

# Dokumentacja wstępna

Autor: Artur Wyróżębski

## Opis projektu

Napisać program, który analizuje kod w podzbiorze Python i znajduje nieużywane elementy programu: zmienne, stałe, klasy, funkcje, metody, konstruktory, etc.

## Zawartość podzbioru języka Python

Analizowany będzie język Python w wersji 3.7.

Dopuszczalne są:

funkcje, klasy, instrukcje warunkowe, pętle, zmienne, moduły, komentarze, standardowe struktury danych (list, range, dict, set), adnotacje typu, wyrażenia lambda.

## Wymagania funkcjonalne

- Odczytywanie i analiza kodu źródłowego języka Python zapisanego w plikach tekstowych oraz ze standardowego wejścia.
- Poprawne ukazanie wszystkich nieużywanych elementów kodu źródłowego na standardowym wyjściu.
- Branie pod uwagę adnotacji typów zmiennych podczas analizy operacji na tych zmiennych i wyświetlenie błędu, jeżeli typy są ze sobą niekompatybilne.

## Wymagania niefunkcjonalne

- Komunikat o nieużywanym elemencie powinien wskazywać miejsce w kodzie, w którym znajduje się ten element oraz przedstawiać nazwę tego elementu i jego rodzaj (klasa, zmienna, funkcja etc.).
- Komunikat o błędach analizy powinien być prosty i przejrzysty.

## Środowisko i technologie

Do utworzenia programu wykorzystany zostanie język Java w wersji 14.

Do tworzenia testów jednostkowych wykorzystania zostanie biblioteka JUnit.

Projekt jest aplikacją konsolową.

## Sposób uruchomienia

Program jest aplikacją konsolową, którą uruchamia się poprzez wywołanie z parametrem reprezentującym ścieżkę do pliku z kodem źródłowym. Program może być również uruchomiony bez jakichkolwiek argumentów, wtedy przyjmuje on kod źródłowy ze standardowego wejścia.

Nazwa programu to 'linter'.

Pliki do analizy muszą kończyć się rozszerzeniem .py.

Przykłady uruchomienia:

```
./linter plik1.py
```

```
./linter
```

## Wyjście:

### **Format wyświetlenia nieużywanego elementu:**

*Unused : wiersz;kolumna (pozycja w źródle) : nazwa : rodzaj elementu*

Rodzaje elementów:

variable, function, class, method, module

Przykłady:

Unused : 3;5 : suma : variable

Unused : 10;3 : fun : function

### **Format wyświetlania błędów:**

*Error : wiersz;kolumna : rodzaj błędu*

Przykłady:

Error : 10;5 : Incompatible types

Error : 15;3 : Undefined name

## Sposób testowania

Napisane zostaną testy jednostkowe oraz integracyjne projektu sprawdzające poprawność implementacji.