

# 課題の実装の進め方

須藤功平

株式会社クリアコード

実践リーダーブルコード

2015-06-24

# 実装する目的

午後の作業の布石



- ✓ 課題の仕様を理解するため
- ✓ 読まれることを意識して書かれたコードを用意するため
- ✓ リーダブルなコードを書いて！

# 目的じゃないこと

- ✓ テクニックをたくさん覚える
- ✓ 難しいプログラムでも実装できるようになる
- ✓ 速く実装できるようになる

# 課題の傾向

- ✓ 技術的に難しいことは意図的に避けている
- ✓ 段階的に改良していく
- ✓ 時間内で実装しきれない分量
- ✓ **すべて実装する必要はない!**

# メモを書く

- ✓ 工夫したことはissueに書く
  - ✓ 1つのissueに1つの工夫でOK
- ✓ メモに含めること
  - ✓ 実際のコードのURL
  - ✓ リーダブルな理由

# メモ例

タイトル: 統一されたスタイル ← リーダブルな書き方の名前

↓本文↓

--

URL: <https://github.com/kou/.../commit/eb02be>

コード:

```
@@ -64,7 +64,7 @@ void add_recipe_to_RecipeList(...) {
    void open_RecipeList(RecipeList *list, char *path) {
        FILE *fp;
        /* ファイルを開く */
-   if ((fp = fopen(path, "r")) == NULL){
+   if ((fp = fopen(path, "r")) == NULL) {
        fprintf (stderr, "ファイルが開けません\n");
        fclose(fp);
        exit(EXIT_FAILURE);
```

理由:

ファイル内でスタイルを統一しているので読みやすくなっている。

# リーダブルコードとは

- ✓ 読む人基準
  - ✓ 「読む人」が 読みやすいならリーダブル
- ✓ 読む人の視点を意識してみて
  - ✓ どんな人が読む？
  - ✓ どんな前提知識がある？

# 困ったら1

- ✓ チューターに相談
  - ✓ チューターは**答えを教えない**
  - ✓ 一緒に考えてくれる
- ✓ **答えを教えない理由**
  - ✓ 参加者が考える機会を奪わないため
  - ✓ 一緒に考えると新しい視点が増える

# 困ったら2

- ✓ まわりの参加者に相談
  - ✓ 答えを教えてもよい
  - ✓ 一緒に考えてもよい
- ✓ 条件
  - ✓ 答えを教えるときは理由も伝える  
(教える参加者が考える機会になる！)
  - ✓ 聞く人は理由も聞く  
(考え方を知ると新しい視点でコードを読める！)

# 実装開始

## ✓ 仕様

- ✓ task.md

## ✓ 練習

- ✓ 講師がtask.mdを読んで説明
- ✓ 仕様1をみんなで実装
- ✓ 実装できたら休憩→仕様2以降実装