

PIISW, W08, IO, 2018/2019, semestr letni

Lista zadań nr 1

mgr inż. Maciej Małecki
maciej.malecki@pwr.edu.pl

mgr inż. Marcin Stachowiak
marcin.stachowiak@pwr.edu.pl

26 lutego 2019

Wprowadzenie

Do realizacji zadań z tej listy niezbędne jest wykonanie zadań z listy 0, w szczególności zaś konieczne jest posiadanie konta edukacyjnego na portalach `github.com` oraz `travis-ci.com`.

Rozwiązanie każdego z zadań musi znaleźć się na prywatnym repozytorium studenta (repozytorium takie tworzone jest w ramach zadania 2).

Oceny

Punkty:	< 8	8 – 9	10 – 11	12 – 13	14 – 15	16 – 17
Ocena:	2,0	3,0	3,5	4,0	4,5	5,0

Zadania

- (2 pkt) Konfiguracja dostępu do serwisu `github.com`.
 - Wygeneruj parę kluczy publicznych/prywatnych o długości 4096 bitów (z użyciem narzędzia `ssh-keygen` lub `putty-keygen`¹).
 - Zarejestruj wygenerowany klucz publiczny w swoim profilu w serwisie `github.com`.
 - Skonfiguruj lokalnego klienta GIT tak, aby dla domeny `github.com` użyty był wygenerowany klucz prywatny².
- (2 pkt) Praca z lokalną kopią repozytorium.
 - Utwórz prywatne repozytorium w ramach organizacji `pwr-piisw`, nazwa repozytorium powinna być w następującej formie: `imie_nazwisko`.
 - Wykorzystując funkcjonalność portalu `github` zaimportuj zawartość repozytorium `https://github.com/pwr-piisw/oasp4js-ng-boot-project-seed.git` do swojego repozytorium prywatnego³.
 - Nadaj uprawnienia (zapis/odczyt) do repozytorium prowadzącemu.

¹Klucz w formacie `putty` należy przekonwertować do formatu OpenSSH.

²Skorzystaj z informacji zawartych w `https://pwr-piisw.github.io/materialy/GIT%20Cheat%20Sheet.pdf`.

³Możliwe jest także użycie `git remote add` oraz `git push` aby zaimportować zawartość repozytorium w przypadku, gdy repozytorium prywatne już istnieje i jest puste.

- (d) Sklonuj zawartość repozytorium z użyciem lokalnego klienta GIT.
 - (e) Zmodyfikuj zawartość pliku `README.adoc` dodając nowy paragraf tekstu.
 - (f) Zatwierdź (ang. *commit*) i skomentuj zmianę, przenieś zmianę (ang. *push*) do repozytorium zdalnego.
3. (3 pkt) Praca z gałęziami oraz *pull request*'ami.
- (a) Utwórz *branch* oraz *wycheckout*'uj go. Wprowadź zmianę w kodzie na branchu i zatwierdź ją. Wypushuj branch.
 - (b) Przejdź na *branch master*. Wprowadź inną zmianę, zatwierdź ją oraz wypushuj.
 - (c) W serwisie `github.com` stwórz *pull request* dla brancha utworzonego w punkcie 3a. W przypadku braku konfliktów, `github` zaoferuje trzy możliwości scalenia zmian. Wybierz opcję: *Rebase and merge*.
4. (4 pkt) Porządkowanie historii.
- (a) Utwórz kolejny *branch* oraz *wycheckout*'uj go. Dokonaj na tym *branchu* trzy zmiany każdą niezależnie *commitując*. Wypushuj branch na zdalne repozytorium.
 - (b) Wróć na gałąź *master* oraz przenieś zmiany z gałęzi utworzonej w punkcie 4a w taki sposób, aby były one widoczne jako jeden commit nie będący merge-commitem. Wypushuj zawartość brancha *master*, pozostaw także *commity* na *branchu* z punktu 4a.
- Wskazówka:** zadanie można wykonać na kilka sposobów, warto poszukać w sieci informacji na temat techniki *git squash*.
5. (2 pkt) Konfiguracja środowiska *continuous integration*.
- (a) Utwórz konto w serwisie `travis-ci.com` oraz skojarz je ze swoim kontem edukacyjnym w serwisie `github.com`.
 - (b) Aktywuj środowisko CI dla repozytorium utworzonego w zadaniu 2, uruchom ręcznie pierwszy *build*.
 - (c) Wprowadź kolejną zmianę do pliku `README.adoc`, zatwierdź i wypushuj, sprawdź czy kolejny *build* uruchamia się automatycznie.
 - (d) Zaktualizuj link do grafiki reprezentującej status *builda* tak, aby wskazywał na właściwe środowisko w ramach `travis-ci`.
6. (4 pkt) W systemie obsługującym bibliotekę istnieją trzy komponenty: komponent zarządzający użytkownikami, komponent odpowiedzialny za książki oraz komponent w którym znajduje się logika i reguły biznesowe procesów wypożyczania książek. Komponent wypożyczania zależy od komponentu obsługi książek, a ten z kolei zależy od komponentu obsługi użytkownika.
- (a) Utwórz podkatalog w repozytorium, pozostałe punkty tego zadania należy wykonać w tym katalogu lub jego podkatalogach.
 - (b) Przygotuj komendy Maven'owe po których uruchomieniu powstanie trójmodułowy projekt zgodny z ww. zależnościami.
 - (c) Główny plik `pom.xml` powinien znajdować się w katalogu 6a, a każdy z modułów powinien mieć swój podkatalog.