

Лабораторна робота 11

Оцінка якості коду

Мета

Дослідити алгоритми визначення якості коду

Завдання

Використовуючи код будь-якого додатку (неважливо, чи він створений вами, або взятий з github/gitlab):

- рівень якості програмування
- складність розуміння програми
- трудомісткість кодування програми
- цикломатичне число Мак-Кейба
- метрика Чепіна

Метрики потрібно виконувати для:

- source code
- декомпільованого коду


В звіті: Зробити висновки впливу компіляції та обфускації на якість коду

Описати алгоритми / формули для обчислення реалізованих характеристик якості коду

При виконанні роботи рекомендовано використовувати: Roslyn (тільки для .net) або Antrl Будь-які інші бібліотеки для аналізу синтаксичного дерева потрібно обговорювати з викладачем

Хід роботи

За допомогою flake8 виконуємо аналіз коду



```
Terminal: Local (2) x + v
vasyl.oliinyk@VAOLIINY-M-71MZ src % python3 -m flake8 rsa.py
rsa.py:4:1: F401 'sys' imported but unused
rsa.py:87:1: E722 do not use bare 'except'
vasyl.oliinyk@VAOLIINY-M-71MZ src %
```

Рис 1 - Результати аналізу

Підраховуємо метрику Мак Кайба

```
vasyl.oliinyk@VAOLIINY-M-71MZ src % python3 -m mccabe rsa.py
7:0: 'multiplicativeInverse' 1
11:0: 'extendedEuclid' 2
20:0: 'convertMessageToNumber' 2
28:0: 'getRandomPrime' 2
35:0: 'isPrime' 4
46:0: 'isComposite' 5
64:0: 'decompose' 2
71:0: 'getKeys' 3
TryExcept 84 3
Loop 95 2
If 116 2
vasyl.oliinyk@VAOLIINY-M-71MZ src %
```

Рис 2 - Результати утіліти mccabe

Утіліта Radon призначена для аналізу та обчислення різних метрик

```
vasyl.oliinyk@VAOLIINY-M-71MZ src % python3 -m radon cc rsa.py
rsa.py
  F 46:0 isComposite - B
  F 35:0 isPrime - A
  F 71:0 getKeys - A
  F 11:0 extendedEuclid - A
  F 20:0 convertMessageToNumber - A
  F 28:0 getRandomPrime - A
  F 64:0 decompose - A
  F 7:0 multiplicativeInverse - A
vasyl.oliinyk@VAOLIINY-M-71MZ src %
```

Рис 3 - Результати виконання radon cyclomatic complexity

CC score	Rank	Risk
1 - 5	A	low - simple block
6 - 10	B	low - well structured and stable block
11 - 20	C	moderate - slightly complex block
21 - 30	D	more than moderate - more complex block
31 - 40	E	high - complex block, alarming
41+	F	very high - error-prone, unstable block

Рис 4 - Таблиця результатів Radon CC

```

vasyl.oliinyk@VAOLIINY-M-71MZ src % python3 -m radon raw rsa.py
rsa.py
  LOC: 117
  LLOC: 87
  SLOC: 87
  Comments: 0
  Single comments: 0
  Multi: 0
  Blank: 30
  - Comment Stats
    (C % L): 0%
    (C % S): 0%
    (C + M % L): 0%
vasyl.oliinyk@VAOLIINY-M-71MZ src %

```

Рис 5 - Результати виконання Radon raw

```

vasyl.oliinyk@VAOLIINY-M-71MZ src % python3 -m radon mi rsa.py
rsa.py - A
vasyl.oliinyk@VAOLIINY-M-71MZ src %

```

Рис 6 - Результати виконання Radon mi

MI score	Rank	Maintainability
100 - 20	A	Very high
19 - 10	B	Medium
9 - 0	C	Extremely low

Рис 7 - Таблиця результатів radon maintainability index

Висновок: у результаті лабораторної роботи я дослідив алгоритм визначення якості коду