

Open/Closed Principle

Włodek Bzyl

Version 1.1, December 11, 2017

Kilka odsyłaczy do łatwych przykładów (za prostych na egzamin).

- Jordan Hudgens, [SOLID Development: Open Closed Principle Guide and Example](#)
- Robert C. Martin, [The Open Closed Principle](#)
- Sergii Makagon, [S\[O\]LID - Open/Closed Principle by example](#)

Projekt zespołowy (do 5 programistów)

W repozytoriach należy umieścić:

1. w katalogu głównym plik *README* w którym:
 - a. opisano proponowane rozszerzenie funkcji programu (biblioteki)
 - b. w szczegółach opisano zapachy w kodzie i wykonane refaktoryzacje, które doprowadziły do zamierzonej otwartości w kodzie
 - c. w szczegółach opisano proces za pomocą którego zaimplementowano proponowane rozszerzenie
 - d. plik *.reek.todo* ignorujący zapachy w kodzie nieopisane w książce M. Fowler „Refaktoryzacja”, na przykład `IrresponsibleModule`, `DuplicateMethodCall`, `UncommunicativeVariableName`
2. plik *Gemfile* z listą użytych gemów
3. w katalogu *lib* kod w Ruby
4. w katalogu *spec* testy RSpec; albo w katalogu *test* testy MiniSpec (lub inne, np. TestUnit)



Plik README należy skonwertować do formatu PDF, wydrukować i dostarczyć przed wyznaczonym terminem. Jeśli plik jest w formacie [AsciiDoctor](#), to do konwersji można użyć programu *asciidoctor-pdf* (jest zainstalowany na serwerze Sigma) w taki sposób:

```
asciidoctor-pdf README.adoc
```

Przykładowe programy i propozycje zmian można znaleźć w katalogach [RinR](#) i [:ant:](#) (brak testów).

Przykład kodu do refaktoryzacji:

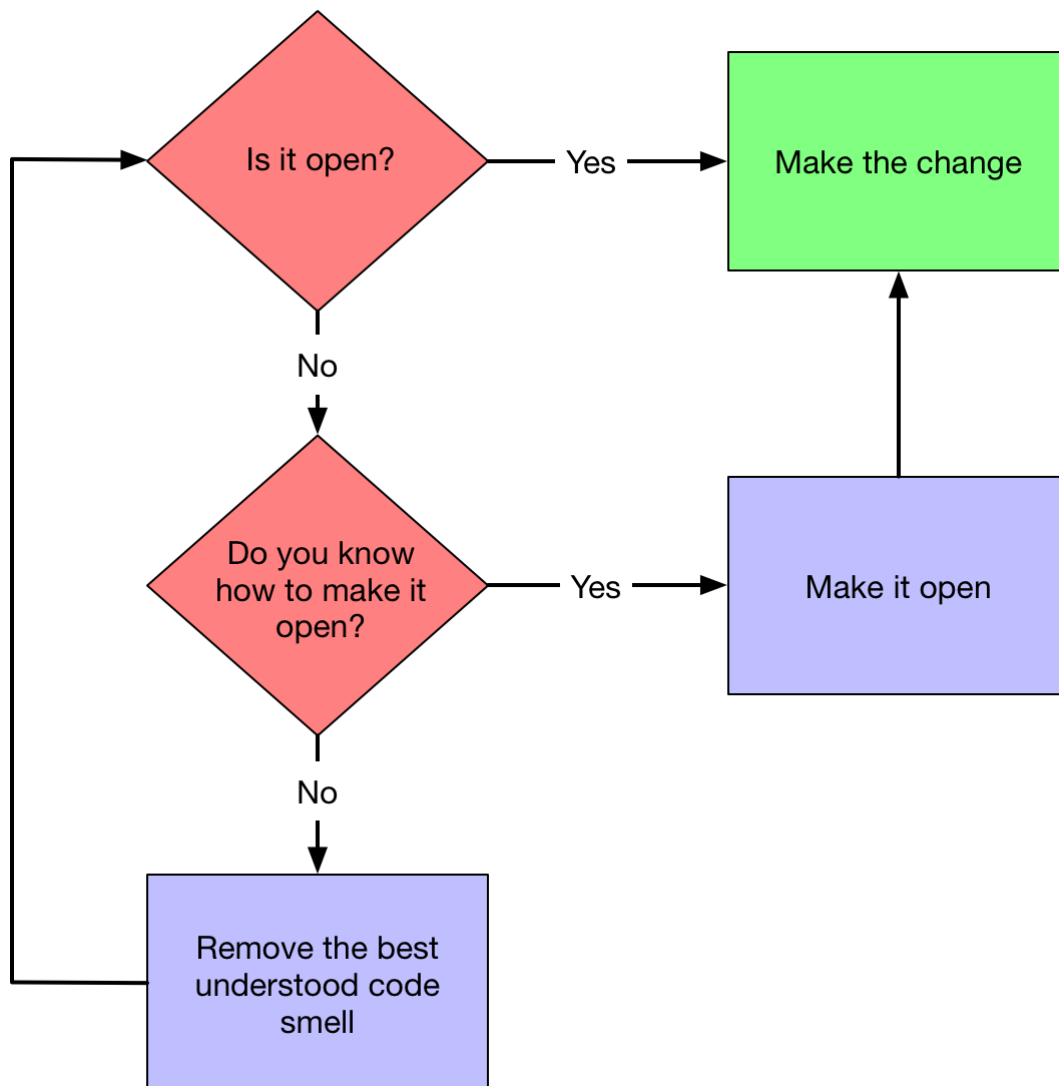
1. [Conway's Game of Life](#); change – użyć biblioteki [Curses](#). Zobacz też [mmoza](#).

W trakcie refaktoryzacji obowiązują następujące reguły:

1. Zmiany w kodzie dotyczą tylko jednego wiersza (instrukcji).
2. Uruchamiamy testy po każdej zmianie.
3. Jeśli jakiś test nie zostanie zaliczony, to wracamy do poprzedniej wersji i wykonujemy inną lepszą zmianę.

Jak to zrobić?

Aby uzyskać zamierzoną otwartość w kodzie kierujemy się tym schematem blokowym.



Source: *99 Bottles of OOP* by S. Metz & K. Owen