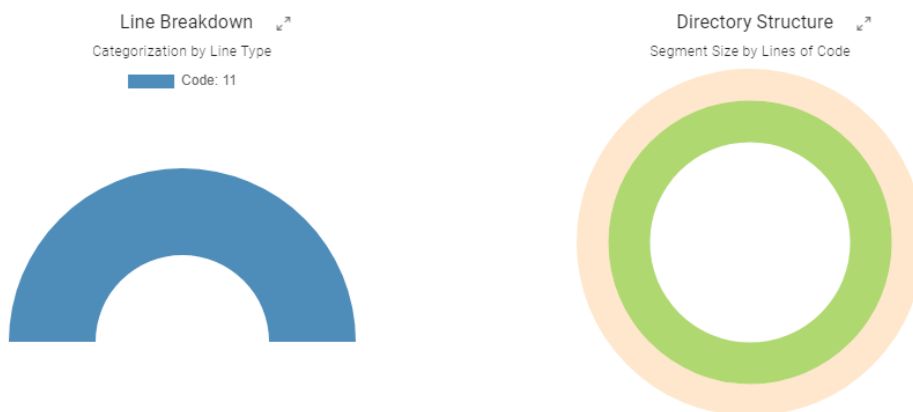


گزارشکار فاز سوم:

کد اولیه:

```
public class PasswordValidatorMain {  
  
    public static void main(String[] args) {  
  
        String input = "Your password input here";  
  
        PasswordValidatorLexer lexer = new  
        PasswordValidatorLexer(CharStreams.fromString(input));  
  
        CommonTokenStream tokens = new CommonTokenStream(lexer);  
  
        PasswordValidatorParser parser = new PasswordValidatorParser(tokens);  
  
        ParseTree tree = parser.password();  
  
        boolean isValid = tree.getText().equals(input);  
  
        System.out.println(isValid);  
  
    }  
  
}
```

نتایج understand



نتایج open understand

11 Lines	1 Files	1 Classes	1 Functions
-------------	------------	--------------	----------------

نتایج تحلیل:

Analysis Log

Summary

1 of 1 project files analyzed
Unanalyzed files: 0
Errors: 0
Warnings: 1
Elapsed time: 00:00:11.21s
Completed at: Monday, January 15, 2024 9:43:30 PM

Java

Analyze Pass1

C:\Users\ArtanTech\Desktop\c4212\p3\javaCode.java
Missing type ParseTree

Analyze Pass2

C:\Users\ArtanTech\Desktop\c4212\p3\javaCode.java

نتایج واقعی :

تعداد متغیرها	تعداد کلاس ها	تعداد ابجکت ها	تعداد ورودی	تعداد خروجی
2	4	4	1	1

بررسی علت نتایج اشتباه understand

در understand تعداد کلاس ها 1 نتیجه گیری شده است. علت ان این است که تنها نام کلاس جاری را انتخاب کرده است ولی باید نام کلاسهای داخل کد را نیز حساب کند.

بعنوان مثال

باید کلاس PasswordValidatorParser را نیز در محاسبات خود مشخص کند.

برای برطرف کردن این مشکل میتوان کد را بگونه ای تغییر داد که ابزار open understand ، understand توانایی تشخیص بهتر ان را داشته باشند.

میتوان کد را اینگونه تغییر داد :

```

public class PasswordValidatorMain {

    public static void main(String[] args) {

        String input = "Your password input here";

        PasswordValidatorLexer lexer = new
        PasswordValidatorLexer(CharStreams.fromString(input));

        CommonTokenStream tokens = new CommonTokenStream(lexer);

        PasswordValidatorParser parser = new PasswordValidatorParser(tokens);

        ParseTree tree = parser.password();

        boolean isValid = tree.getText().equals(input);

        System.out.println(isValid);

    }

    private class PasswordValidatorLexer{

    }

    private class CommonTokenStream {

    }

    private class PasswordValidatorParser {

    }

```

در نتیجه با ایجاد این تغییرات یعنی با ایجاد کلاسهای جداگونه برای هر کلاسی که داخل کد نام آن استفاده شده است میتوان باعث شد که این کلاسها توسط **open understand** , **understand** شناسایی شوند.

18
Lines

1
Files

4
Classes

1
Functions

