

ソフトウェアテスト

[10] リファクタリング演習

Software Testing
[10] Exercise: Refactoring

あまん ひろひさ
阿萬 裕久 (AMAN Hirohisa)
aman@ehime-u.ac.jp

演習の目的

- プログラムの**リファクタリング**を行い, 分かりやすく書くことの重要性を体験・学習する
- 主な内容
 - インデント, 変数名, 関数名が整理されていない**分かりにくいプログラム**の改善
 - 長い**整理されていないプログラム**の改善
 - **リファクタリングとともに回帰テストを実施**

重要性(じゅうようせい) : importance

体験(たいけん) : experience

分かりにくい(わかりにくい) : hard to comprehend

整理されていない(せいりされていない) : poorly organized, poorly written

回帰テスト(かいきテスト) : regression testing

演習の内容

演習は1人で行うか、または、
2人で協力してペアプログラミングを行ってもよい
(どちらのスタイルでもよい)

□ 課題1

インデント, 変数名, 関数名が全く整理されていない
Cプログラムをリファクタリングする
ただし, 回帰テストも一緒に行う

□ 課題2

与えられたCプログラムをリファクタリングする
ただし, 回帰テストも一緒に行う

(C) 2007-2024 Hirohisa AMAN

3

Note: you can do these exercises in a group of two students (pair programming), or you can do them by yourself.

課題1(かだい1): exercise-1

Given a poorly written C program, refactor the source code and do regression testing.

課題2(かだい2): exercise-2

Given a poorly organized C program, refactor the source code and do regression testing.

課題1 (1/2)

【対象プログラム】 `sample1001.c`

□ このプログラムは整数の並べ替えを行うものとなっているが、**書き方が悪い**ために他人には理解できない内容である

①まず、仕様書 `spec1001.pdf` を見て、
テストケースを作り、テストを行いなさい
内容は `test-report1001.xlsx`
に書きなさい(5個以上は考えること)

(C) 2007-2024 Hirohisa AMAN

4

整数の並べ替え(せいすうのならべかえ): sorting of integers

書き方が悪い(かきかたがわるい): the way of coding is poor (bad)

他人には理解できない(たにんにはりかいできない): other people cannot comprehend it

<exercise steps>

1) Download `sample1001.c`, `spec1001.pdf`, and `test-report1001.xlsx`.

These files are included in the exercise material kit (zip file).

Then, test `sample1001.c` and write the results in `test-report1001.xlsx`; the specification is described in `spec1001.pdf`.

You should consider more than five test cases.

課題1 (2/2)

②以下の点に注意して `sample1001.c` の
リファクタリングを行いなさい

■ インデント

■ 変数の名前

■ 関数の名前

必要に応じて
コメント文を追加
するのもよい

③そして、自分が用意したテストケースを使って
テストを行い(`test1001.xlsx`に記録), リファ
クタリングした後も正しく動作することを確認
しなさい(これを回帰テストという)

(C) 2007-2024 Hirohisa AMAN

5

<exercise steps>

2) Refactor the source code written in `sample1001.c` while taking care of the indentations, names of variables and names of functions.

You can add some comments when you need.

3) After refactoring, test `sample1001.c` again using the same test cases you described in `test-report1001.xlsx`.

The aim of those tests is to check whether or not your refactoring activity made a fault into the code.

It is referred to as “regression testing.”

リファクタリングした後も正しく動作することを確認(リファクタリングしたあとでもただしくどうさすることをかくにん): check if the program runs correctly after the refactoring

回帰テスト(かいきテスト): regression testing

課題1の提出

- Teams から
sample1001.c と
test-report1001.xlsx
を提出しなさい

もしも2人組でペアプログラミング
を行った場合は、2人とも同じもの
を提出しなさい

[10] Exercise-1 (refactoring)

- 提出〆切: **13:00, May 20**

提出〆切(ていしゅつしめきり) : submission due (deadline)

If you did the exercise in a group of two students, each of the group members should submit the same files.

課題2 (1/2)

【対象プログラム】 [sample1002.c](#)

□ このプログラムは2進数と10進数の変換を行うものとなっているが**書き方が悪い**

①まず、仕様書 [spec1002.pdf](#) を見て、
テストケースを作り、テストを行いなさい
内容は [test-report1002.xlsx](#)
に書きなさい(5個以上は考えること)

(C) 2007-2024 Hirohisa AMAN

7

2進数(2しんすう): binary number

10進数(10しんすう): decimal number

変換(へんかん): conversion

<exercise steps>

1) Download [sample1002.c](#), [spec1002.pdf](#), and [test-report1002.xlsx](#).

These files are included in the exercise material kit (zip file).

Then, test [sample1002.c](#) and write the results in [test-report1002.xlsx](#); the specification is described in [spec1002.pdf](#).

You should consider more than five test cases.

課題2 (2/2)

②以下の点に注意して `sample1002.c` の
リファクタリングを行いなさい

- `main` 関数の内容を関数へ分割する
- 変数名を意味のあるものにする
- ハードコーディングをマクロへ変える

必要に応じて
コメント文を追加
するのよ

③そして、自分が用意したテストケースでテストを行い(`test-report1002.xlsx`に記録), リファクタリングした後も正しく動作することを確認しなさい

(C) 2007-2024 Hirohisa AMAN

8

<exercise steps>

2) Refactor the source code written in `sample1002.c` while taking care of the following points:

- divide the contents of main function into smaller functions;
- change the variable names to meaningful names;
- replace the hard-coded values with macros (labels).

You can add some comments when you need.

3) After refactoring, test `sample1002.c` again using the same test cases you described in `test-report1002.xlsx`.

関数へ分割(かんすうへぶんかつ): divide into small functions

変数名を意味のあるものにする(へんすうめいをいみのあるものにする): change the variable names to meaningful names

ハードコーディングをマクロへ変える: replace the hard-coded values with macros (labels)

課題2の提出

- Teams から
sample1002.c と
test-reprot1002.xlsx
を提出しなさい

もしも2人組でペアプログラミング
を行った場合は、2人とも同じもの
を提出しなさい

[10] Exercise-2 (refactoring)

- 提出〆切: **13:00, May 20**

提出〆切(ていしゅつしめきり) : submission due (deadline)

If you did the exercise in a group of two students, each of the group members must submit the same files.

Survey に答えなさい

- ☐ 今回の演習では1人でやってもよいし、2人組でペアプログラミングをやってもよい
- ☐ みなさんがどちらを選んだのかを知りたいので、最初に Survey に答えてください

[10] Survey (All students MUST answer first)

Answer the survey first.

You can do these exercise by yourself, or in a group of two students (pair programming).

You all must answer the survey to declare which style do you choose.