

Метапрограмування

Лабораторна робота № 2

Проектування та розробка статичного аналізатора програмних кодів

© Д.О. Терлецький, Г.І. Гогерчак 2020.

© Київський національний університет імені Тараса Шевченка, 2020.

Завдання

Використовуючи мову програмування Python розробити у вигляді окремого пакету для `pip` консольну інтерактивну утиліту статичного аналізу та модифікації програмних кодів для певної мови програмування. Пакет повинен бути опублікований на pypi.org. Система повинна вміти:

1. **Виконувати аналіз структури дерева проєкту** та виявляти файли, що містять програмні коди певною мовою програмування (в залежності від обраного варіанту) шляхом аналізу розширень файлів.
2. **Виконувати аналіз дотримання конвенцій щодо іменування та документування** в рамках окремих файлів з програмними кодами на певній мові програмування (в залежності від обраного варіанту) і виявляти невідповідності та помилки. У випадку виявлення невідповідностей та помилок, система повинна здійснювати відповідні записи у спеціальний файл `*_verification.log`, вказуючи повний шлях до файлу, номер рядка та тип помилки чи невідповідності у форматі

`Id. File Path: Line Number - Error Code: Error Message`

3. **Виконувати модифікації окремих частин програмних кодів**, які не відповідають конвенціям щодо іменування та документування для обраної мови програмування, з метою приведення програмних кодів до максимальної відповідності даним конвенціям. Під час модифікації програмних кодів система повинна здійснювати відповідні записи у спеціальний файл `*_fixing.log`, вказуючи повний шлях до файлу, номер рядка та виконану модифікацію у форматі

Id. File Path: Line Number - Modification Description

В описі пакету потрібно чітко вказати які саме конвенції щодо найменування та документування перевіряє та виправляє розроблений пакет.

Функціональні вимоги до системи

Забороняється використовувати будь-які сторонні інструменти для лексичного або синтаксичного аналізу програмних кодів (готові лексери, спеціалізовані бібліотеки, тощо) як частини системи.

Потрібно реалізувати:

- Консольну утиліту (у вигляді окремого пакету для `pip`), яка повинна інстальватися за допомогою `pip` та запускатися з командного рядка терміналу операційної системи і мати довідку з описом усіх основних команд та ключів.
- Окремі команди та набір ключів для виконання усіх функцій системи.
- Набір параметрів та ключів для аналізу та модифікації окремих файлів, каталогів з файлами, каталогів з підкаталогами та файлами. Орієнтований формат синтаксису:

```
python3 SLangCCF --verify -(p|d|f) /..  
python3 SLangCCF -v -(p|d|f) /..
```

```
python3 SLangCCF --fix -(p|d|f) /..  
python3 SLangCCF -f -(p|d|f) /..
```

```
python3 SLangCCF --help  
python3 SLangCCF -h
```

SLang - source language
CCF - code conventions fixer

```
-p - project  
-d - directory  
-f - file
```

/.. - path to project, directory or file

Тестування та демонстрація:

- Для часткового тестування роботи системи потрібно порівнювати результати її роботи з відповідними результатами аналізу програмних кодів за допомогою [IDEs JetBrains](#), які дозволяють розробку програмного забезпечення за допомогою відповідної мови програмування зазначеної в обраному варіанті.
- Для демонстрації роботи системи потрібно використовувати каталоги та файли із базової версії обраної мови програмування та/або з [GitHub](#) або [GitLab](#).

Варіанти

Варіанти полегшеної складності

1. **Go. (7 балів)** Детальна інформація про конвенції:
 - [Effective Go](#)
 - [Effective Go: Commentary](#)
 - [Effective Go: Names](#)
2. **CoffeeScript. (8 балів)** Детальна інформація про конвенції:
 - [CoffeeScript Style Guide](#)
 - [CoffeeScript Style Guide: Naming Conventions](#)
 - [CoffeeScript Style Guide: Comments](#)

Варіанти середньої складності

3. **Dart. (9 балів)** Детальна інформація про конвенції:
 - [Dart Style Guide](#)
 - [Dart Style Guide: Names](#)
 - [Dart Style Guide: Comments](#)
4. **Kotlin. (9 балів)** Детальна інформація про конвенції:
 - [Kotlin Coding Conventions](#)
 - [Kotlin Coding Conventions: Naming rules](#)
 - [Kotlin Coding Conventions: Documentation comments](#)
5. **Rust. (9 балів)** Детальна інформація про конвенції:
 - [Rust Style Guidelines](#)
 - [Rust Style Guidelines: Naming conventions](#)
 - [Rust Style Guidelines: Comments](#)
6. **PHP. (10 балів)** Детальна інформація про конвенції:
 - [PHP Style Guide](#)
 - [PHP Style Guide: 6. Comments](#)
 - [PHP Style Guide: Filename](#)
 - [PHP Style Guide: 5.2. Namespace Name](#)
 - [PHP Style Guide: 9.1. Function Name](#)
 - [PHP Style Guide: 11.3. Class Name](#)
 - [PHP Style Guide: 11.4. Class Documentation](#)

7. **Swift. (10 балів)** Детальна інформація про конвенції:

- [Swift Style Guide](#)
- [Swift Style Guide: File Names](#)
- [Swift Style Guide: Naming](#)
- [Swift Style Guide: Documentation Comments](#)

Варіанти підвищеної складності

8. **Java. (11 балів)** Детальна інформація про конвенції:

- [Java Code Conventions](#)
- [Java Code Conventions: 2 - File Names](#)
- [Java Code Conventions: 3.1.1 Beginning Comments](#)
- [Java Code Conventions: 5 - Comments](#)
- [Java Code Conventions: 9 - Naming Conventions](#)

9. **Ruby. (11 балів)** Детальна інформація про конвенції:

- [The Ruby Style Guide](#)
- [The Ruby Style Guide: Naming Conventions](#)
- [The Ruby Style Guide: Comments](#)
- [The Ruby Style Guide: Comment Annotations](#)
- [The Ruby Style Guide: Magic Comments](#)

10. **C#. (12 балів)** Детальна інформація про конвенції:

- [C# Code Style Guide](#)
- [C# Code Style Guide: Comments](#)
- [C# Code Style Guide: Documentation Comments](#)
- [C# Code Style Guide: Naming Rules](#)

11. **C++. (12 балів)** Детальна інформація про конвенції:

- [Google C++ Style Guide](#)
- [Google C++ Style Guide: Naming](#)
- [Google C++ Style Guide: Comments](#)

12. **JavaScript. (12 балів)** Детальна інформація про конвенції:

- [Google JavaScript Style Guide](#)
- [Google JavaScript Style Guide: 2.1 File name](#)
- [Google JavaScript Style Guide: 6 Naming](#)

- [Google JavaScript Style Guide: 7 JSDoc](#)

13. **Python. (12 балів)** Детальна інформація про конвенції:

- [PEP 8 – Style Guide for Python Code](#)
- [PEP 8 – Style Guide for Python Code: Comments](#)
- [PEP 8 – Style Guide for Python Code: Naming Conventions](#)

14. **Scala. (12 балів)** Детальна інформація про конвенції:

- [Scala Style Guide](#)
- [Scala Style Guide: Naming Conventions](#)
- [Scala Style Guide: Scaladoc](#)