

# Sztuka Wytwarzania Oprogramowania

## Laboratorium 4

- Laboratorium rozpocznie się od wprowadzenia i prezentacji narzędzi przez Prowadzącego.
  - Po skończonej prezentacji każdy Zespół otrzymuje dostęp do projektu w swojej grupie w systemie GitLab (<https://gitlab-stud.elka.pw.edu.pl/>).
  - W projekcie znajduje się implementacja systemu do zarządzania sprzedażą przedmiotów, gdzie główna funkcjonalność znajduje się w klasie **WhiteBear**. Kod źródłowy symuluje sytuację, w której konieczna jest rozbudowa systemu i przed jego modyfikacją konieczne jest „posprzątanie” kodu.
  - W ramach laboratorium do wykonania jest następujący zestaw zadań:
1. Skompiluj projekt i zaktualizuj odpowiednio plik .gitignore, aby nie komitować do repozytorium zbędnych plików będących wynikiem budowania projektu oraz kompilacji.
  2. Wygeneruj raport podsumowujący złożoność obliczeniową funkcji z wykorzystaniem narzędzia pmccabe. Raport umieść w pliku mccabe-report.txt. Dokumentacja narzędzia znajduje się pod linkiem: <https://people.debian.org/~bame/pmccabe/overview.html>
  3. Popraw strukturę projektu zgodnie z dobrymi praktykami tworzenia oprogramowania tworząc dedykowane katalogi.
  4. Przeprowadź dowolną refaktoryzację metody o największej złożoności obliczeniowej, zmniejszając jej złożoność obliczeniową co najmniej o połowę.
  5. Usuń możliwie jak najwięcej brzydkich zapachów kodu (ang. code smell) takich jak np. martwy kod, itd. Wszelkie usterki powinny być naprawiane w osobnych commitach oraz opisane w odpowiedni sposób, dlaczego usunięte konstrukcje są błędne.
  6. Napisz dedykowane testy dla przedmiotów w sklepie WhiteBear zgodnie z zamieszczoną specyfikacją zapisaną w pliku ExpectedTestOutput.txt. Specyfikacja przedstawia w jaki sposób zmieniają się atrybuty przedmiotów w kolejnych dniach od dostawy produktu do sklepu.

W sklepie znajdują się 4 rodzaje produktów:

- Kamizelka dodająca plus 7 punktów mocy;
- Dojrzewający ser;
- Bilety na koncert
- Zbroja Legolas, Had of Gollum

Każdy produkt ma zdefiniowaną inną ilość dni w ciągu których ma być sprzedany od daty sprowadzenia towaru do sklepu oraz wartość przedmiotu, która może się zmieniać w zależności od tego jak długo produkt jest w sklepie. Format specyfikacji zawiera następujące informacje:

Wiersz podsumowujący indeks dnia oraz w kolejnych wierszach podsumowane są atrybuty produktów znajdujących się w sklepie.

- Kolumna 1 – zawiera nazwę przedmiotu, np. Aged Cheese;
- Kolumna 2 – który określa ilość dni, jakie mamy na sprzedaż przedmiotu;
- Kolumna 3 – określa jak wartościowy jest dany przedmiot.

Np.

Day is 0

+7 Yellow Vest, 9, 19

Aged Cheese, 9, 21

Tickets to a concert, 9, 22

Legolas, Hand of Gollum, 10, 20

Ze specyfikacji wynika, że np. produkt „Aged Cheese” po sprowadzeniu go do sklepu powinien być sprzedany w ciągu 9 dni i jego wartość w dniu sprowadzenia do sklepu wynosi 22.

Projekt i repozytorium zostają na systemie GitLab do wglądu i oceny przez Prowadzącego. W ramach laboratorium nie ma narzuconego sposobu realizacji projektu. Pierwsze 4 zadania są proste i mogą być wykonywane niezależnie. Dwa kolejne zadania są bardziej złożone i mogą być rozwiązane przez studenta w dowolny sposób. Możliwe jest wykorzystanie dodanych w kodzie podpowiedzi, ich modyfikacja lub też całkowite ich usunięcie i napisanie swoich własnych rozwiązań. Finalna wersja kodu powinna zachować jego pierwotną funkcjonalność oraz przedstawiać kod, z którego usunięto możliwie najwięcej niepoprawnych konstrukcji. Pełną liczbę punktów można uzyskać rozwiązując zadania 1-4 oraz usuwając co najmniej dwóch zapachów kodu nie licząc zmniejszenia złożoności obliczeniowej oraz dodając testy jednostkowe dla podanej specyfikacji.

Ocenie podlegają:

- Jakość i trafność komentarzy umieszczanych podczas przeglądu kodu.
- Zastosowanie się do sugerowanych zmian (włączając to odrzucenie niepotrzebnych).
- Komentarze do *commitów*.
- Dokonane zmiany w kodzie oraz uzupełniające do zmian komentarze.