生成AIG テーマ講義
「品質とメトリクス」

増田研究室生成AIG　小池温大

講義の準備

- 演習を行うので、Onedriveにある下のzipファイルを自分たちのサーバに送ってください。
- 情シス_増田研究室/24_卒研/doc/テーマ講義/生成AIG/theme_practice.zip
  - 分からない方は、onedriveの情シス_増田研究室/24_卒研/doc/増田研サーバ環境の使い方を参照してください

今日のアジェンダ

1.ソフトウェアメトリクス

2.サイクロマティック複雑度(CCN)とその演習

1.ソフトウェアメトリクス

- 皆さんはコーディングする時に何か気を付けていることはありますか?

- 品質がいいのは何をもって品質がいいですか？
  - 開発者にとっては、可読性、バグ検出数、
  - 顧客にとっては、操作性、機能性

1.ソフトウェアメトリクス

- ソフトウェアの品質や開発プロセスを数値で評価する方法。
- 数値化することで問題を明確化し、改善できるようにするのが目的
- 例えばソースコードの複雑さを測ることで、読みやすいかどうか、保守できるかどうかを判断することをできる。

1.ソフトウェアメトリクス

- プロダクトメトリクス
  - ソフトウェア製品の特性におけるメトリクス
  - 循環的複雑度、LOC、テストカバレッジ、機能性、保守性
- プロセスメトリクス
  - ソフトウェア開発プロセスにおけるメトリクス
  - バグ発見数、トレーサビリティ、DRE(欠陥除去率)
- プロジェクトメトリクス
  - プロジェクトの健全性と進捗におけるメトリクス
  - 人〇、予算遵守率(CPI)、スケジュール遵守率(SPI)

2.循環的複雑度(CCN)

- サイクロマティック複雑度、CCN
- プログラムの複雑さを表す指標のひとつ。
- 制御文や条件分岐に基づいて測定される。
- テストケースを設計する時にも使われる。

表 1　循環的複雑度(CCN)の解析

複雑度	目安
1～10	シンプルで理解しやすいプログラム
11～20	若干複雑なプログラム
21～	複雑なプログラムであり、完全に理解してテストを作成するのが困難
50～	保守困難、バグが確実に潜在する プログラム

2.循環的複雑度(CCN)演習

- Lizard
  - ソフトウェアメトリクス計測ツール

  - ソースコードのあるフォルダにlizardをかけることで循環的複雑度やLOC(コード行数)などのソフトウェアメトリクスを調べることができる。

2.循環的複雑度(CCN)演習

1.増田研究室のサーバに入る

ssh root@swelab2.mc.yc.tcu.ac.jp –p sshポート番号

2.Zipの解凍

unzip theme_practice.zip

3.lizard のインストール

pip install lizard

4.lizardのインストール確認

lizard

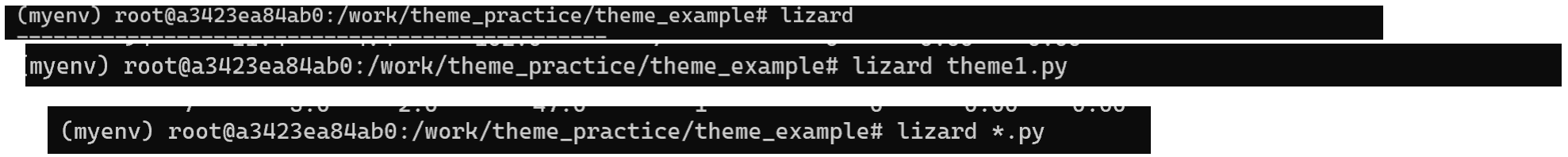
何か出てきたらOK

2.演習1

1.theme_practiceフォルダに入る。

2.theme_exampleフォルダに入る。

3.Lizardコマンドを打って、循環的複雑度を調べてください。
そしてどうしてこのような数値になっているかを調べてください。



.2.循環的複雑度(CCN)演習

```
(myenv) root@a3423ea84ab0:/work/theme_practice/theme_example# lizard
================================================
  NLOC    CCN   token  PARAM  length  location
------------------------------------------------
      10      3     70      0      13 main@1-13@./.ipynb_checkpoints/theme2-checkpoint.py
      13      6    114      0      16 main@1-16@./.ipynb_checkpoints/theme7-checkpoint.py
      17      6    141      0      20 main@1-20@./.ipynb_checkpoints/theme5-checkpoint.py
       5      2     47      0       9 main@1-9@./.ipynb_checkpoints/theme1-checkpoint.py
      11      4     83      0      14 main@1-14@./.ipynb_checkpoints/theme4-checkpoint.py
      13      6    114      0      16 main@1-16@./.ipynb_checkpoints/theme6-checkpoint.py
      11      4    145      0      11 main@1-11@./.ipynb_checkpoints/theme3-checkpoint.py
      17      6    141      0      20 main@1-20@./theme5.py
      11      4     83      0      14 main@1-14@./theme4.py
      13      6    114      0      16 main@1-16@./theme6.py
      13      6    114      0      16 main@1-16@./theme7.py
      10      3     70      0      13 main@1-13@./theme2.py
       5      2     47      0       9 main@1-9@./theme1.py
13 file analyzed.
==============================================================
NLOC    Avg.NLOC  AvgCCN  Avg.token  function_cnt    file
--------------------------------------------------------------
     12      10.0     3.0       70.0         1     ./.ipynb_checkpoints/theme2-checkpoint.py
     15      13.0     6.0      114.0         1     ./.ipynb_checkpoints/theme7-checkpoint.py
     19      17.0     6.0      141.0         1     ./.ipynb_checkpoints/theme5-checkpoint.py
      7       5.0     2.0       47.0         1     ./.ipynb_checkpoints/theme1-checkpoint.py
     13      11.0     4.0       83.0         1     ./.ipynb_checkpoints/theme4-checkpoint.py
     15      13.0     6.0      114.0         1     ./.ipynb_checkpoints/theme6-checkpoint.py
     13      11.0     4.0      145.0         1     ./.ipynb_checkpoints/theme3-checkpoint.py
     19      17.0     6.0      141.0         1     ./theme5.py
     13      11.0     4.0       83.0         1     ./theme4.py
     15      13.0     6.0      114.0         1     ./theme6.py
     15      13.0     6.0      114.0         1     ./theme7.py
     12      10.0     3.0       70.0         1     ./theme2.py
      7       5.0     2.0       47.0         1     ./theme1.py

===============================================================================================================
No thresholds exceeded (cyclomatic_complexity > 15 or length > 1000 or nloc > 1000000 or parameter_count > 100)
==========================================================================================
Total nloc   Avg.NLOC  AvgCCN  Avg.token   Fun Cnt  Warning cnt   Fun Rt   nloc Rt
------------------------------------------------------------------------------------------
Total nloc   Avg.NLOC  AvgCCN  Avg.token   Fun Cnt  Warning cnt   Fun Rt   nloc Rt
```

2.演習1

```
(myenv) root@a3423ea84ab0:/work/theme_practice/theme_example# lizard
================================================
  NLOC    CCN   token  PARAM  length  location
------------------------------------------------
      13      6    114      0      16 main@1-16@./theme6.py
      13      6    114      0      16 main@1-16@./theme7.py
      17      6    141      0      20 main@1-20@./theme5.py
      11      4     83      0      14 main@1-14@./theme4.py
       5      2     47      0       9 main@1-9@./theme1.py
      11      4    145      0      11 main@1-11@./theme3.py
      10      3     70      0      13 main@1-13@./theme2.py
7 file analyzed.
==============================================================
NLOC    Avg.NLOC  AvgCCN  Avg.token  function_cnt    file
--------------------------------------------------------------
     15      13.0     6.0      114.0         1     ./theme6.py
     15      13.0     6.0      114.0         1     ./theme7.py
     19      17.0     6.0      141.0         1     ./theme5.py
     13      11.0     4.0       83.0         1     ./theme4.py
      7       5.0     2.0       47.0         1     ./theme1.py
     13      11.0     4.0      145.0         1     ./theme3.py
     12      10.0     3.0       70.0         1     ./theme2.py
```

3.演習1

- theme1.py 単純ループを使った例　　CCN:2
- theme2.py 二重ループを使った例　　CCN:3
- theme3.py　三重ループを使った例　　CCN:4
- theme4.py　二重ループ+if文を使った例　　CCN:4
- theme5.py　theme6.pyに分岐を増やした例　CCN:6
- theme6.py　二重ループを2個使った例　　CCN:6
- theme7.py　theme6.pyにif文を使った例　　CCN:6

演習2

- themepractice.pyを開いてください。
  - そしてthemepractice.pyにlizardをかけてみてください
  - きっと循環的複雑度(CNN)が高くなっているかと思います。
  - こちらを結果を変えずにコードを修正して循環的複雑度を低くしてください。
  - 循環的複雑度を10以下までを目標とします。

クイズ

- 以下の5つのメトリクスがプロダクトメトリクス、プロセスメトリクス、プロジェクトメトリクスのいずれに入るか答えよ

  - 信頼性
  - 人月
  - モジュール結合度
  - DRE
  - トレーサビリティ

クイズ

- 循環的複雑度の説明として正しいか間違っているか答えよ。

  1.制御文や条件分岐が使われていない順次プログラムの循環的複雑度は0である。

  2.サイクロマティック複雑度ともいう。

  3.テストケースを設計する際にも参考になる指標である。

  4.循環的複雑度が高ければ高いほど、品質はいい

4.自分の研究

- グループ:生成AIG
- 研究内容
  - ChatGPTに競技プログラミングサイト、Atcoderの問題を解かせ、正確性や循環的複雑度、LOCなどのソフトウェアメトリクスから生成AIが作るコードの品質を調べる研究

  - 対象:Atcoder,BaekJoon