity = add_java_file_entity(file_path, file_name)

        tree = get_parse_tree(file_path)
        listener = ImportListener(p.files)
        walker = ParseTreeWalker()
        walker.walk(listener, tree)

    for i in listener.repository:
        imported_entity = add_imported_entity(i, p.files)
        add_references(importing_entity, imported_entity, i)

```

## Modify (Deref) Partial and Modifyby (Deref) Partial

We have defined most of the initial functions and set ups in the previous sections, In this section we will describe the functions that are specifically designed to find the modify partial command and shape our reference table in the database.

### Get prefixes:

In order to find the name of our entity kind with get kind name method(explained in the previous parts) we need to have these prefixes.

```
def get_prefixes(ctx, ctx_type):
    branches = ctx.parentCtx.children
    prefixes = ""
    for branch in branches:
        if type(branch).__name__ == ctx_type:
            break
    prefixes += branch.getText() + " "
    return prefixes
```

### Class Listener:

This class, is out listener which checks through our file and adds the classes to the database table of entities.

```
class ClassListener(JavaParserLabeledListener):
    def __init__(self, files, file_name):
        self.files = files
        self.file_name = file_name

    def enterClassDeclaration(self, ctx: JavaParserLabeled.ClassDeclarationContext):
        parent_entity, parent_file_path = get_parent(self.file_name, self.files)
        prefixes = get_prefixes(ctx, "ClassDeclarationContext")
        kind_name = get_kind_name(prefixes, kind="Class")
```

```

obj, _ = EntityModel.get_or_create(
    _kind=KindModel.get_or_none(_name=kind_name).get_id(),
    _parent=parent_entity.get_id(),
    _name=ctx.IDENTIFIER().getText(),
    _longname=parent_file_path,
    _contents=ctx.getText(),
)

```

### Modify deref Listener:

This is the main part of this section's code. Search scope checks current until it finds a type name that is in the type names list; and if it finds a class or method it returns current. In the following two methods we check whether the scope is a class or a method. Therefore we add the entity to our table setting the attributes kind, parent, name, long name using the methods explained before. At last we also add the id and content and return the object. The enter expressions 6 and 21 are for “=” and “+,-,\*,...” expression which can modify a variable. Last but not least we have the main function which after being called from the enter expressions, checks whether there is an opening for a scope and uses search scope to declare that. After that we check if the scope is a class or a method using those two functions explained earlier; and set the correct name and entity kind for each one of them and pass it on to add reference for the final result to be added to our database.

```

class ModifyListener(JavaParserLabeledListener):
    def __init__(self, files, file_name):
        self.files = files
        self.file_name = file_name

    @staticmethod
    def search_scope(ctx, type_names):
        # Traverse bottom up until reaching a class or method
        current = ctx.parentCtx
        while current is not None:
            type_name = type(current).__name__
            if type_name in type_names:

```

```

        return current
    current = current.parentCtx
    return None

def make_scope_class(self, ctx, file_name):
    prefixes = get_prefixes(ctx, "ClassDeclarationContext")
    kind_name = get_kind_name(prefixes, kind="Class")
    kind_id = KindModel.get_or_none(_name=kind_name).get_id()
    name = ctx.IDENTIFIER().getText()
    parent_entity, parent_file_path = get_parent(file_name, self.files)
    content = ctx.getText()

    obj = EntityModel.get_or_none(
        _kind=KindModel.get_or_none(_name=kind_name).get_id(),
        _parent=parent_entity.get_id(),
        _name=name,
        _longname=parent_file_path,
    )
    return {
        "id": obj.get_id(),
        "kind_id": kind_id,
        "parent_id": parent_entity.get_id(),
        "name": name,
        "longname": parent_file_path,
        "content": content
    }

def make_scope_method(self, ctx, file_name):
    prefixes = get_prefixes(ctx, "MethodDeclarationContext")
    kind_name = get_kind_name(prefixes, kind="Method")
    kind_id = KindModel.get_or_none(_name=kind_name).get_id()
    name = ctx.IDENTIFIER().getText()
    content = ctx.getText()
    parent_ctx = self.search_scope(ctx, ["ClassDeclarationContext"])
    parent_entity = self.make_scope_class(parent_ctx, file_name)

    obj, _ = EntityModel.get_or_create(
        _kind=kind_id,

```

```

        _parent=parent_entity['id'],
        _name=name,
        _longname=f"{parent_entity['name']}.{name}",
        _contents=content,
    )
    return {
        "id": obj.get_id(),
        "kind_id": kind_id,
        "parent_id": parent_entity['id'],
        "name": name,
        "longname": f"{parent_entity['name']}.{name}",
        "content": content
    }

def enterExpression6(self, ctx: JavaParserLabeled.Expression6Context):
    self.modify_deref_partial(ctx)

def enterExpression21(self, ctx: JavaParserLabeled.Expression21Context):
    self.modify_deref_partial(ctx)

def modify_deref_partial(self, ctx):
    lhs_text = ctx.children[0].getText()
    scope = None
    if "[" in lhs_text and "]" in lhs_text:
        var_name = ctx.children[0].children[0].getText()
        var_entity = add_var_entity(var_name)

        line, col = str(ctx.start).split(",")[3][:-1].split(':')

        file_entity, _ = get_parent(self.file_name, self.files)
        file_id = file_entity.get_id()

        ref_dict = { 'line': line, 'column': col, 'file_id': file_id, 'text': ctx.getText() }

        scope_ctx = self.search_scope(ctx, ["ClassDeclarationContext",
"MethodDeclarationContext"])
        if type(scope_ctx).__name__ == "ClassDeclarationContext":
            scope = self.make_scope_class(scope_ctx, self.file_name)

```

```

elif type(scope_ctx).__name__ == "MethodDeclarationContext":
    scope = self.make_scope_method(scope_ctx, self.file_name)

    add_references(scope, var_entity, ref_dict)

```

### Add variable entity:

Add each variable entity to the database table of entities.

```

def add_var_entity(var_name):
    obj, _ = EntityModel.get_or_create(
        _kind=KindModel.get_or_none(_name="Java Unresolved Variable").get_id(),
        _name=var_name,
        _longname=var_name,
    )
    return {
        "id": obj.get_id(),
        "kind_id": KindModel.get_or_none(_name="Java Unresolved Variable").get_id(),
        "name": var_name,
        "longname": var_name,
    }

```

### Add references:

Just as we explained in the last section, this method adds the final objects to our reference table in the database. The kind id is set to either “Java Modify Deref Partial” or “Java Modifyby Deref Partial”. The file, line and column in which the modifying has happened are all set in their attributes. We have also added the entity which does the action and the scope the action has taken place on.

```

def add_references(scope, ent, ref_dict):
    ref, _ = ReferenceModel.get_or_create(
        _kind=KindModel.get_or_none(_name="Java Modify Deref Partial").get_id(),
        _file=ref_dict['file_id'],
        _line=ref_dict['line'],
    )

```

```

        _column=ref_dict['column'],
        _ent=ent['id'],
        _scope=scope['id'],
    )
    inverse_ref, _ = ReferenceModel.get_or_create(
        _kind=KindModel.get_or_none(_name="Java Modifyby Deref Partial").get_id(),
        _file=ref_dict['file_id'],
        _line=ref_dict['line'],
        _column=ref_dict['column'],
        _ent=scope['id'],
        _scope=ent['id'],
    )

```

### Main:

After creating an instance of the project and initializing the database and setting the java files; we get then parse tree of each file and traverse through it. We make our listeners and walk through them, as explained before the expressions and scopes of the file are checked for a modified variable in the scope mentioned. After detecting and setting the attributes of each modify deref partial that has happened, we have the final results as a table.

#### def main():

```

    info = get_project_info(PRJ_INDEX, REF_NAME)
    p = Project(info['DB_PATH'], info['PROJECT_PATH'], info['PROJECT_NAME'])
    p.init_db()
    p.get_java_files()

```

#### for file\_name, file\_path in p.files:

```

    tree = get_parse_tree(file_path)
    walker = ParseTreeWalker()

```

```

    class_listener = ClassListener(p.files, file_name)
    listener = ModifyListener(p.files, file_name)
    walker.walk(class_listener, tree)
    walker.walk(listener, tree)

```



## Counting the number of statements

For this section we have to count the number of statements. There are two types of statements, declaration and executable. For doing so we used our phase one codes such as our listeners to walk through the tree. We can see the codes for the two other types in the section below.

### Codes:

As our tests resulted packages, imports and other things that count as a statement are added as a enter declaration( there are comments added before each section to label them). Also we have a method named update repository that gets the CTX and the count of the statements we have. Then in this method we add the statement counter with the increment count that has been passed on to the method. Then we go to find scope method so that we can get the type of its father. Just as we used to go up the tree in phase one to reach a class or method but now we include all the statements in all of the methods of one class; therefore we added them to a list. Then we have to check the type so we can add the right name to our dictionary. Also for the two types that are both included as statements we just divided the types which were counted in each case( according to the documentation).

### Countstmt:

```
class StatementListener(JavaParserLabeledListener):
    def __init__(self, files):
        self.repository = {}
        self.files = files
        self.counter = 0

    def enterPackageDeclaration(self, ctx:
JavaParserLabeled.PackageDeclarationContext):
        self.counter += 1
```

```

def enterImportDeclaration(self, ctx: JavaParserLabeled.ImportDeclarationContext):
    self.counter += 1

def enterInterfaceMethodDeclaration(self, ctx:
JavaParserLabeled.InterfaceMethodDeclarationContext):
    self.update_repository(ctx, 1)

def enterMethodDeclaration(self, ctx: JavaParserLabeled.MethodDeclarationContext):
    self.update_repository(ctx, 1)

def enterFieldDeclaration(self, ctx: JavaParserLabeled.FieldDeclarationContext):
    self.update_repository(ctx, 1)

def enterLocalVariableDeclaration(self, ctx:
JavaParserLabeled.LocalVariableDeclarationContext):
    self.update_repository(ctx, 1)

# for
def enterStatement3(self, ctx: JavaParserLabeled.Statement3Context):
    if len(ctx.children[2].children) < 3:
        self.update_repository(ctx, 2)
    else:
        self.update_repository(ctx, 3)
    for i in ctx.children:
        if i == ';':
            self.update_repository(ctx, 1)

# semi-colon
def enterStatement14(self, ctx: JavaParserLabeled.Statement14Context):
    self.update_repository(ctx, 1)

# call
def enterStatement15(self, ctx: JavaParserLabeled.Statement15Context):
    self.update_repository(ctx, 1)

# return
def enterStatement10(self, ctx: JavaParserLabeled.Statement10Context):
    self.update_repository(ctx, 1)

```

```

# break
def enterStatement12(self, ctx: JavaParserLabeled.Statement12Context):
    self.update_repository(ctx, 1)

# throw
def enterStatement11(self, ctx: JavaParserLabeled.Statement11Context):
    self.update_repository(ctx, 1)

# continue
def enterStatement13(self, ctx: JavaParserLabeled.Statement13Context):
    self.update_repository(ctx, 1)

def enterAnnotationMethodOrConstantRest0(self, ctx:
JavaParserLabeled.AnnotationMethodOrConstantRest0Context):
    self.update_repository(ctx, 1)

def update_repository(self, ctx, increment):
    self.counter += increment
    keys = get_keys(ctx)
    for key in keys:
        if key in self.repository:
            self.repository[key] += increment
        else:
            new_dict = {key: 0}
            new_dict[key] += increment
            self.repository.update(new_dict)

if __name__ == '__main__':
    stmt_main(PRJ_INDEX, StatementListener, METRIC_NAME, LAST_LOG)

```

## Countstmtdecl:

```

class StatementListener(JavaParserLabeledListener):
    def __init__(self, files):
        self.repository = {}
        self.files = files
        self.counter = 0

    def enterPackageDeclaration(self, ctx:
JavaParserLabeled.PackageDeclarationContext):
        self.counter += 1

    def enterImportDeclaration(self, ctx: JavaParserLabeled.ImportDeclarationContext):
        self.counter += 1

    def enterInterfaceMethodDeclaration(self, ctx:
JavaParserLabeled.InterfaceMethodDeclarationContext):
        self.update_repository(ctx, 1)

    def enterMethodDeclaration(self, ctx: JavaParserLabeled.MethodDeclarationContext):
        self.update_repository(ctx, 1)

    def enterFieldDeclaration(self, ctx: JavaParserLabeled.FieldDeclarationContext):
        self.update_repository(ctx, 1)

    def enterLocalVariableDeclaration(self, ctx:
JavaParserLabeled.LocalVariableDeclarationContext):
        self.update_repository(ctx, 1)

    # for
    def enterStatement3(self, ctx: JavaParserLabeled.Statement3Context):
        self.update_repository(ctx, 1)

    # semi-colon
    def enterStatement14(self, ctx: JavaParserLabeled.Statement14Context):
        self.update_repository(ctx, 1)

```

```

# call
def enterStatement15(self, ctx: JavaParserLabeled.Statement15Context):
    self.update_repository(ctx, 1)

def enterAnnotationMethodOrConstantRest0(self, ctx:
JavaParserLabeled.AnnotationMethodOrConstantRest0Context):
    self.update_repository(ctx, 1)

def update_repository(self, ctx, increment):
    self.counter += increment
    keys = get_keys(ctx)
    for key in keys:
        if key in self.repository:
            self.repository[key] += increment
        else:
            new_dict = {key: 0}
            new_dict[key] += increment
            self.repository.update(new_dict)

```

### Countstmtexte:

```

class StatementListener(JavaParserLabeledListener):
    def __init__(self, files):
        self.repository = { }
        self.files = files
        self.counter = 0

# return
def enterStatement10(self, ctx: JavaParserLabeled.Statement10Context):
    self.update_repository(ctx, 1)

# for
def enterStatement3(self, ctx: JavaParserLabeled.Statement3Context):
    self.update_repository(ctx, 2)

# break
def enterStatement12(self, ctx: JavaParserLabeled.Statement12Context):

```

```

        self.update_repository(ctx, 1)

# throw
def enterStatement11(self, ctx: JavaParserLabeled.Statement11Context):
    self.update_repository(ctx, 1)

# continue
def enterStatement13(self, ctx: JavaParserLabeled.Statement13Context):
    self.update_repository(ctx, 1)

def enterAnnotationMethodOrConstantRest0(self, ctx:
JavaParserLabeled.AnnotationMethodOrConstantRest0Context):
    self.update_repository(ctx, 1)

def update_repository(self, ctx, increment):
    self.counter += increment
    keys = get_keys(ctx)
    for key in keys:
        if key in self.repository:
            self.repository[key] += increment
        else:
            new_dict = {key: 0}
            new_dict[key] += increment
            self.repository.update(new_dict)

```

## Test:

```

def test_understand_kinds():
    info = get_project_info(PRJ_INDEX, REF_NAME)
    db = und.open(info['DB_PATH'])
    my_set = set()
    for ent in db.ents():
        cycle = ent.metric(['CountStmtExe']).get('CountStmtExe', 0)
        ent_kind = ent.kind()
        if cycle and ent_kind.__repr__().startswith("Java"):
            if str(ent_kind) != "Package":
                if cycle and 'Java' in ent_kind.__repr__():

```

```

        my_set.add(ent_kind.__repr__())
        print(ent_kind)
        print(ent_kind.__repr__())
        print(ent.longname())
        print(cycle)
        print("-" * 25)

    for i in my_set:
        print(i)

if __name__ == '__main__':
    test_understand_kinds()

```

## Cyclomatic

For this section we have to count the cyclomatic for each method and then each project that is the sum of the methods. As shown in the picture before there is a formula to count the cyclomatic for each method. We need to count And, Or, Catch, Do, For, If, ?, While, Switch and Case + 1.

### Codes:

At first we must check if there the statements mentioned earlier in our code, once we reach one we will get the context and start a bottom up traverse till we reach the parent method( using search scope). At last we will build the final result dictionary consisting of the key set to the name of the methods and the values set to the cyclomatic number of each.

### Code:

```

def make_enum_scope():
    return [
        {
            "kind": 'Java Method Public Member',
            "name": 'values',

```

```

    "longname": "",
    "cyclomatic": 1
  },
  {
    "kind": 'Java Method Public Member',
    "name": 'valueOf',
    "longname": "",
    "cyclomatic": 1
  }
]

```

```

def make_method_scope(ctx):
    prefixes = get_method_prefixes(ctx)
    name = ""
    kind = ""
    if type(ctx).__name__ == "MethodDeclarationContext":
        name = ctx.IDENTIFIER().getText()
        if name == "main":
            is_main = True
        else:
            is_main = False
        kind = get_kind_name(prefixes, is_main=is_main)
    elif type(ctx).__name__ == "GenericMethodDeclarationContext":
        name = ctx.children[1].IDENTIFIER().getText()
        kind = get_kind_name(prefixes, is_generic=True)
    elif type(ctx).__name__ == "ConstructorDeclarationContext":
        name = ctx.IDENTIFIER().getText()
        kind = get_kind_name(prefixes, is_constructor=True)
    elif type(ctx).__name__ == "GenericConstructorDeclarationContext":
        name = ctx.children[1].IDENTIFIER().getText()
        kind = get_kind_name(prefixes, is_constructor=True, is_generic=True)
    elif type(ctx).__name__ == "LambdaExpressionContext":
        name = "(lambda_expr)"
        kind = get_kind_name(prefixes, is_lambda=True)

    return {
        "kind": kind,

```



```

    "name": name,
    "longname": ""
}

```

```

def get_kind_name(prefixes, is_constructor=False, is_lambda=False, is_generic=False,
is_main=False):

```

```

    p_static = ""
    p_final = ""
    p_generic = ""
    p_main = ""

```

```

    if "static" in prefixes:
        p_static = "Static"

```

```

    if "final" in prefixes:
        p_final = "Final"

```

```

    if is_generic:
        p_generic = "Generic"

```

```

    if "private" in prefixes:
        p_visibility = "Private"
    elif "public" in prefixes:
        p_visibility = "Public"
    elif "protected" in prefixes:
        p_visibility = "Protected"
    else:
        p_visibility = "Default"

```

```

    if is_main:
        p_main = "Main"

```

```

    if is_constructor:
        s = f"Java Method Constructor Member {p_visibility}"
        s = " ".join(s.split())
        return s

```

```

    elif is_lambda:

```

```

s = f"Java Method Lambda"
s = " ".join(s.split())
return s

else:
    s = f"Java {p_static} {p_final} {p_generic} Method {p_visibility} {p_main}
Member"
    s = " ".join(s.split())
    return s

def get_method_ctx(ctx):
    # Traverse bottom up until reaching a method
    current = ctx.parentCtx
    while current is not None:
        type_name = type(current).__name__
        if type_name in ['MethodDeclarationContext',
                        'GenericMethodDeclarationContext',
                        'ConstructorDeclarationContext',
                        'GenericConstructorDeclarationContext']:
            return current
        current = current.parentCtx
    return None

class CyclomaticListener(JavaParserLabeledListener):
    def __init__(self):
        # repository of ctx
        self.repository = []
        self.project_cyclomatic = 0

    def update_repository(self, ctx, kind=None):
        if kind:
            scope_ctx = ctx

        else:
            scope_ctx = get_method_ctx(ctx)
            if scope_ctx is None:

```

```

        return

    parent_scope_ctx = scope_ctx.parentCtx
    if type(parent_scope_ctx).__name__ in [
        'GenericMethodDeclarationContext',
        'GenericConstructorDeclarationContext'
    ]:
        scope_ctx = parent_scope_ctx

    prefixes = get_method_prefixes(scope_ctx)
    if 'abstract' not in prefixes:
        self.repository.append(scope_ctx)
        self.project_cyclomatic += 1

    def enterGenericMethodDeclaration(self, ctx:
    JavaParserLabeled.GenericMethodDeclarationContext):
        self.update_repository(ctx, kind='GenericMethodDeclarationContext')

    def enterMethodDeclaration(self, ctx: JavaParserLabeled.MethodDeclarationContext):
        # Parent can be genericMethodDeclaration
        if type(ctx.parentCtx).__name__ == "GenericMethodDeclarationContext":
            return
        self.update_repository(ctx, kind='MethodDeclarationContext')

    def enterGenericConstructorDeclaration(self, ctx:
    JavaParserLabeled.GenericConstructorDeclarationContext):
        self.update_repository(ctx, kind='GenericConstructorDeclarationContext')

    def enterConstructorDeclaration(self, ctx:
    JavaParserLabeled.ConstructorDeclarationContext):
        # Parent can be genericConstructorDeclaration
        if type(ctx.parentCtx).__name__ == "GenericConstructorDeclarationContext":
            return
        self.update_repository(ctx, kind='ConstructorDeclarationContext')

    def enterEnumDeclaration(self, ctx: JavaParserLabeled.EnumDeclarationContext):
        # valueOf and values for Enums
        self.repository.append(ctx)

```

```

        self.project_cyclomatic += 2

def enterLambdaExpression(self, ctx: JavaParserLabeled.LambdaExpressionContext):
    self.repository.append(ctx)
    self.project_cyclomatic += 1

# while
def enterStatement4(self, ctx: JavaParserLabeled.Statement4Context):
    self.update_repository(ctx)

# ternary
def enterExpression20(self, ctx: JavaParserLabeled.Expression20Context):
    self.update_repository(ctx)

# if
def enterStatement2(self, ctx: JavaParserLabeled.Statement2Context):
    self.update_repository(ctx)

# for
def enterStatement3(self, ctx: JavaParserLabeled.Statement3Context):
    self.update_repository(ctx)

# do
def enterStatement5(self, ctx: JavaParserLabeled.Statement5Context):
    self.update_repository(ctx)

# catch
def enterCatchClause(self, ctx: JavaParserLabeled.CatchClauseContext):
    self.update_repository(ctx)

# case
def enterSwitchLabel(self, ctx: JavaParserLabeled.SwitchLabelContext):
    if ctx.children[0].getText() == "case":
        self.update_repository(ctx)

def main():
    info = get_project_info(PRJ_INDEX)
    p = Project(info['PROJECT_PATH'], info['PROJECT_NAME'])

```

```

p.get_java_files()

walker = ParseTreeWalker()

cyclomatic_listener = CyclomaticListener()
cyclomatic_list = []
ent_kind_set = set()

for file_name, file_path in p.files:
    tree = get_parse_tree(file_path)
    walker.walk(cyclomatic_listener, tree)

cyclomatic_counter = Counter(cyclomatic_listener.repository)

for ctx in cyclomatic_counter:
    cyclomatic = cyclomatic_counter[ctx]

    if type(ctx).__name__ == 'EnumDeclarationContext':
        cyclomatic_list.extend(make_enum_scope())
    else:
        cyclomatic_obj = make_method_scope(ctx)
        cyclomatic_obj['cyclomatic'] = cyclomatic
        cyclomatic_list.append(cyclomatic_obj)
        ent_kind_set.add(cyclomatic_obj['kind'])

report_cyclomatic(cyclomatic_listener.project_cyclomatic, ent_kind_set,
cyclomatic_list)

if __name__ == '__main__':
    main()

```

## Test:

```

def test():
    info = get_project_info(PRJ_INDEX, REF_NAME)
    db = und.open(info['DB_PATH'])

```

```

project_cyclomatic = 0
cyclomatic_list = []
ent_kind_set = set()

for ent in db.ents():
    ent_kind = ent.kind().__repr__()
    ent_cyclomatic = ent.metric(['Cyclomatic']).get('Cyclomatic', 0)

    if ent_cyclomatic and ent_kind.startswith("Java"):
        project_cyclomatic += ent_cyclomatic
        cyclomatic_list.append({
            'kind': ent_kind,
            'name': ent.simplename(),
            'cyclomatic': ent_cyclomatic,
            'longname': ent.longname()
        })
        ent_kind_set.add(ent_kind)

report_cyclomatic(project_cyclomatic, ent_kind_set, cyclomatic_list)

if __name__ == '__main__':
    test()

```

## Helper Methods:

We have united the methods that were needed in more than one file.

## Codes:

# Common Helper Functions

```

class Project:
    def __init__(self, project_dir, project_name=None):
        self.project_dir = project_dir
        self.project_name = project_name
        self.files = []

    def get_java_files(self):
        for dir_path, _, file_names in os.walk(self.project_dir):
            for file in file_names:

```

```

        lowercase_file = str(file).lower()
        if lowercase_file.endswith('.java'):
            path = os.path.join(dir_path, file)
            path = path.replace("/", "\\")
            path = os.path.abspath(path)
            self.files.append((file, path))

def get_project_info(index, ref_name=None):
    project_names = [
        'calculator_app',    # 0
        'JSON',              # 1
        'testing_legacy_code', # 2
        'TheAlgorithms',     # 3
        'jhotdraw-develop',  # 4
        'xerces2j',          # 5
        'jvlt-1.3.2',        # 6
        'jfreechart',        # 7
        'ganttproject',      # 8
        '105_freemind',      # 9
        'custom'             # 10
    ]
    project_name = project_names[index]
    db_path = f"../../databases/{ref_name}/{project_name}"
    if ref_name == "origin":
        db_path = db_path + ".udb"
    else:
        db_path = db_path + ".oudb"
    project_path = f"../../benchmarks/{project_name}"

    db_path = os.path.abspath(db_path)
    project_path = os.path.abspath(project_path)

    return {
        'PROJECT_NAME': project_name,
        'DB_PATH': db_path,
        'PROJECT_PATH': project_path,
    }

def get_parse_tree(file_path):
    file = FileStream(file_path, encoding="utf-8")
    lexer = JavaLexer(file)
    tokens = CommonTokenStream(lexer)
    parser = JavaParserLabeled(tokens)
    return parser.compilationUnit()

def get_parent(parent_file_name, files):
    file_names, file_paths = zip(*files)

```

```

parent_file_index = file_names.index(parent_file_name)
parent_file_path = file_paths[parent_file_index]
return parent_file_path

def report_metric(project_metric_count, ent_kind_set, project_metric_list, metric_name):
    sorted_list = sorted(project_metric_list, key=lambda d: (d['val'], d['name']))
    for e in sorted_list:
        print({
            'val': e['val'],
            'name': e['name'],
            'kind': e['kind'],
            # 'ln': e['longname']
        })
    print("-" * 25)

    print(f"Entities with {metric_name}: {len(ent_kind_set)}")
    for i in sorted(ent_kind_set):
        print(i)
    print("-" * 25)

    print(f"Project total {metric_name}: {project_metric_count}")

# Prefix producers

def get_class_prefixes(ctx, ctx_type):
    branches = ctx.parentCtx.children
    prefixes = ""
    for branch in branches:
        if type(branch).__name__ == ctx_type:
            break
        prefixes += branch.getText() + " "
    return prefixes

def get_method_prefixes(ctx):
    access_branches = ctx.parentCtx.parentCtx.children
    type_branches = ctx.children
    prefixes = []

    for branch in access_branches:
        if type(branch).__name__ == "ModifierContext":
            prefixes.append(branch.getText())

    for branch in type_branches:
        if type(branch).__name__ == "TypeTypeOrVoidContext":
            prefixes.append(branch.getText())

```



`return` prefixes

# Statement Helper Functions

```
def stmt_main(prj_index, listener_class, metric_name, last_log=False):
    info = get_project_info(prj_index)
    p = Project(info['PROJECT_PATH'], info['PROJECT_NAME'])
    p.get_java_files()
    walker = ParseTreeWalker()

    ent_kind_set = {'Java File'}
    project_metric_list = []
    project_metric_counter = 0

    for file_name, file_path in p.files:
        tree = get_parse_tree(file_path)
        listener = listener_class(p.files)
        walker.walk(listener, tree)

    file_metric_dict = listener.repository
    file_metric_count = listener.counter

    for ent, count in file_metric_dict.items():
        ent_name = ent.split('$$$')[0]
        remain = ent.split('$$$')[1]
        ent_kind, ent_longname = remain.split('-', 1)
        if str(ent_name).startswith('package'):
            ent_longname = ent_longname.replace('package', '')
            ent_longname = ent_longname.replace('; class', '.')
            ent_longname = ent_longname.replace(' ', '')

        new_metric = {
            'val': count,
            'name': ent_name,
            'kind': ent_kind,
            'longname': ent_longname
        }
        if not last_log:
            print({i: new_metric[i] for i in new_metric if i != 'longname'})

        project_metric_list.append(new_metric)
        project_metric_counter += count
        ent_kind_set.add(ent_kind)

    new_metric = {
        'val': file_metric_count,
        'name': file_name,
        'kind': 'Java File',
        'longname': file_path
```

```

    }
    if not last_log:
        print({i: new_metric[i] for i in new_metric if i != 'longname'})

    project_metric_list.append(new_metric)
    project_metric_counter += file_metric_count

if last_log:
    report_metric(project_metric_counter, ent_kind_set, project_metric_list, metric_name)

def get_keys(ctx):
    result = find_scope(ctx)
    keys = []
    for res in result:
        if res['static_type'] != "":
            key = str(res['method_name']) + '$$$' + str(res['kind_name']) + '-' + str(res['access_type']) + '' +
str(res['static_type']) + '' \
            + str(res['return_type']) + '' + str(res['method_name'])
        else:
            key = str(res['method_name']) + '$$$' + str(res['kind_name']) + '-' + str(res['access_type']) + '' \
            + str(res['return_type']) + '' + str(res['method_name'])
        keys.append(key)
    return keys

# Scope makers

def make_scope_interface(ctx):
    prefixes = get_class_prefixes(ctx, "InterfaceDeclarationContext")
    kind_name = get_kind_name(prefixes, kind="Class")
    class_name = ctx.children[1]
    return_type = ctx.children[0].getText()
    access_type = ctx.parentCtx.parentCtx.children[0].getText()
    return {
        'kind_name': kind_name,
        'method_name': class_name,
        'return_type': return_type,
        'access_type': access_type,
        'static_type': ""
    }

def make_scope_lambda(ctx):
    prefixes = get_class_prefixes(ctx, "LambdaExpressionContext")
    kind_name = get_kind_name(prefixes, kind="Class", is_lambda=True)
    class_name = ctx.children[1]
    return_type = ctx.children[0].getText()
    access_type = ctx.parentCtx.parentCtx.children[0].getText()
    if ctx.parentCtx.parentCtx.children[1].getText() == 'static':

```

```

        static_type = ctx.parentCtx.parentCtx.children[1].getText()
    else:
        static_type = ""
    return {
        'kind_name': kind_name,
        'method_name': class_name,
        'return_type': return_type,
        'access_type': access_type,
        'static_type': static_type
    }

def make_scope_class(ctx):
    prefixes = get_class_prefixes(ctx, "ClassDeclarationContext")
    kind_name = get_kind_name(prefixes, kind="Class")
    class_name = ctx.children[1].getText()
    return_type = ctx.children[0].getText()
    access_type = ctx.parentCtx.parentCtx.children[0].getText()
    static_type = ""
    if len(ctx.parentCtx.parentCtx.children) > 1:
        if ctx.parentCtx.parentCtx.children[1].getText() == 'static':
            static_type = ctx.parentCtx.parentCtx.children[1].getText()
    return {
        'kind_name': kind_name,
        'method_name': class_name,
        'return_type': return_type,
        'access_type': access_type,
        'static_type': static_type
    }

def make_scope_method(ctx):
    prefixes = get_method_prefixes(ctx)
    kind_name = get_kind_name(prefixes, kind="Method")
    method_name = ctx.children[1]
    return_type = ctx.children[0].getText()
    access_type = ctx.parentCtx.parentCtx.children[0].getText()
    static_type = ""
    if len(ctx.parentCtx.parentCtx.children) > 1:
        if ctx.parentCtx.parentCtx.children[1].getText() == 'static':
            static_type = ctx.parentCtx.parentCtx.children[1].getText()
    return {
        'kind_name': kind_name,
        'method_name': method_name,
        'return_type': return_type,
        'access_type': access_type,
        'static_type': static_type
    }

```

```

def make_scope_constructor(ctx):
    prefixes = get_method_prefixes(ctx)
    kind_name = get_kind_name(prefixes, kind="Method", is_constructor=True)
    method_name = ctx.IDENTIFIER().getText()
    return_type = ctx.children[0].getText()
    access_type = ctx.parentCtx.parentCtx.children[0].getText()
    static_type = ""
    if len(ctx.parentCtx.parentCtx.children) > 1:
        if ctx.parentCtx.parentCtx.children[1].getText() == 'static':
            static_type = ctx.parentCtx.parentCtx.children[1].getText()
    return {
        'kind_name': kind_name,
        'method_name': method_name,
        'return_type': return_type,
        'access_type': access_type,
        'static_type': static_type
    }

# Scope finders

def search_scope(ctx, type_names):
    # Traverse bottom up until reaching a class or method
    scope_list = []
    current = ctx
    while current is not None:
        type_name = type(current).__name__
        if type_name in type_names:
            scope_list.append(current)
            current = current.parentCtx
    return scope_list

def find_scope(ctx):
    scope = []
    if str(ctx.children[0]) == 'package':
        return [{
            'kind_name': 'Java Package',
            'method_name': ctx.children[1].getText(),
            'return_type': "",
            'access_type': "",
            'static_type': ""
        }]
    scope_ctx = search_scope(ctx, ["ClassDeclarationContext", "MethodDeclarationContext",
                                   "InterfaceDeclarationContext", "AnnotationTypeDeclarationContext",
                                   "ConstructorDeclarationContext", "LambdaExpressionContext"])
    for item in scope_ctx:
        if type(item).__name__ == "ClassDeclarationContext":
            scope.append(make_scope_class(item))
        elif type(item).__name__ == "ConstructorDeclarationContext":

```

```

        scope.append(make_scope_constructor(item))
    elif type(item).__name__ == "MethodDeclarationContext":
        scope.append(make_scope_method(item))
    elif type(item).__name__ == "InterfaceDeclarationContext":
        scope.append(make_scope_interface(item))
    elif type(ctx).__name__ == "LambdaExpressionContext":
        scope.append(make_scope_lambda(item))
return scope

def get_kind_name(prefixes, kind, is_constructor=False, is_lambda=False):
    p_static = ""
    p_abstract = ""
    p_generic = ""
    p_type = "Type"
    p_visibility = "Default"
    p_member = "Member"

    if "static" in prefixes:
        p_static = "Static"

    if "generic" in prefixes:
        p_generic = "Generic"

    if "abstract" in prefixes:
        p_abstract = "Abstract"
    elif "final" in prefixes:
        p_abstract = "Final"

    if is_constructor:
        s = f"Java Method Constructor Member {p_visibility}"
        s = " ".join(s.split())
        return s

    if "private" in prefixes:
        p_visibility = "Private"
    elif "public" in prefixes:
        p_visibility = "Public"
    elif "protected" in prefixes:
        p_visibility = "Protected"

    if kind == "Interface":
        p_member = ""

    if kind == "Method":
        p_type = ""

    elif is_lambda:
        s = f"Java Method Lambda"
        s = " ".join(s.split())

```

```

    return s

s = f"Java {p_static} {p_abstract} {p_generic} {kind} {p_type} {p_visibility} {p_member}"
s = " ".join(s.split())
return s

```

## Test:

```

def test():
    info = get_project_info(PRJ_INDEX, REF_NAME)
    db = und.open(info['DB_PATH'])

    project_metric_count = 0
    metric_list = []
    ent_kind_set = set()

    for ent in db.ents():
        ent_kind = ent.kind().__repr__()
        ent_metric = ent.metric([METRIC_NAME]).get(METRIC_NAME, 0)

        if ent_kind == "Java Package":
            continue

        if ent_metric and ent_kind.startswith("Java"):
            project_metric_count += ent_metric
            metric_list.append({
                'kind': ent_kind,
                'name': ent.simplename(),
                'val': ent_metric,
                'longname': ent.longname()
            })
            ent_kind_set.add(ent_kind)

    report_metric(project_metric_count, ent_kind_set, metric_list, METRIC_NAME)

```

## Tests and results

There were several benchmark projects tested and compared, so we will share the final tables of a few results for some of the projects. (The rest can be found on our source files).

**The resulted tables are as follows:**

### Import(calculator app example)

```

1. ref name: Java Import
2. ref scope: D:\Documents\University\Semesters\Term 8\Compiler\Project P1\benchmarks\calculator_app\src\com\calculator\app\display\println.java || kind: Java File
3. ref ent: com.calculator.app.display.print_success || kind: Java Class Type Public Member
4. file location: println.java || line: 3
-----
1. ref name: Java Import
2. ref scope: D:\Documents\University\Semesters\Term 8\Compiler\Project P1\benchmarks\calculator_app\src\com\calculator\app\display\println.java || kind: Java File
3. ref ent: com.calculator.app.display.print_fail || kind: Java Class Type Public Member
4. file location: println.java || line: 4
-----
1. ref name: Java Import
2. ref scope: D:\Documents\University\Semesters\Term 8\Compiler\Project P1\benchmarks\calculator_app\src\com\calculator\app\display\println.java || kind: Java File
3. ref ent: java.Lang.System || kind: Java Unknown Class Type Member
4. file location: println.java || line: 6
-----
1. ref name: Java Import
2. ref scope: D:\Documents\University\Semesters\Term 8\Compiler\Project P1\benchmarks\calculator_app\src\com\calculator\app\display\print_fail.java || kind: Java File
3. ref ent: java.Lang.System || kind: Java Unknown Class Type Member
4. file location: print_fail.java || line: 3
-----
1. ref name: Java Import
2. ref scope: D:\Documents\University\Semesters\Term 8\Compiler\Project P1\benchmarks\calculator_app\src\com\calculator\app\display\print_success.java || kind: Java File
3. ref ent: java.Lang.System || kind: Java Unknown Class Type Member
4. file location: print_success.java || line: 3
-----

```

# Compiler Project

DB Browser for SQLite - D:\Documents\University\Semesters\Term B\Compiler\Project P1\database\import\calculator\_app.oudb

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project Save Project Attach Database Close Database

Database Structure Browse Data Edit Pragnas Execute SQL

Table: entitymodel

|        | _id    | kind   | _arent | _name                | _longname                                  | _value | _type  | _contents                                 |
|--------|--------|--------|--------|----------------------|--|--------|--------|---|
| Filter | Filter | Filter | Filter | Filter               | Filter                                     | Filter | Filter | Filter                                    |
| 1      | 1      | 1      | NULL   | println.java         | D:\Documents\University\Semesters\Term ... | NULL   | NULL   | package com.calculator.app.display;...    |
| 2      | 2      | 1      | NULL   | print_fail.java      | D:\Documents\University\Semesters\Term ... | NULL   | NULL   | package com.calculator.app.display;...    |
| 3      | 3      | 1      | NULL   | print_success.java   | D:\Documents\University\Semesters\Term ... | NULL   | NULL   | package com.calculator.app.display;...    |
| 4      | 4      | 1      | NULL   | Main.java            | D:\Documents\University\Semesters\Term ... | NULL   | NULL   | package com.calculator.app.init;...       |
| 5      | 5      | 1      | NULL   | basic_operation.j... | D:\Documents\University\Semesters\Term ... | NULL   | NULL   | package com.calculator.app.method;...     |
| 6      | 6      | 1      | NULL   | fibonacci.java       | D:\Documents\University\Semesters\Term ... | NULL   | NULL   | package com.calculator.app.method;...     |
| 7      | 7      | 1      | NULL   | integral.java        | D:\Documents\University\Semesters\Term ... | NULL   | NULL   | package com.calculator.app.method;...     |
| 8      | 8      | 1      | NULL   | printLog.java        | D:\Documents\University\Semesters\Term ... | NULL   | NULL   | package com.calculator.app.method;...     |
| 9      | 9      | 98     | 3      | print_success        | com.calculator.app.display.print_success   | NULL   | NULL   | classprint_success(publicstaticvoidpr...  |
| 10     | 10     | 98     | 2      | print_fail           | com.calculator.app.display.print_fail      | NULL   | NULL   | classprint_fail(publicvoidprint_fail_m... |
| 11     | 11     | 84     | NULL   | System               | java.lang.System                           | NULL   | NULL   |   |
| 12     | 12     | 98     | 7      | integral             | com.calculator.app.method.integral         | NULL   | NULL   | classintegralextendsbasic_operation(...   |
| 13     | 13     | 98     | 6      | fibonacci            | com.calculator.app.method.fibonacci        | NULL   | NULL   | classfibonacixtendsbasic_operation...     |
| 14     | 14     | 98     | 1      | println              | com.calculator.app.display.println         | NULL   | NULL   | classprintln(publicvoidprint(Stringtex... |
| 15     | 15     | 98     | 4      | Main                 | com.calculator.app.init.Main               | NULL   | NULL   | classMain(publicstaticvoidmain(Strin...   |
| 16     | 16     | 98     | 8      | printLog             | com.calculator.app.method.printLog         | NULL   | NULL   | classprintLogextendsprintln(@Overri...    |

Go to: 1

1 - 16 of 16

UTF-8

Edit Database Cell

Mode: Text

```
1 package com.calculator.app.display;
2
3 import com.calculator.app.display.print_success;
4 import com.calculator.app.display.print_fail;
5
6 import java.lang.System;
```

Type of data currently in cell: Text / Numeric  
819 character(s)

Apply

Remote

Identity Select an identity to connect

DBHub.io Local Current Database

Name Last modified Size

DB Browser for SQLite - D:\Documents\University\Semesters\Term B\Compiler\Project P1\database\import\calculator\_app.oudb

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project Save Project Attach Database Close Database

Database Structure Browse Data Edit Pragnas Execute SQL

Table: referencemodel

|        | _id    | kind_id | _file_id | _line  | _column | _ent_id | _scope_id |
|--------|--------|---------|----------|--------|---------|---------|-----------|
| Filter | Filter | Filter  | Filter   | Filter | Filter  | Filter  | Filter    |
| 1      | 1      | 206     | 1        | 3      | 0       | 9       | 1         |
| 2      | 2      | 207     | 1        | 3      | 0       | 1       | 9         |
| 3      | 3      | 206     | 1        | 4      | 0       | 10      | 1         |
| 4      | 4      | 207     | 1        | 4      | 0       | 1       | 10        |
| 5      | 5      | 206     | 1        | 6      | 0       | 11      | 1         |
| 6      | 6      | 207     | 1        | 6      | 0       | 1       | 11        |
| 7      | 7      | 206     | 2        | 3      | 0       | 11      | 2         |
| 8      | 8      | 207     | 2        | 3      | 0       | 2       | 11        |
| 9      | 9      | 206     | 3        | 3      | 0       | 11      | 3         |
| 10     | 10     | 207     | 3        | 3      | 0       | 3       | 11        |
| 11     | 11     | 206     | 4        | 3      | 0       | 12      | 4         |
| 12     | 12     | 207     | 4        | 3      | 0       | 4       | 12        |
| 13     | 13     | 206     | 4        | 4      | 0       | 13      | 4         |
| 14     | 14     | 207     | 4        | 4      | 0       | 4       | 13        |
| 15     | 15     | 206     | 4        | 5      | 0       | 9       | 4         |
| 16     | 16     | 207     | 4        | 5      | 0       | 4       | 9         |
| 17     | 17     | 206     | 4        | 6      | 0       | 14      | 4         |
| 18     | 18     | 207     | 4        | 6      | 0       | 4       | 14        |
| 19     | 19     | 206     | 5        | 3      | 0       | 15      | 5         |
| 20     | 20     | 207     | 5        | 3      | 0       | 5       | 15        |
| 21     | 21     | 206     | 7        | 3      | 0       | 16      | 7         |
| 22     | 22     | 207     | 7        | 3      | 0       | 7       | 16        |
| 23     | 23     | 206     | 8        | 3      | 0       | 14      | 8         |

Go to: 1

1 - 24 of 24

UTF-8

Edit Database Cell

Mode: Text

```
1
```

Type of data currently in cell: Text / Numeric  
1 character(s)

Apply

Remote

Identity Select an identity to connect

DBHub.io Local Current Database

Name Last modified Size



```
D:\Documents\University\Semesters\Term 8\Compiler\Project P1\OpenUnderstand\openunderstand>python g10_tests.py
1. ref name: Import
2. ref scope: C:\Users\alise\Desktop\calculator_app\src\com\calculator\app\method\integral.java || kind: File
3. ref ent: com.calculator.app.method.printLog || kind: Public Class
4. file location: C:\Users\alise\Desktop\calculator_app\src\com\calculator\app\method\integral.java || line: 3
-----
1. ref name: Import
2. ref scope: C:\Users\alise\Desktop\calculator_app\src\com\calculator\app\display\println.java || kind: File
3. ref ent: com.calculator.app.display.print_success || kind: Public Class
4. file location: C:\Users\alise\Desktop\calculator_app\src\com\calculator\app\display\println.java || line: 3
-----
1. ref name: Import
2. ref scope: C:\Users\alise\Desktop\calculator_app\src\com\calculator\app\display\println.java || kind: File
3. ref ent: com.calculator.app.display.print_fail || kind: Public Class
4. file location: C:\Users\alise\Desktop\calculator_app\src\com\calculator\app\display\println.java || line: 4
-----
1. ref name: Import
2. ref scope: C:\Users\alise\Desktop\calculator_app\src\com\calculator\app\display\println.java || kind: File
3. ref ent: java.lang.System || kind: Unknown Class
4. file location: C:\Users\alise\Desktop\calculator_app\src\com\calculator\app\display\println.java || line: 6
-----
1. ref name: Import
2. ref scope: C:\Users\alise\Desktop\calculator_app\src\com\calculator\app\init\Main.java || kind: File
3. ref ent: com.calculator.app.method.integral || kind: Public Class
4. file location: C:\Users\alise\Desktop\calculator_app\src\com\calculator\app\init\Main.java || line: 3
-----
1. ref name: Import
2. ref scope: C:\Users\alise\Desktop\calculator_app\src\com\calculator\app\init\Main.java || kind: File
3. ref ent: com.calculator.app.method.fibonacci || kind: Public Class
4. file location: C:\Users\alise\Desktop\calculator_app\src\com\calculator\app\init\Main.java || line: 4
-----
1. ref name: Import
2. ref scope: C:\Users\alise\Desktop\calculator_app\src\com\calculator\app\init\Main.java || kind: File
3. ref ent: com.calculator.app.display.print_success || kind: Public Class
4. file location: C:\Users\alise\Desktop\calculator_app\src\com\calculator\app\init\Main.java || line: 5
-----
1. ref name: Import
2. ref scope: C:\Users\alise\Desktop\calculator_app\src\com\calculator\app\init\Main.java || kind: File
3. ref ent: com.calculator.app.display.println || kind: Public Class
4. file location: C:\Users\alise\Desktop\calculator_app\src\com\calculator\app\init\Main.java || line: 6
-----
1. ref name: Import
2. ref scope: C:\Users\alise\Desktop\calculator_app\src\com\calculator\app\display\print_success.java || kind: File
3. ref ent: java.lang.System || kind: Unknown Class
4. file location: C:\Users\alise\Desktop\calculator_app\src\com\calculator\app\display\print_success.java || line: 3
```

## Import(xerces2j example)

```

1. ref name: Java Import
2. ref scope: D:\Documents\University\Semesters\Term 8\Compiler\Project P1\benchmarks\xerces2j\src\org\apache\html\dom\HTMLDocumentImpl.java || kind: Java File
3. ref ent: org.w3c.dom.html.HTMLTitleElement || kind: Java Unknown Class Type Member
4. file location: HTMLDocumentImpl.java || line: 40
-----
1. ref name: Java Import
2. ref scope: D:\Documents\University\Semesters\Term 8\Compiler\Project P1\benchmarks\xerces2j\src\org\apache\html\dom\HTMLDOMImplementationImpl.java || kind: Java File
3. ref ent: org.apache.xerces.dom.DOMImplementationImpl || kind: Java Class Type Public Member
4. file location: HTMLDOMImplementationImpl.java || line: 20
-----
1. ref name: Java Import
2. ref scope: D:\Documents\University\Semesters\Term 8\Compiler\Project P1\benchmarks\xerces2j\src\org\apache\html\dom\HTMLDOMImplementationImpl.java || kind: Java File
3. ref ent: org.w3c.dom.DOMException || kind: Java Unknown Class Type Member
4. file location: HTMLDOMImplementationImpl.java || line: 21
-----
1. ref name: Java Import
2. ref scope: D:\Documents\University\Semesters\Term 8\Compiler\Project P1\benchmarks\xerces2j\src\org\apache\html\dom\HTMLDOMImplementationImpl.java || kind: Java File
3. ref ent: org.w3c.dom.html.HTMLDOMImplementation || kind: Java Interface Type Public
4. file location: HTMLDOMImplementationImpl.java || line: 22
-----
1. ref name: Java Import
2. ref scope: D:\Documents\University\Semesters\Term 8\Compiler\Project P1\benchmarks\xerces2j\src\org\apache\html\dom\HTMLDOMImplementationImpl.java || kind: Java File
3. ref ent: org.w3c.dom.html.HTMLDocument || kind: Java Unknown Class Type Member
4. file location: HTMLDOMImplementationImpl.java || line: 23
-----

```

# Compiler Project

DB Browser for SQLite - D:\Documents\University\Semesters\Term 8\Compiler\Project P1\database\import\excer2j.oudb

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project Save Project Attach Database Close Database

Database Structure Browse Data Edit Pragma Execute SQL

Table: entitymodel

|        | _id    | _kind_id | _parent | _name                          | _longname                                  | _value | _type  | _contents |
|--------|--------|----------|---------|--------------------------------|--|--------|--------|-----------|
| Filter | Filter | Filter   | Filter  | Filter                         | Filter                                     | Filter | Filter | Filter    |
| 1      | 1      | 1        | NULL    | HTMLAnchorElementImpl.java     | D:\Documents\University\Semesters\Term ... | NULL   | NULL   | /s...     |
| 2      | 2      | 1        | NULL    | HTMLAppletElementImpl.java     | D:\Documents\University\Semesters\Term ... | NULL   | NULL   | /s...     |
| 3      | 3      | 1        | NULL    | HTMLAreaElementImpl.java       | D:\Documents\University\Semesters\Term ... | NULL   | NULL   | /s...     |
| 4      | 4      | 1        | NULL    | HTMLBaseElementImpl.java       | D:\Documents\University\Semesters\Term ... | NULL   | NULL   | /s...     |
| 5      | 5      | 1        | NULL    | HTMLBaseFontElementImpl.java   | D:\Documents\University\Semesters\Term ... | NULL   | NULL   | /s...     |
| 6      | 6      | 1        | NULL    | HTMLBodyElementImpl.java       | D:\Documents\University\Semesters\Term ... | NULL   | NULL   | /s...     |
| 7      | 7      | 1        | NULL    | HTMLBRElementImpl.java         | D:\Documents\University\Semesters\Term ... | NULL   | NULL   | /s...     |
| 8      | 8      | 1        | NULL    | HTMLBuilder.java               | D:\Documents\University\Semesters\Term ... | NULL   | NULL   | /s...     |
| 9      | 9      | 1        | NULL    | HTMLButtonElementImpl.java     | D:\Documents\University\Semesters\Term ... | NULL   | NULL   | /s...     |
| 10     | 10     | 1        | NULL    | HTMLCollectionImpl.java        | D:\Documents\University\Semesters\Term ... | NULL   | NULL   | /s...     |
| 11     | 11     | 1        | NULL    | HTMLDirectoryElementImpl.java  | D:\Documents\University\Semesters\Term ... | NULL   | NULL   | /s...     |
| 12     | 12     | 1        | NULL    | HTMLDivElementImpl.java        | D:\Documents\University\Semesters\Term ... | NULL   | NULL   | /s...     |
| 13     | 13     | 1        | NULL    | HTMLDListElementImpl.java      | D:\Documents\University\Semesters\Term ... | NULL   | NULL   | /s...     |
| 14     | 14     | 1        | NULL    | HTMLDocumentImpl.java          | D:\Documents\University\Semesters\Term ... | NULL   | NULL   | /s...     |
| 15     | 15     | 1        | NULL    | HTMLDOMImplementationImpl.java | D:\Documents\University\Semesters\Term ... | NULL   | NULL   | /s...     |
| 16     | 16     | 1        | NULL    | HTMLElementImpl.java           | D:\Documents\University\Semesters\Term ... | NULL   | NULL   | /s...     |
| 17     | 17     | 1        | NULL    | HTMLFieldSetElementImpl.java   | D:\Documents\University\Semesters\Term ... | NULL   | NULL   | /s...     |
| 18     | 18     | 1        | NULL    | HTMLFontElementImpl.java       | D:\Documents\University\Semesters\Term ... | NULL   | NULL   | /s...     |
| 19     | 19     | 1        | NULL    | HTMLFormControl.java           | D:\Documents\University\Semesters\Term ... | NULL   | NULL   | /s...     |
| 20     | 20     | 1        | NULL    | HTMLFormElementImpl.java       | D:\Documents\University\Semesters\Term ... | NULL   | NULL   | /s...     |
| 21     | 21     | 1        | NULL    | HTMLFrameElementImpl.java      | D:\Documents\University\Semesters\Term ... | NULL   | NULL   | /s...     |
| 22     | 22     | 1        | NULL    | HTMLFrameSetElementImpl.java   | D:\Documents\University\Semesters\Term ... | NULL   | NULL   | /s...     |
| 23     | 23     | 1        | NULL    | HTMLHeadElementImpl.java       | D:\Documents\University\Semesters\Term ... | NULL   | NULL   | /s...     |

Go to: 1

1 - 24 of 1331

UTF-8

Mode: Text

1

2

3

4

Type of data currently in cell: Text / Numeric  
4708 character(s)

Apply

Remote

Identity: Select an identity to connect

DBHub.io Local Current Database

Name Last modified Size

SQL Log Plot DB Schema Remote

UTF-8

DB Browser for SQLite - D:\Documents\University\Semesters\Term 8\Compiler\Project P1\database\import\excer2j.oudb

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project Save Project Attach Database Close Database

Database Structure Browse Data Edit Pragma Execute SQL

Table: referencemodel

|        | _id    | _kind_id | _file_id | _line  | _column | _ent_id | _scope_id |
|--------|--------|----------|----------|--------|---------|---------|-----------|
| Filter | Filter | Filter   | Filter   | Filter | Filter  | Filter  | Filter    |
| 1      | 1      | 206      | 1        | 19     | 0       | 712     | 1         |
| 2      | 2      | 207      | 1        | 19     | 0       | 1       | 712       |
| 3      | 3      | 206      | 2        | 19     | 0       | 713     | 2         |
| 4      | 4      | 207      | 2        | 19     | 0       | 2       | 713       |
| 5      | 5      | 206      | 3        | 19     | 0       | 714     | 3         |
| 6      | 6      | 207      | 3        | 19     | 0       | 3       | 714       |
| 7      | 7      | 206      | 4        | 19     | 0       | 715     | 4         |
| 8      | 8      | 207      | 4        | 19     | 0       | 4       | 715       |
| 9      | 9      | 206      | 5        | 19     | 0       | 716     | 5         |
| 10     | 10     | 207      | 5        | 19     | 0       | 5       | 716       |
| 11     | 11     | 206      | 6        | 19     | 0       | 717     | 6         |
| 12     | 12     | 207      | 6        | 19     | 0       | 6       | 717       |
| 13     | 13     | 206      | 7        | 19     | 0       | 718     | 7         |
| 14     | 14     | 207      | 7        | 19     | 0       | 7       | 718       |
| 15     | 15     | 206      | 8        | 20     | 0       | 719     | 8         |
| 16     | 16     | 207      | 8        | 20     | 0       | 8       | 719       |
| 17     | 17     | 206      | 8        | 22     | 0       | 720     | 8         |
| 18     | 18     | 207      | 8        | 22     | 0       | 8       | 720       |
| 19     | 19     | 206      | 8        | 23     | 0       | 721     | 8         |
| 20     | 20     | 207      | 8        | 23     | 0       | 8       | 721       |
| 21     | 21     | 206      | 8        | 24     | 0       | 722     | 8         |
| 22     | 22     | 207      | 8        | 24     | 0       | 8       | 722       |
| 23     | 23     | 206      | 8        | 25     | 0       | 723     | 8         |

Go to: 1

1 - 24 of 9450

UTF-8

Mode: Text

1

Type of data currently in cell: Text / Numeric  
1 character(s)

Apply

Remote

Identity: Select an identity to connect

DBHub.io Local Current Database

Name Last modified Size

SQL Log Plot DB Schema Remote

UTF-8

```

1. ref name: Import
2. ref scope: C:\Users\alise\Desktop\benchmarks\xerces2j\src\org\apache\xerces\dom\SecuritySupport.java || kind: File
3. ref ent: java.security.PrivilegedAction || kind: Unknown Class
4. file location: C:\Users\alise\Desktop\benchmarks\xerces2j\src\org\apache\xerces\dom\SecuritySupport.java || line: 25
-----
1. ref name: Import
2. ref scope: C:\Users\alise\Desktop\benchmarks\xerces2j\src\org\apache\xerces\dom\SecuritySupport.java || kind: File
3. ref ent: java.security.PrivilegedActionException || kind: Unknown Class
4. file location: C:\Users\alise\Desktop\benchmarks\xerces2j\src\org\apache\xerces\dom\SecuritySupport.java || line: 26
-----
1. ref name: Import
2. ref scope: C:\Users\alise\Desktop\benchmarks\xerces2j\src\org\apache\xerces\dom\SecuritySupport.java || kind: File
3. ref ent: java.security.PrivilegedExceptionAction || kind: Unknown Class
4. file location: C:\Users\alise\Desktop\benchmarks\xerces2j\src\org\apache\xerces\dom\SecuritySupport.java || line: 27
-----
1. ref name: Import
2. ref scope: C:\Users\alise\Desktop\benchmarks\xerces2j\src\org\apache\xerces\dom\DocumentImpl.java || kind: File
3. ref ent: org.apache.xerces.dom.events.EventImpl || kind: Public Class
4. file location: C:\Users\alise\Desktop\benchmarks\xerces2j\src\org\apache\xerces\dom\DocumentImpl.java || line: 31
-----
1. ref name: Import
2. ref scope: C:\Users\alise\Desktop\benchmarks\xerces2j\src\org\apache\xerces\dom\DocumentImpl.java || kind: File
3. ref ent: org.apache.xerces.dom.events.MouseEventImpl || kind: Public Class
4. file location: C:\Users\alise\Desktop\benchmarks\xerces2j\src\org\apache\xerces\dom\DocumentImpl.java || line: 32
-----
1. ref name: Import
2. ref scope: C:\Users\alise\Desktop\benchmarks\xerces2j\src\org\apache\xerces\dom\DocumentImpl.java || kind: File
3. ref ent: org.apache.xerces.dom.events.MutationEventImpl || kind: Public Class
4. file location: C:\Users\alise\Desktop\benchmarks\xerces2j\src\org\apache\xerces\dom\DocumentImpl.java || line: 33
-----
1. ref name: Import
2. ref scope: C:\Users\alise\Desktop\benchmarks\xerces2j\src\org\apache\xerces\dom\DocumentImpl.java || kind: File
3. ref ent: org.apache.xerces.dom.events.UIEventImpl || kind: Public Class
4. file location: C:\Users\alise\Desktop\benchmarks\xerces2j\src\org\apache\xerces\dom\DocumentImpl.java || line: 34

```

**Modify deref(jhotdraw-develop example)**

```

ctype['-']=CT_DIGIT
1. ref name: Java Modify Deref Partial
2. ref scope: StreamPosTokenizer.parseNumbers || kind: Java Method Default Member
3. ref ent: ctype || kind: Java Variable
4. file location: StreamPosTokenizer.java || line: 349
-----
ctype['+']=CT_DIGIT
1. ref name: Java Modify Deref Partial
2. ref scope: StreamPosTokenizer.parsePlusAsNumber || kind: Java Method Default Member
3. ref ent: ctype || kind: Java Variable
4. file location: StreamPosTokenizer.java || line: 354
-----
buf[i++]=(char)c
1. ref name: Java Modify Deref Partial
2. ref scope: StreamPosTokenizer.nextToken || kind: Java Method Default Member
3. ref ent: buf || kind: Java Variable
4. file location: StreamPosTokenizer.java || line: 727
-----
buf[i++]=(char)c
1. ref name: Java Modify Deref Partial
2. ref scope: StreamPosTokenizer.nextToken || kind: Java Method Default Member
3. ref ent: buf || kind: Java Variable
4. file location: StreamPosTokenizer.java || line: 803

```

# Compiler Project

DB Browser for SQLite - D:\Documents\University\Semesters\Term 8\Compiler\Project P1\database\modify\jhotdraw-develop.odb

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project Save Project Attach Database Close Database

Database Structure Browse Data Edit Pragma Execute SQL

Table: entitymodel

| _id    | _kind_id | _parent_id | _name                        | _longname                         | _value | _type  | _contents                               |
|--------|----------|------------|------------------------------|-----------------------------------|--------|--------|---|
| Filter | Filter   | Filter     | Filter                       | Filter                            | Filter | Filter | Filter                                  |
| 888    | 888      | 33         | 883 drawParagraph            | TextAreaFigure.drawParagraph      | NULL   | NULL   | Rectangle2D.Double.drawParagraph(...    |
| 889    | 889      | 33         | 883 getPreferredSize         | TextAreaFigure.getPreferredSize   | NULL   | NULL   | Dimension2D.Double.getPreferredSize(... |
| 890    | 890      | 98         | 206 TextFigure               | D:...                             | NULL   | NULL   | classTextFigureextendsAbstractAttri...  |
| 891    | 891      | 98         | 208 TriangleFigure           | D:...                             | NULL   | NULL   | classTriangleFigureextendsAbstractA...  |
| 892    | 892      | 79         | 209 AbstractConnectionHandle | D:...                             | NULL   | NULL   | classAbstractConnectionHandleexten...   |
| 893    | 893      | 79         | 210 AbstractHandle           | D:...                             | NULL   | NULL   | classAbstractHandleimplementsHand...    |
| 894    | 894      | 79         | 211 AbstractRotateHandle     | D:...                             | NULL   | NULL   | classAbstractRotateHandleextendsA...    |
| 895    | 895      | 98         | 212 BezierControlPointHandle | D:...                             | NULL   | NULL   | classBezierControlPointHandleextend...  |
| 896    | 896      | 152        | NULL newvalue.x              | newvalue.x                        | NULL   | NULL   | NULL                                    |
| 897    | 897      | 33         | 895 trackEnd                 | BezierControlPointHandle.trackEnd | NULL   | NULL   | voidtrackEnd(Pointanchor,Pointlead,...  |
| 898    | 898      | 152        | NULL newvalue.y              | newvalue.y                        | NULL   | NULL   | NULL                                    |
| 899    | 899      | 98         | 213 BezierNodeHandle         | D:...                             | NULL   | NULL   | classBezierNodeHandleextendsAbstr...    |
| 900    | 900      | 98         | 214 BezierOutlineHandle      | D:...                             | NULL   | NULL   | classBezierOutlineHandleextendsAbs...   |
| 901    | 901      | 98         | 215 BezierScaleHandle        | D:...                             | NULL   | NULL   | classBezierScaleHandleextendsAbstr...   |
| 902    | 902      | 98         | 216 BoundsOutlineHandle      | D:...                             | NULL   | NULL   | classBoundsOutlineHandleextendsAb...    |
| 903    | 903      | 98         | 217 CloseHandle              | D:...                             | NULL   | NULL   | classCloseHandleextendsLocatorHan...    |
| 904    | 904      | 98         | 218 ConnectionEndHandle      | D:...                             | NULL   | NULL   | classConnectionEndHandleextendsAb...    |
| 905    | 905      | 98         | 219 ConnectionStartHandle    | D:...                             | NULL   | NULL   | classConnectionStartHandleextendsA...   |
| 906    | 906      | 98         | 220 ConnectorHandle          | D:...                             | NULL   | NULL   | classConnectorHandleextendsAbstrac...   |
| 907    | 907      | 98         | 221 DragHandle               | D:...                             | NULL   | NULL   | classDragHandleextendsAbstractHan...    |
| 908    | 908      | 98         | 222 FontSizeHandle           | D:...                             | NULL   | NULL   | classFontSizeHandleextendsLocatorH...   |
| 909    | 909      | 98         | 224 HandleAttributeKeys      | D:...                             | NULL   | NULL   | classHandleAttributeKeys{privatestat... |
| 910    | 910      | 79         | 225 LocatorHandle            | D:...                             | NULL   | NULL   | classLocatorHandleextendsAbstractH...   |

Go to: 1

UTF-8

Edit Database Cell

Mode: Text

1 CloseHandle

Type of data currently in cell: Text / Numeric  
11 character(s)

Remote

Identity Select an identity to connect

DBHub.io Local Current Database

Name Last modified Size

SQL Log Plot DB Schema Remote

DB Browser for SQLite - D:\Documents\University\Semesters\Term 8\Compiler\Project P1\database\modify\jhotdraw-develop.odb

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project Save Project Attach Database Close Database

Database Structure Browse Data Edit Pragma Execute SQL

Table: referencemodel

| _id    | _kind_id | _file_id | _line  | _column | _ent_id | _scope_id |
|--------|----------|----------|--------|---------|---------|-----------|
| Filter | Filter   | Filter   | Filter | Filter  | Filter  | Filter    |
| 1      | 1        | 210      | 71     | 227     | 16      | 732       |
| 2      | 2        | 211      | 71     | 227     | 16      | 733       |
| 3      | 3        | 210      | 71     | 234     | 20      | 732       |
| 4      | 4        | 211      | 71     | 234     | 20      | 734       |
| 5      | 5        | 210      | 71     | 237     | 20      | 732       |
| 6      | 6        | 211      | 71     | 237     | 20      | 734       |
| 7      | 7        | 210      | 71     | 242     | 8       | 732       |
| 8      | 8        | 211      | 71     | 242     | 8       | 735       |
| 9      | 9        | 210      | 83     | 953     | 12      | 747       |
| 10     | 10       | 211      | 83     | 953     | 12      | 748       |
| 11     | 11       | 210      | 85     | 438     | 16      | 751       |
| 12     | 12       | 211      | 85     | 438     | 16      | 752       |
| 13     | 13       | 210      | 89     | 1302    | 12      | 747       |
| 14     | 14       | 211      | 89     | 1302    | 12      | 758       |
| 15     | 15       | 210      | 187    | 432     | 12      | 865       |
| 16     | 16       | 211      | 187    | 432     | 12      | 866       |
| 17     | 17       | 210      | 187    | 433     | 12      | 867       |
| 18     | 18       | 211      | 187    | 433     | 12      | 868       |
| 19     | 19       | 210      | 205    | 103     | 20      | 884       |
| 20     | 20       | 211      | 205    | 103     | 20      | 885       |
| 21     | 21       | 210      | 205    | 111     | 28      | 886       |
| 22     | 22       | 211      | 205    | 111     | 28      | 887       |
| 23     | 23       | 210      | 205    | 158     | 16      | 887       |

Go to: 1

UTF-8

Edit Database Cell

Mode: Text

1

Type of data currently in cell: Text / Numeric  
1 character(s)

Remote

Identity Select an identity to connect

DBHub.io Local Current Database

Name Last modified Size

SQL Log Plot DB Schema Remote



**Modify deref(JSON example)**

```

names[i]=fields[i].getName()
1. ref name: Java Modify Deref Partial
2. ref scope: JSONObject.getNames || kind: Java Method Default Member
3. ref ent: names || kind: Java Variable
4. file location: JSONObject.java || line: 844
-----
chars[pos]=this.next()
1. ref name: Java Modify Deref Partial
2. ref scope: JSONTokenizer.next || kind: Java Method Default Member
3. ref ent: chars || kind: Java Variable
4. file location: JSONTokenizer.java || line: 275
-----
this.stack[this.top]=jo
1. ref name: Java Modify Deref Partial
2. ref scope: JSONWriter.push || kind: Java Method Default Member
3. ref ent: this.stack || kind: Java Variable
4. file location: JSONWriter.java || line: 295
-----
circle[i]=c
1. ref name: Java Modify Deref Partial
2. ref scope: XMLTokenizer.skipPast || kind: Java Method Default Member
3. ref ent: circle || kind: Java Variable
4. file location: XMLTokenizer.java || line: 371
-----
circle[offset]=c
1. ref name: Java Modify Deref Partial
2. ref scope: XMLTokenizer.skipPast || kind: Java Method Default Member
3. ref ent: circle || kind: Java Variable
4. file location: XMLTokenizer.java || line: 409

```

# Compiler Project

DB Browser for SQLite - D:\Documents\University\Semesters\Term 8\Compiler\Project P1\database\modify\JSON.oudb

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project Save Project Attach Database Close Database

Database Structure Browse Data Edit Pragma Execute SQL

Table: entitymodel

| _id    | _kind_id | _parent_id | _name  | _longname                   | _value                                     | _type  | _contents                                 |
|--------|----------|------------|--------|-----------------------------|--|--------|---|
| Filter | Filter   | Filter     | Filter | Filter                      | Filter                                     | Filter | Filter                                    |
| 22     | 22       | 1          | NULL   | XMLSchemaTypeConverter.java | D:\Documents\University\Semesters\Term ... | NULL   | package org.json;...                      |
| 23     | 23       | 98         | 1      | CDL                         | D:\Documents\University\Semesters\Term ... | NULL   | classCDL{privateStringgetValue(JSO...     |
| 24     | 24       | 98         | 2      | Cookie                      | D:\Documents\University\Semesters\Term ... | NULL   | classCookie{publicstaticStringescape...   |
| 25     | 25       | 98         | 3      | CookieList                  | D:\Documents\University\Semesters\Term ... | NULL   | classCookieList{publicstaticJSONObje...   |
| 26     | 26       | 98         | 4      | HTTP                        | D:\Documents\University\Semesters\Term ... | NULL   | classHTTP{publicstaticfinalStringCRL...   |
| 27     | 27       | 98         | 5      | HTTPTokener                 | D:\Documents\University\Semesters\Term ... | NULL   | classHTTPTokenerextendsJSONToke...        |
| 28     | 28       | 98         | 6      | JSONArray                   | D:\Documents\University\Semesters\Term ... | NULL   | classJSONArrayimplementsIterable<...      |
| 29     | 29       | 98         | 7      | JSONException               | D:\Documents\University\Semesters\Term ... | NULL   | classJSONExceptionextendsRuntimeE...      |
| 30     | 30       | 98         | 8      | JSONML                      | D:\Documents\University\Semesters\Term ... | NULL   | classJSONML{privatestaticObjectpars...    |
| 31     | 31       | 98         | 9      | JSONObject                  | D:\Documents\University\Semesters\Term ... | NULL   | classJSONObject{privatestaticfinalcla...  |
| 32     | 32       | 95         | 9      | Null                        | D:\Documents\University\Semesters\Term ... | NULL   | classNull{@OverrideprotectedfinalOb...    |
| 33     | 33       | 152        | NULL   | names                       | names                                      | NULL   | NULL                                      |
| 34     | 34       | 33         | 31     | getNames                    | JSONObject.getNames                        | NULL   | String[]getNames(Objectobject)...         |
| 35     | 35       | 98         | 10     | JSONPointer                 | D:\Documents\University\Semesters\Term ... | NULL   | classJSONPointer{privatestaticfinalSt...  |
| 36     | 36       | 95         | 10     | Builder                     | D:\Documents\University\Semesters\Term ... | NULL   | classBuilder{privatefinalList<String>r... |
| 37     | 37       | 98         | 11     | JSONPointerException        | D:\Documents\University\Semesters\Term ... | NULL   | classJSONPointerExceptionextendsJS...     |
| 38     | 38       | 98         | 15     | JSONStringer                | D:\Documents\University\Semesters\Term ... | NULL   | classJSONStringerextendsJSONWriter...     |
| 39     | 39       | 98         | 16     | JSONTokener                 | D:\Documents\University\Semesters\Term ... | NULL   | classJSONTokener{privatefinalcharac...    |
| 40     | 40       | 152        | NULL   | chars                       | chars                                      | NULL   | NULL                                      |
| 41     | 41       | 33         | 39     | next                        | JSONTokener.next                           | NULL   | Stringnext(intn)throwsJSONExceptio...     |
| 42     | 42       | 98         | 17     | JSONWriter                  | D:\Documents\University\Semesters\Term ... | NULL   | classJSONWriter{privatestaticfinalint...  |
| 43     | 43       | 152        | NULL   | this_stack                  | this_stack                                 | NULL   | NULL                                      |
| 44     | 44       | 33         | 42     | push                        | JSONWriter.push                            | NULL   | voidpush(JSONObjectjo)throwsJSON...       |

Go to: 1

22 - 45 of 50

UTF-8

Edit Database Cell

Mode: Text

1 D:\Documents\University\Semesters\Term 8\Compiler\Project P1\benchmarks\JSON\src\main\java\org\json\JSONPointerException.java

Type of data currently in cell: Text / Numeric  
125 character(s)

Apply

Remote

Identity Select an identity to connect

DBHub.io Local Current Database

Name Last modified Size

SQL Log Plot DB Schema Remote

UTF-8

DB Browser for SQLite - D:\Documents\University\Semesters\Term 8\Compiler\Project P1\database\modify\JSON.oudb

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project Save Project Attach Database Close Database

Database Structure Browse Data Edit Pragma Execute SQL

Table: referencemodel

| _id    | _kind_id | _file_id | _line  | _column | _ent_id | _scope_id |
|--------|----------|----------|--------|---------|---------|-----------|
| Filter | Filter   | Filter   | Filter | Filter  | Filter  | Filter    |
| 1      | 1        | 210      | 9      | 844     | 12      | 33        |
| 2      | 2        | 211      | 9      | 844     | 12      | 34        |
| 3      | 3        | 210      | 16     | 275     | 12      | 40        |
| 4      | 4        | 211      | 16     | 275     | 12      | 41        |
| 5      | 5        | 210      | 17     | 295     | 8       | 43        |
| 6      | 6        | 211      | 17     | 295     | 8       | 44        |
| 7      | 7        | 210      | 21     | 371     | 12      | 49        |
| 8      | 8        | 211      | 21     | 371     | 12      | 50        |
| 9      | 9        | 210      | 21     | 409     | 12      | 49        |
| 10     | 10       | 211      | 21     | 409     | 12      | 50        |

Go to: 1

1 - 10 of 10

UTF-8

Edit Database Cell

Mode: Text

1

Type of data currently in cell: Text / Numeric  
1 character(s)

Apply

Remote

Identity Select an identity to connect

DBHub.io Local Current Database

Name Last modified Size

SQL Log Plot DB Schema Remote

UTF-8



## Cyclomatic:

- calculator\_app

calculator\_app

|            | # Cyclomatic | # Entity Set | # Methods |
|------------|--------------|--------------|-----------|
| Understand | 20           | 3            | 17        |
| Our Work   | 20           | 3            | 17        |

```
{'val': 1, 'name': 'calculate_rectangle', 'kind': 'Java Method Public Member'}
{'val': 1, 'name': 'main', 'kind': 'Java Static Method Public Main Member'}
{'val': 1, 'name': 'multiplication', 'kind': 'Java Static Method Public Member'}
{'val': 1, 'name': 'multiplication', 'kind': 'Java Static Method Public Member'}
{'val': 1, 'name': 'pow', 'kind': 'Java Static Method Public Member'}
{'val': 1, 'name': 'print', 'kind': 'Java Method Public Member'}
{'val': 1, 'name': 'print', 'kind': 'Java Method Public Member'}
{'val': 1, 'name': 'print', 'kind': 'Java Method Public Member'}
{'val': 1, 'name': 'print_fail', 'kind': 'Java Method Public Member'}
{'val': 1, 'name': 'print_fail_message', 'kind': 'Java Method Public Member'}
{'val': 1, 'name': 'print_success', 'kind': 'Java Method Public Member'}
{'val': 1, 'name': 'print_success_message', 'kind': 'Java Static Method Public Membe'}
{'val': 1, 'name': 'sum', 'kind': 'Java Static Method Public Member'}
{'val': 1, 'name': 'sum', 'kind': 'Java Static Method Public Member'}
{'val': 2, 'name': 'calculate_pow_x_n', 'kind': 'Java Method Public Member'}
{'val': 2, 'name': 'calculate_to_n', 'kind': 'Java Static Method Public Member'}
{'val': 2, 'name': 'calculate_x', 'kind': 'Java Method Public Member'}
```

```
-----
Entities with cyclomatic: 3
Java Method Public Member
Java Static Method Public Main Member
Java Static Method Public Member
-----
```

```
Project Cyclomatic Summary
Project total cyclomatic: 20
```

## • JSON

JSON

|            | # Cyclomatic | # Entity Set | # Methods |
|------------|--------------|--------------|-----------|
| Understand | 1088         | 14           | 327       |
| Our Work   | 1088         | 14           | 327       |

```
{'val': 11, 'name': 'toString', 'kind': 'Java Static Method Public Member'}
{'val': 11, 'name': 'toString', 'kind': 'Java Static Method Public Member'}
{'val': 12, 'name': 'similar', 'kind': 'Java Method Public Member'}
{'val': 12, 'name': 'valueToString', 'kind': 'Java Static Method Public Member'}
{'val': 12, 'name': 'write', 'kind': 'Java Method Public Member'}
{'val': 13, 'name': 'JSONObject', 'kind': 'Java Method Constructor Member Public'}
{'val': 13, 'name': 'quote', 'kind': 'Java Static Method Public Member'}
{'val': 13, 'name': 'similar', 'kind': 'Java Method Public Member'}
{'val': 13, 'name': 'stringToNumber', 'kind': 'Java Static Method Protected Member'}
{'val': 13, 'name': 'stringToNumber', 'kind': 'Java Static Method Private Member'}
{'val': 14, 'name': 'writeValue', 'kind': 'Java Static Final Method Default Member'}
{'val': 18, 'name': 'nextString', 'kind': 'Java Method Public Member'}
{'val': 22, 'name': 'toString', 'kind': 'Java Static Method Public Member'}
{'val': 25, 'name': 'nextMeta', 'kind': 'Java Method Public Member'}
{'val': 28, 'name': 'nextToken', 'kind': 'Java Method Public Member'}
{'val': 43, 'name': 'parse', 'kind': 'Java Static Method Private Member'}
{'val': 45, 'name': 'parse', 'kind': 'Java Static Method Private Member'}
```

```
-----
Entities with cyclomatic: 14
Java Final Method Protected Member
Java Generic Method Public Member
Java Method Constructor Member Private
Java Method Constructor Member Protected
Java Method Constructor Member Public
Java Method Private Member
Java Method Protected Member
Java Method Public Member
Java Static Final Method Default Member
Java Static Generic Method Private Member
Java Static Method Default Member
Java Static Method Private Member
Java Static Method Protected Member
Java Static Method Public Member
-----
```

```
Project Cyclomatic Summary
Project total cyclomatic: 1088
```

## • testing\_legacy\_code

testing\_legacy\_code

|            | # Cyclomatic | # Entity Set | # Methods |
|------------|--------------|--------------|-----------|
| Understand | 296          | 8            | 246       |
| Our Work   | 296          | 8            | 246       |

```
{ val : 2, name : test1, kind : Java Method Public Member }
{'val': 2, 'name': 'test1', 'kind': 'Java Method Public Member'}
{'val': 2, 'name': 'test1', 'kind': 'Java Method Public Member'}
{'val': 2, 'name': 'test1', 'kind': 'Java Method Public Member'}
{'val': 2, 'name': 'test1', 'kind': 'Java Method Public Member'}
{'val': 2, 'name': 'test1', 'kind': 'Java Method Public Member'}
{'val': 2, 'name': 'test2', 'kind': 'Java Method Public Member'}
{'val': 2, 'name': 'test2', 'kind': 'Java Method Public Member'}
{'val': 2, 'name': 'test2', 'kind': 'Java Method Public Member'}
{'val': 2, 'name': 'test2', 'kind': 'Java Method Public Member'}
{'val': 2, 'name': 'test4', 'kind': 'Java Method Public Member'}
{'val': 2, 'name': 'test4', 'kind': 'Java Method Public Member'}
{'val': 2, 'name': 'test4', 'kind': 'Java Method Public Member'}
{'val': 2, 'name': 'test4', 'kind': 'Java Method Public Member'}
{'val': 2, 'name': 'validate', 'kind': 'Java Method Private Member'}
{'val': 2, 'name': 'validate', 'kind': 'Java Method Private Member'}
{'val': 3, 'name': 'invoke', 'kind': 'Java Method Public Member'}
{'val': 3, 'name': 'invoke', 'kind': 'Java Method Public Member'}
{'val': 3, 'name': 'invoke', 'kind': 'Java Method Public Member'}
{'val': 3, 'name': 'invoke', 'kind': 'Java Method Public Member'}
{'val': 3, 'name': 'invoke', 'kind': 'Java Method Public Member'}
{'val': 5, 'name': 'getTripsByUser', 'kind': 'Java Method Public Member'}
{'val': 5, 'name': 'getTripsByUser', 'kind': 'Java Method Public Member'}
```

```
-----
Entities with cyclomatic: 8
Java Method Constructor Member Private
Java Method Constructor Member Public
Java Method Private Member
Java Method Protected Member
Java Method Public Member
Java Static Generic Method Public Member
Java Static Method Private Member
Java Static Method Public Member
-----
```

```
Project Cyclomatic Summary
Project total cyclomatic: 296
```

- jfreechart

jfreechart

|            | # Cyclomatic | # Entity Set | # Methods |
|------------|--------------|--------------|-----------|
| Understand | 20675        | 19           | 10719     |
| Our Work   | 20675        | 19           | 10719     |

```
{'val': 30, 'name': 'iterateToFindRangeBounds', 'kind': 'Java Static Generic Method Public Memb
{'val': 31, 'name': 'drawItem', 'kind': 'Java Method Public Member'}
{'val': 32, 'name': 'drawItem', 'kind': 'Java Method Public Member'}
{'val': 33, 'name': 'draw', 'kind': 'Java Method Public Member'}
{'val': 33, 'name': 'drawItem', 'kind': 'Java Method Public Member'}
{'val': 33, 'name': 'equals', 'kind': 'Java Method Public Member'}
{'val': 37, 'name': 'draw', 'kind': 'Java Method Public Member'}
{'val': 47, 'name': 'equals', 'kind': 'Java Method Public Member'}
{'val': 47, 'name': 'hashCode', 'kind': 'Java Method Public Member'}
{'val': 50, 'name': 'equals', 'kind': 'Java Method Public Member'}
{'val': 52, 'name': 'equals', 'kind': 'Java Method Public Member'}
{'val': 57, 'name': 'equals', 'kind': 'Java Method Public Member'}
```

```
-----
Entities with cyclomatic: 19
Java Final Method Protected Member
Java Final Method Public Member
Java Generic Method Protected Member
Java Generic Method Public Member
Java Method Constructor Member Default
Java Method Constructor Member Private
Java Method Constructor Member Protected
Java Method Constructor Member Public
Java Method Default Member
Java Method Lambda
Java Method Private Member
Java Method Protected Member
Java Method Public Member
Java Static Final Method Public Member
Java Static Generic Method Public Member
Java Static Method Default Member
Java Static Method Private Member
Java Static Method Protected Member
Java Static Method Public Member
-----
```

```
Project Cyclomatic Summary
Project total cyclomatic: 20675
```

## Count Statement:

- `calculator_app`

|            | # Metric |
|------------|----------|
| Understand | 240      |
| Our Work   | 240      |

- `xerces2j`

|            | # Metric |
|------------|----------|
| Understand | 224969   |
| Our Work   | 224713   |

- `testing_legacy_code`

|            | # Metric |
|------------|----------|
| Understand | 5508     |
| Our Work   | 5536     |

## Count Statement Declaration:

- `calculator_app`

|            | # Metric |
|------------|----------|
| Understand | 129      |
| Our Work   | 127      |

- `xerces2j`

|            | # Metric |
|------------|----------|
| Understand | 72531    |
| Our Work   | 72176    |

- `testing_legacy_code`

|            | # Metric |
|------------|----------|
| Understand | 1840     |
| Our Work   | 1813     |

## Count Statement Execution:

- `calculator_app`

|            | # Metric |
|------------|----------|
| Understand | 111      |
| Our Work   | 113      |

- `xerces2j`

|            | # Metric |
|------------|----------|
| Understand | 152438   |
| Our Work   | 152537   |

- `testing_legacy_code`

|            | # Metric |
|------------|----------|
| Understand | 3668     |
| Our Work   | 3723     |

## Procedure and Challenges

As we all know in every project there are many challenges and problems waiting for us. The first step to every project is always one of the hardest ones since it is like entering a whole new world. As for this project our team tried to read and search about the purpose of this project, so the first step was to find out what Understand is and what it does as a tool.

After searching different sources and reading some parts of this tool's user manual; we figured out that going through your codes for a simple purpose or error is very time and energy consuming, while this tool simply allows you to see most of the aspects your code's structure has.

After figuring out the necessity of this project and attending to the classes with additional notes about the project structure, we had a 5 days period to learn more about the structure of the code and our tasks, by reading the documentation of this project. Then we installed and tried working with different tools needed to complete this project such as DB browser for SQLite and understand itself.

After all the setup was done we tried to study the codes in the GitHub repository and detect the main problems and issues. The next step was how to divide the work between us, although most of the time the team worked together but some of the parts were done individually. The challenging part was how to handle different entity kinds and add all the necessary data to our database tables correctly. Apart from that issue considering all of the possible options was confusing, hard and time consuming. But we tried to get help from the documentation and the type of data that we could see in the parse tree and Understand's supported types. The implementation was done but we still needed to test it, we tried running one of the test cases and checking the tables, which seemed like a success. We also compared it to the results that understand would give us. But as we tried different benchmarks the execution speed was slow and ended up troubling us.

The last part was to write a documentation that indicates all the work and effort done during this time. We tried to write this documentation as simple



and clear as possible, that is to explain every part of our code. But because of the connections the codes have it was hard to maintain a fine flow for the reader's mind and we hope that we succussed doing so.

In the second phase one of the hardest parts were the details we had missed when we tested the benchmark projects and compared the results.

The main code base was kind of easy to develop; but after running the benchmark projects we had to refactor our code to get the result closer and closer to understand's results. Especially in the bigger projects the number that understand would detect was far more bigger than our results. This refactoring and fixing the details took a lot of effort, energy and time. For example in the cyclomatic task understand would count 2 for enums(value, value of) but we thought it is supposed to be only for methods. Or for example it didn't count a cyclomatic for abstract methods, but it did for the methods of an interface class. Since there wasn't any clear source to detect these, we had to go through all of the logs and a lot of comparing had to be done. Therefore we spent and awful lot of time on these details to get the closest results possible.

In the count statement task the main problem was that it had a lot of entities that had to be considered, so the details were even more complex.

Apart for that it would give us the sum of all the statement counts. For example it would give us the sum of statement counts in each of the methods of a class.

## Conclusions and Recommendations

After finishing this project we think Understand is a useful tool and having an open source Api can help a lot to customize and analyze your code. Now that we can spot out each import we can take control of our sources, inheritance and structure of our code, therefore we can prevent spaghetti code from happening and get one step closer to having a clean code. It goes the same for modify, detecting where our entities and variables are changing can help a lot with debugging and to prevent with changes in wrong scopes for our variables.

For future work this project can be developed to much more completed version of itself for example we can add more entity detection and analyze the code more so that each reference has more attributes. After doing so all of the features have to go through more filters of testing, since the runtime isn't in the favor of many test the codes should be optimized many times, each time getting better and more efficient.

## Additional Sources

<https://www.scitools.com/>

<https://documentation.scitools.com/pdf/understand.pdf>

<https://m-zakeri.github.io/OpenUnderstand/>

[https://en.wikipedia.org/wiki/Understand\\_\(software\)](https://en.wikipedia.org/wiki/Understand_(software))

Phase two specific:

[Source code metrics](#)

[Core source code metrics development](#)

<https://www.gatevidyalay.com/compiler-design>

<https://www.gatevidyalay.com/first-and-follow-compiler-design>