

コーディングスタイルの 違いを利用したソフト ウェア再利用の検出

B4 : 西山 知弥

背景

- OSSの種類が多岐にわたり, 開発者が身近にOSSに接するようになった現在, OSSが安易に再利用されがちに
 - 意図しない混入でライセンス違反を起こす事例が発生
- ソースコード混入を目で見てチェックして, 防ぐには限界があり, コストがかかりすぎる

先行研究

- コードクローン・・・ソースコード中での類似または一致した部分
- protexIP・・・開発中のソフトウェアにOSSのソースコードが含まれているかをチェックするツール

研究の動機（着眼点）

- コーディングスタイル？？？なぜ？？？

コードのコピペを検出するのではなく、書き方が違うことを検出できれば良い

- 事前準備のコスト（データ収集コスト）

目的とアプローチ

- **目的**

ソフトウェア(ソースコード)の再利用を検出する

- **アプローチ**

コーディングスタイルの違いを利用する

コーディングスタイル

- プログラムを読みやすくするために, 字下げや空白のあけ方, 変数名の付け方などに定める一定のルール
- プログラミング言語の文法とは異なる

Checkstyle

- Checkstyleは, プログラマーがコーディングスタイルに則ったJavaコードを書くのを補助する開発ツール
- インデントサイズや変数名の付け方, 空白スペースの入れ方など, コーディングスタイルに関するチェックを行うことができる

Checkstyleの構造

```
<?xml version="1.0"?>
<!DOCTYPE module PUBLIC
    "-//Puppy Crawl//DTD Check Configuration 1.3//EN"
    "http://www.puppcrawl.com/dtds/configuration_1_3.dtd">

<module name="Checker">
    <module name="FileLength">
        <property name="max" value="5"/>
    </module>
</module>
```

- Checkstyle の設定は xml で記述する.
- Checkstyle の検証機能は**モジュール**と呼ばれる単位で提供されている.
- 設定ファイルは, このモジュールを指定することで記述する.
- 設定ファイルのルートは, name に Checker を指定した <module> タグにする.
- この下に, 適用したいモジュールを <module> タグで記述していく.
- 上記例では, FileLength というモジュールの使用を宣言している.(FileLength は, ファイルの行数をチェックするモジュール.)
- <module> にパラメータを渡す必要がある場合は, <property> タグを使用する.
- モジュールによってパラメータの有無は異なるので, ドキュメントを参照して必要かどうかを判断する.

今まで行ったこと

- tomcat, spring-framework, wildflyのチェックスタイルの違いを調査し対応表を作成

チェック	tomcat	spring-framework	WildFly
Anotation			
AnotationLocation	○	○	
AnnotationOnSameLine			
AnnotationUseStyle	○	○	
MissingDeprecated			
MissingOverride	○	○	
PackageAnnotation	○	○	○
SuppressWarnings			○
SuppressWarningsHolder			○
	4	4	3

- GitHubからメイン言語がJavaのプロジェクトをスター数で降順にして1000件取得
→その中からcheckstyle.xmlを含むものを選定した

今後すること

- 選定した**100**件のプロジェクトについて, **3**つのプロジェクトのようにチェックスタイル対応表の作成する
→ Pythonでプログラムを書く
- あるプロジェクトのルールを, 他のプロジェクトに当てはめて違反がどれくらい起こるか確認