

Recenzja projektu

<https://github.com/Pankrip/nntp>

Artur Jopek, Krystian Krakowski

Kod

Recenzowany przez nas projekt został napisany w języku Haskell, w przemyślany sposób został on podzielony na odpowiednie pliki i foldery. Zawiera on przykładowe testy oraz dokumentację których oceną zajmiemy się później. Pierwszą rzeczą rzucającą się w oczy jest duża ilość modułów, podzielona ze względu na funkcjonalności oraz konteksty ich użycia, które łączą się w spójną całość. Kod jest napisany w sposób przejrzysty z odpowiednimi wcięciami oraz komentarzami, choć ich treść może świadczyć o tym, że autorzy projektu mieli w zamiarze dopracować działanie ich serwera nntp wzbogacając go o metody które nie zostały ujęte w projekcie. W głównym folderze projektu (src) znajdują się katalogi Client, Init, StateMachine oraz conf. Podział bardzo pozytywnie wpływa na czytelność całego projektu, choć pozostało sporo plików w głównym katalogu, które można by jeszcze przenieść do podkatalogów. Pierwszy z podkatalogów zawiera pliki opisujące stan klienta podczas działania programu oraz przechowuje niezbędne informacje dotyczące połączenia. Kolejny z katalogów posiada plik którego zadaniem jest zamiana pliku konfiguracyjnego lub opcje linii poleceń na strukturę typu "Options" wynikającej z zaimportowanej biblioteki o tej samej nazwie. StateMachine jak sama nazwa wskazuje jest odpowiedzialny za maszynę stanów. Ostatni z tych katalogów zawiera plik konfiguracyjny potrzebny podczas działania programu.

W kodzie wszechobecny jest wzorzec collection pipeline, co znacznie zwiększa przejrzystość i czytelność. Styl wcięć i ogólnego formatowania kodu jest czytelny i spójny przez cały projekt. Moduły są jasno podzielone na kolejne, podpisane sekcje, grupujące funkcje według ich przeznaczenia. Widać przemyślaną kolejność umieszczania funkcji, funkcje pomocnicze umieszczane są na końcach modułów, a funkcje mające większe znaczenie bliżej początku. Same funkcje, jak i typy w sygnaturach są okomentowane, co bardzo pomaga w szybkim zrozumieniu działania tych funkcji.

Repo

Autorzy w sposób sprawny posługiwali się githubem, wrzucając wiele commitów dotyczących projektu. Na szczególną uwagę zasługuje codzienna systematyczność w ostatnim tygodniu o czym mogą świadczyć statystyki aktywności znajdujące się na githubie. Na ich podstawie możemy wysunąć wnioski, że praca do wykonania przy projekcie była równo podzielona między autorów. Commit messages, niosą pewne informacje o treści commitów, oraz zachowana jest konsekwencja co do stylu ich pisania. Mogłyby jednak być trochę dłuższe i nieść więcej informacji co do treści (np. "add commands" - jakie komendy, z jakim przeznaczeniem). Autorzy zamieścili też plik readme, niosący pewne informacje o

treści i odnośniki do informacji pomagających zrozumieć istotę stworzonego projektu, co jest dużym plusem.

Testy

W projekcie możemy znaleźć wiele testów parametrycznych oraz jednostkowych, sprawdzające działanie serwera nntp oraz poszczególnych jego funkcji. Podczas testowania są wykorzystywane różne biblioteki takie jak Hspec oraz QuickCheck, dzięki którym jest przeprowadzona dokładna analiza działającego programu pod względem możliwych błędów. W katalogu testów również występuje podział na katalogi, odpowiadający ich zawartości, analogiczny do podziału w src. Ułatwia to znalezienie testów do odpowiadającego im modułu. Funkcje pomocnicze zostały wyodrębnione do osobnego modułu (Test.Helpers), dzięki temu nie zaciemniają one modułów z testami.

Dokumentacja

Dokumentacja generowana z komentarzy, jest przejrzysta, zwięźle opisuje przeznaczenie funkcji. Pojawiają się nagłówki opisujące cały moduł, nie są one jednak standardem. Brakuje także w readme, lub dokumentacji jasnego opisu dla potencjalnego użytkownika projektu, w jaki sposób projekt uruchomić i użytkować.