

Podstawowy warsztat informatyka — lista 11

Zadanie 1 (2 punkty). To zadanie nie wymaga wcześniejszej znajomości języka Prolog ani algorytmu quicksort. Wystarczy wiedzieć, że na pracowniach jest zainstalowany interpreter Prologa, który można odpalić poleceniem `swipl`, a kod w Prologu to ciąg klauzul. Klauzule są typowo postaci:

`wniosek :- przeslanka1, ..., przeslankan.`

Taka linia odpowiada implikacji $przeslanka1 \wedge \dots \wedge przeslankan \Rightarrow wniosek$.

Ściągnij repozytorium <https://github.com/piotrekiiuwr/quicksort17/>. Reprezentuje ono historię pewnej wadliwej implementacji algorytmu quicksort w języku Prolog. Twoim zadaniem będzie poprawić tę implementację. W tym celu:

1. Przejrzyj historię commitów. Znajdź (na podstawie opisów) taki, który jest jasnym oszustwem polegającym na usunięciu testu, który nie działa.
2. Ściągnij wcześniejszą wersję i sprawdź, czy rzeczywiście odpalenie programu z testami (poleceniem opisanym w README.md) zwraca informację, że jeden z testów nie jest spełniony.
Uwaga: Ten test jest zrandomizowany i z małym prawdopodobieństwem może być spełniony, więc lepiej uruchomić go dwa razy. Stąd wziął się cały problem – w czasie edycji, która wprowadziła błąd, akurat przypadkiem test przeszedł.
3. Pobieraj coraz to wcześniejsze wersje tego programu aż dojdiesz do takiej, gdzie oba testy są spełnione. Wywnioskuj (na podstawie opisu kolejnego commita), co zostało popsute i jak to naprawić.
4. Ściągnij jeszcze raz najnowszy commit i napraw program. Przywróć wcześniejszą wersję pliku z testami aby upewnić się, że wszystko działa.

Zadanie 2 (2 punkty). (Kontynuacja zadania 1 z listy 10)

Sprawdź czy wykonałeś zadanie 1 z listy 10 poprawnie — jeżeli nie to usuń niepotrzebne pliki i dodaj te brakujące wg poleceń z poprzedniej listy (np. usuń plik `o-mnie.txt` z gałęzi `master`, czy plik `o-komputerze.txt` z gałęzi `ja`). Jeśli były konieczne poprawki to je skomituuj.

Przejdź do gałęzi `ja` i dodaj tam plik `ulubiony-film` zawierający nazwę Twojego ulubionego filmu, a następnie skomituuj te zmiany.

Ojej, zapomniałem napisać, że w komicie dotyczącym miasta w Afryce trzeba było dodać jeszcze liczbę ludności tego miasta! Trzeba to natychmiast poprawić. Wykorzystaj polecenia `git rebase` oraz `git commit --amend`, aby to zrobić. Następnie poukładaj komity tak, jak były.

Czas połączyć trzy gałęzie w jedną. Przejdź do gałęzi `master` i dodaj do niej komity dotyczące pliku `o-mnie.txt` z gałęzi `ja` poleceniem `git cherry-pick`. Następnie dołącz gałąź `komputer` poleceniem `git merge`.

Przejdź do komita (nie gałęzi!), w którym stworzyłeś plik `ulubiony-film.txt`. Dopisz do niego rok wydania tego filmu, a następnie skomituuj zmiany. Dodaj tag do tego komita. Przejdź do gałęzi `master` i zrób `merge` z otagowanym komitem.

Wyślij zmiany do repozytorium zdalnego.

Zadanie 3 (6* punktów, zadanie dla chętnych). Stwórz swoje cv w L^AT_EXu, pracując w repozytorium założonym pod adresem <https://classroom.github.com/a/2i10tWvs>. Powinno mieć długość 1 lub 2 stron i spełniać wszystkie standardy (dla przykładu, powinno zawierać klauzulę o zgodzie na przetwarzanie danych). Rozwiązania należy przesłać najpóźniej 7 stycznia 2020 o 8.00. Nie musisz podawać swoich prawdziwych danych osobowych.