

# Zaawansowane Języki Programowania – Zadanie 1

## Gilded Rose – Refactoring

Andrzej Chorostian  
Informatyka 2st. Stacjonarne  
1r. gr.2 indeks: 246750

### 1. Repozytorium i źródła

Moje repozytorium: <https://github.com/AChorostian/GildedRose>

Oryginalny kod zadania: <https://github.com/emilybache/GildedRose-Refactoring-Kata>

### 2. Cel

Oryginalny program symuluje zachowanie pewnych przedmiotów. W każdym dniu zmienia się ich wartość oraz data „ważności” czyli za ile dni należy je sprzedać.

Do kodu należało dodać zachowanie dla nowego przedmiotu „Conjured Mana Cake”. Jego jakość powinna obniżać się dwukrotnie szybciej w porównaniu do innych przedmiotów.

Głównym celem natomiast była refaktoryzacja kodu, czyli sprawienie, aby był czytelniejszy, mniej skomplikowany oraz aby w przyszłości dodawanie nowych zachowań dla przedmiotów było łatwiejsze.

### 3. Refaktoryzacja

Proces refaktoryzacji kodu wyglądał następująco:

#### Ver.0

Pobrałem oryginalny kod w języku Python do repozytorium.

#### Ver.1

Odwrociłem część instrukcji warunkowych

Skróciłem inkrementacje i dekrementacje zmiennych z postaci „a = a + 1” do postaci „a += 1”.

#### Ver.2

Wyciągnąłem zachowanie przedmiotów „Backstage” oraz „Aged Brie” do osobnego rozpatrywania, ponieważ bardziej się różniły od pozostałych przedmiotów.

#### Ver.3

Podłączyłem repozytorium do Narzędzi kontrolujących jakość kodu: Codacy i CodeClimate.

(To nie jest część refaktoryzacji)

#### Ver.4

Wyciągnąłem zwiększanie jakości przedmiotu do osobnej metody, aby nie duplikować w kodzie sprawdzania warunku: jakość mniejsza od 50.

#### Ver.5

Wyciągnąłem zmniejszanie jakości przedmiotu do osobnej metody, aby nie duplikować w kodzie sprawdzania warunku: jakość większa od 0.

#### Ver.6

Dodałem obsługę zachowania dla nowego przedmiotu „Conjured Mana Cake”.

#### Ver.7

Dodałem przerwanie iteracji w pętli po wykryciu i realizacji zachowania dla przedmiotów „Sulfuras”, „Backstage” i „Aged Brie”, aby oszczędzić sprawdzania pozostałych warunków.

#### Ver.8

Wydzieliłem zachowania „Backstage” i „Aged Brie” do osobnych metod, aby skrócić metodę „update\_quality”.

#### Ver.9

Dodałem możliwość podania wartości do metod zmieniających jakość przedmiotu.

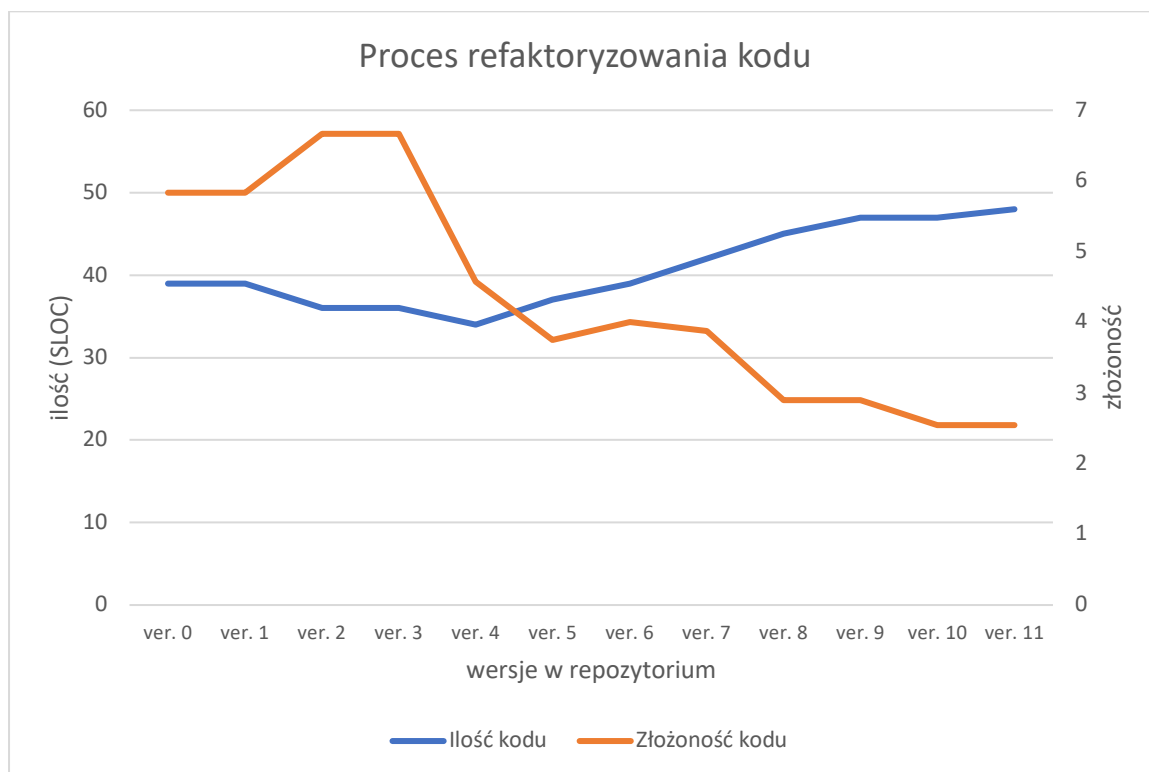
#### Ver.10

Wydzieliłem osobno metodę uaktualniającą przedmiot z wyłączeniem „Sulfuras”.

#### Ver.11

Usunąłem zbędne komentarze i poprawiłem odstępy linii między klasami, metodami i importowanymi modułami.

Oto wykres przedstawiający jak zmieniła się złożoność oraz ilość kodu:



## 4. Rezultat

Finalnie, złożoność kodu zmniejszyła się o 56%, kiedy ilość wzrosła tylko o 23%.

