

# **pyrevise: プログラミング教育を対象にした 生成AI によるエラー文の理解支援**

**京都産業大学**

**玉田研究室**

**学籍番号：153678**

**氏 名：城越 悠仁**

# 概要

- プログラミング初学者を対象とする.
- 生成AIを用いてプログラミングへの理解度を高める.
- 実行時に出力されるエラー文に着目する.
- プログラミング初学者の学習支援ツールの作成を行う.

# 背景

## プログラミングを苦手になる理由

- エラー文を理解することができない。
- エラー箇所を特定することができない。
- エラー文が長くて、どこを読めばいいか分からない。
- デバッグ方法が分からない。



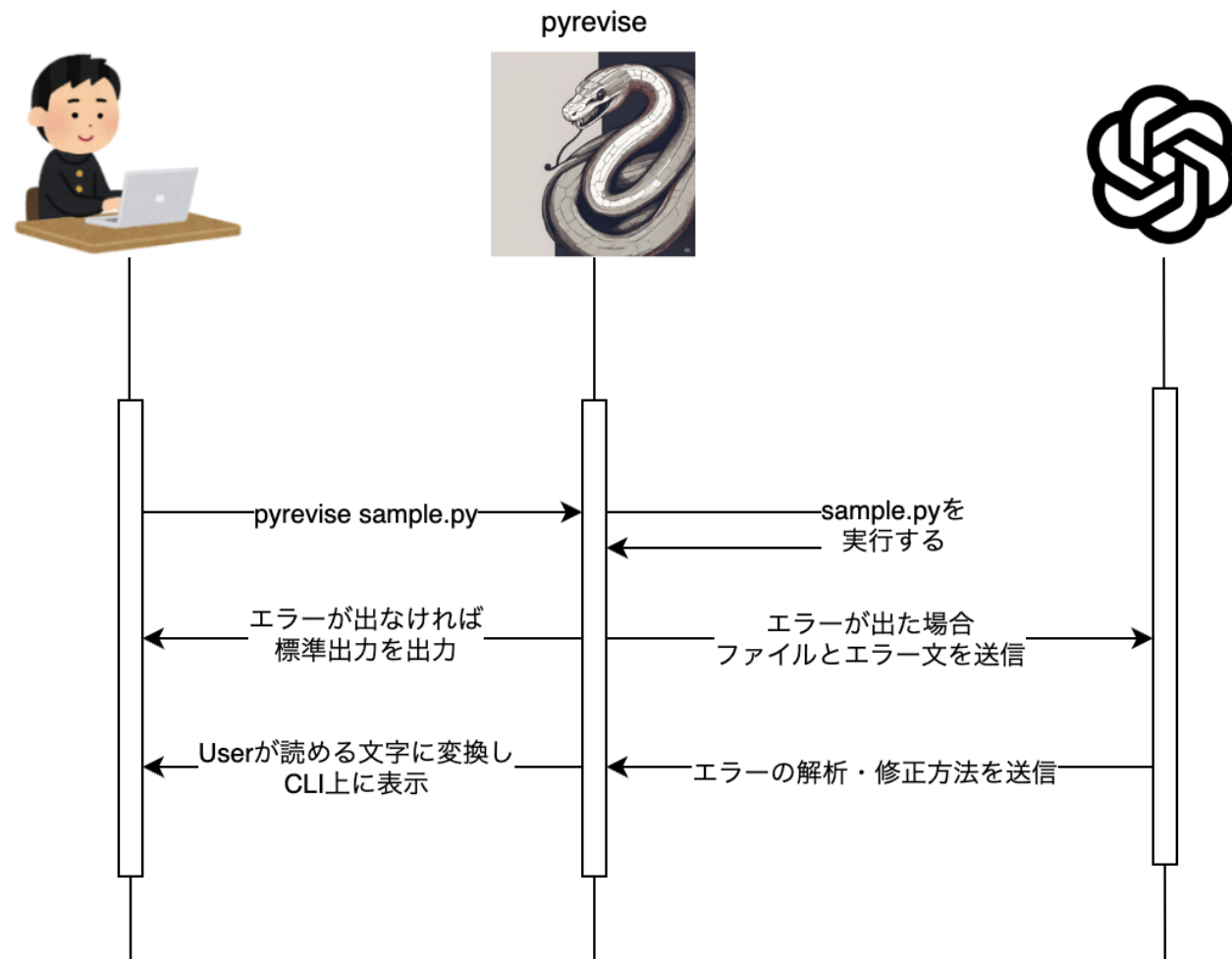
# 背景

## 近年の生成AIの発展

- 生成AIの発展は様々な可能性がある.
- 教育分野で生成AIは幅広く活用されている.
- ChatGPTを用いてツールを開発する.



# pyreviseの仕様



# pyreviseの機能

## 実行ファイル

```
1 def divide_by_zero():
2     return 10 / 0
3
4 divide_by_zero()
```

## 実行例

```
~ $ pyrevise divide_by_zero.py
エラーメッセージ:
Traceback (most recent call last):
  File "/app/input/divide_by_zero.py", line 5, in <module>
    divide_by_zero()
  File "/app/input/divide_by_zero.py", line 2, in divide_by_zero
    return 10 / 0
    ~~~^~~
ZeroDivisionError: division by zero
```

### 解析結果:

このプログラムコードはゼロ除算エラーを引き起こしています。プログラムが10を0で割ろうとしています。

Pythonでは、ゼロでの除算は許容されていません。したがって、Pythonはゼロ除算エラーをスローします。

このエラーを修正するためには、割る数が0でないことを確認する必要があります。エラー処理として以下のように修正することができます。

```
```python
def divide_by_zero(x, y):
    if y == 0:
        return "Cannot divide by zero"
    else:
        return x / y

divide_by_zero(10, 0)
```
```

これにより、関数が0での除算を試みるときに、エラーメッセージが返されます。

また、例外処理を使用して、例外がスローされたときにプログラムが停止しないようにすることもできます。

```
```python
def divide(x, y):
    try:
        return x / y
    except ZeroDivisionError:
        return "Cannot divide by zero"

divide(10, 0)
```
```

この関数では、`ZeroDivisionError`が発生すると、この特定のエラーメッセージが返されます。関数は、エラーが発生した場合でも、プログラム全体を停止させることなく処理を続けることができます。

# pyreviseの機能

## ログ機能

|                |       |                             |                    |  |
|----------------|-------|-----------------------------|--------------------|--|
| 2025/1/27 3:03 | ERROR | /app/input/status_values.py | 12.960866212844849 | File<br>"/app/input/status_values.py", line<br>20<br>print(¥n)<br>^<br>SyntaxError : unexpected character<br>after line continuation character   |
| 2025/1/27 3:04 | INFO  | /app/input/status_values.py | 0.015949965        | max: 985 min: 2 sum: 43036<br>average: 430.36<br><br>86 511 691 509 117 34 91<br>232 133 707<br><br>674 543 280 655 289 315 498<br>657 395 464<br>495 561 26 53 248 980 312<br>835 226 190<br><br>786 248 303 317 983 264 246<br>592 83 291<br><br>857 97 295 271 231 151 719<br>879 16 476<br><br>135 7 888 299 2 488 282 731<br>51 312 |

- 学習者の復習に活用可能である.
- このログファイルを活用し,実験を行う.

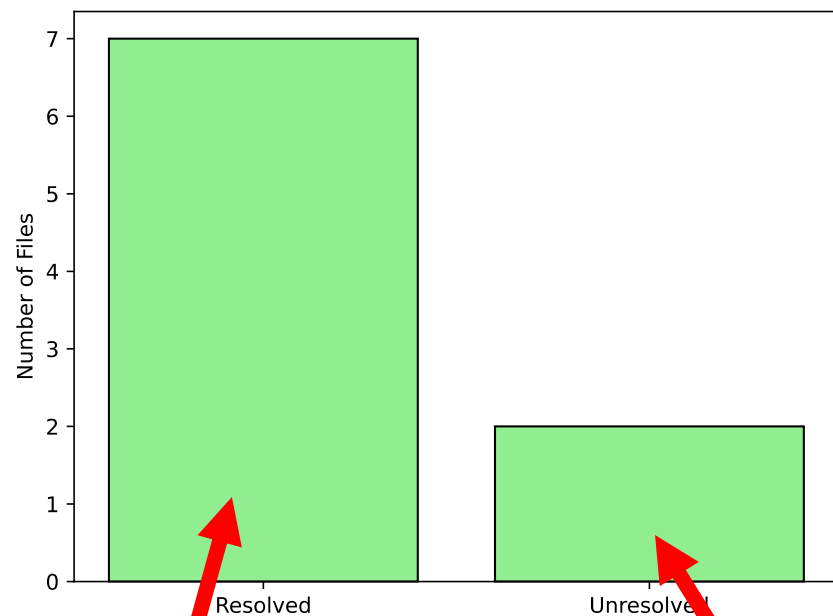
# 評価実験

- 1時間という制限時間の中で6人の学生にツールを使用してもらう。
- Python初学者に本学の基礎プログラミング演習Ⅱの授業の課題を解いてもらう。
- ログファイルのデータから4つの指標を基にツールの有用性を調査する。
  - エラー修正速度の推移
  - エラー発生頻度
  - エラー解決率の割合
  - エラー再発防止率の推移



# 結果

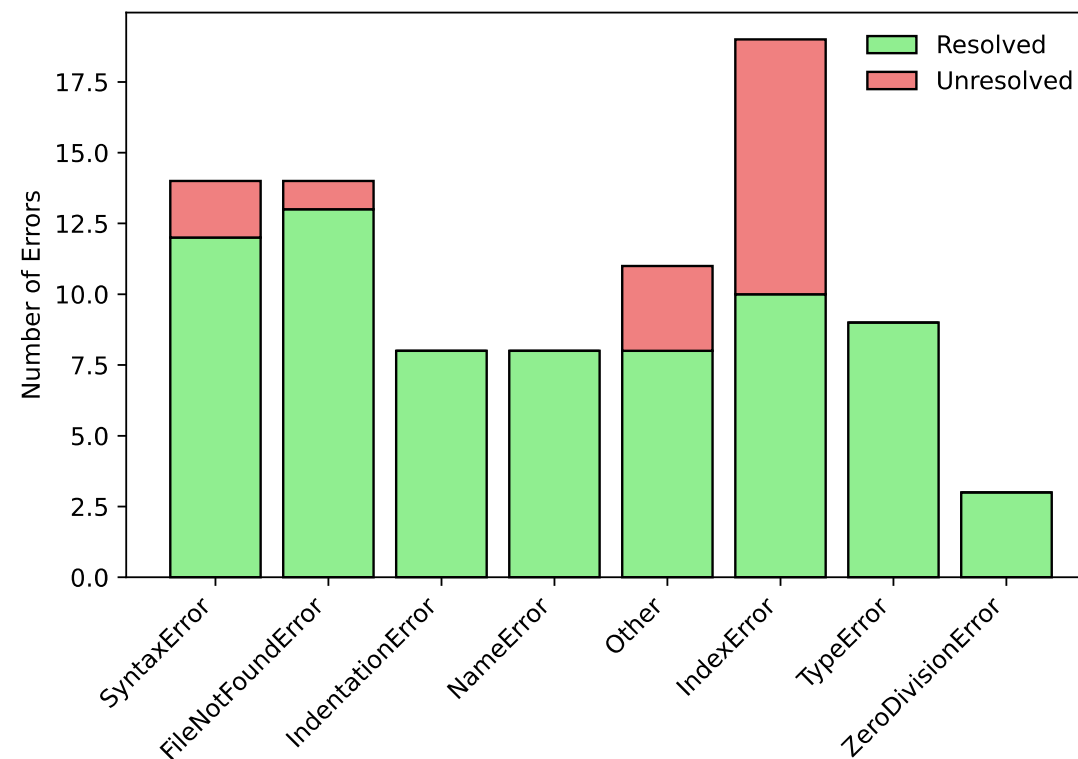
## エラー解決率



エラーを  
解決できた  
ファイル数

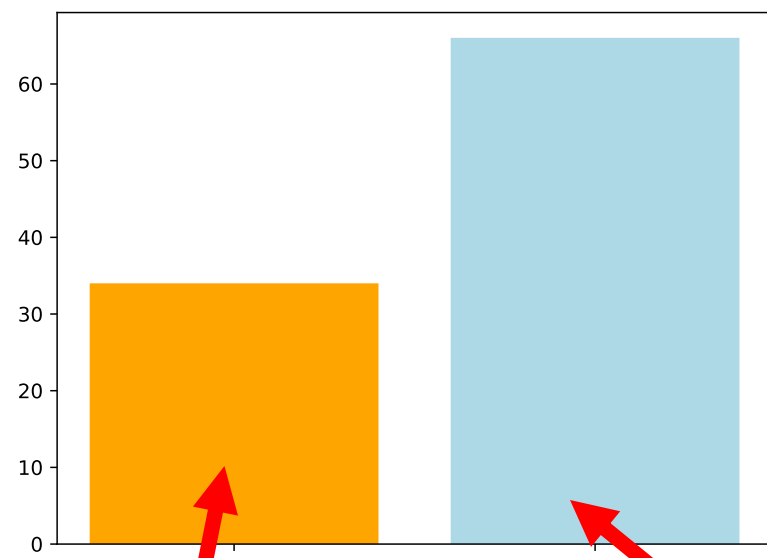
エラーを解決  
できなかった  
ファイル数

## エラー種別解決率



# 結果

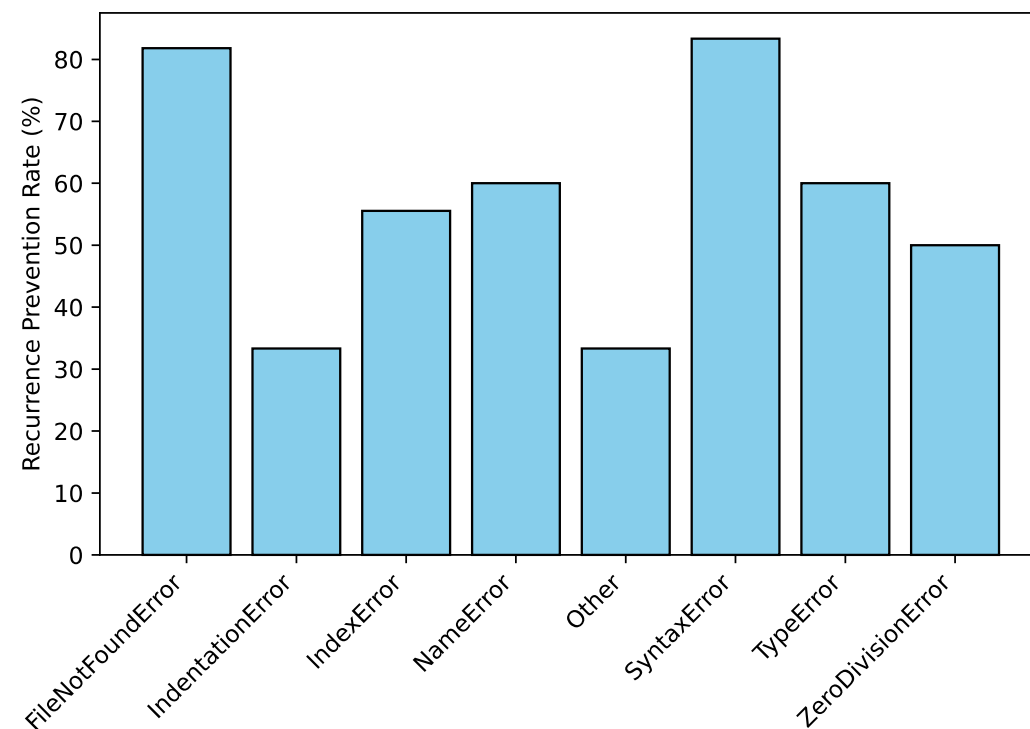
## エラー再発防止率



エラーが再発した確率

エラーの再発を  
防止できた確率

## エラー種別再発防止率



# まとめ

- 本ツールは学習者に効果的な支援をしたといえる.
- 全てのエラーを解決することができなかった.
- フィードバックの最適化などの改善の余地がある.