wydział elektrotechniki elektroniki informatyki i automatyki

METODY TESTOWANIA OPROGRAMOWANIA

LABORATORIUM 0

Praca z systemem kontroli wersji Git oraz platformą GitHub

Reguły formatowania i pisania kodu

wersja 1.0

przygotował:
dr inż. Radosław Adamus

Efekty:

Po ukończeniu laboratorium będziesz:
1. Potrafił wykonywać podstawowe operacje z wykorzystaniem systemu Git i platformy GitHub wymagane do pracy w ramach laboratorium przedmiotu MTO.
2. Potrafił przesyłać rezultaty zadań za pomocą operacji pull request.
3. Znał zasady formatowania kodu źródłowego oraz stosowane konwencje i praktyki programistyczne.
4. Potrafił konfigurować IDE dla potrzeb automatycznego formatowania kodu.

Wymagania wstępne:

1. Posiadanie konta na platformie Github.

Narzędzia:

1. Git
2. GitHub
3. Eclipse IDE (lub inne)

Opis:

Celem niniejszego laboratorium jest zapoznanie się z procesem pracy z systemem Git dla potrzeb kolejnych laboratoriów oraz z regułami formatowania kodu w Java. W ramach laboratorium należy przećwiczyć cały proces pracy z kodem źródłowym i repozytorium Git wykorzystując przykładowe repozytorium udostępnionego iomówionego przez prowadzącego.

Uwaga:

Proces poznany w trakcie niniejszego laboratorium będzie wymagany do pracy w kolejnych laboratoriach. Aby można było je wykonać wymagane jest ukończenie laboratorium 0.

1. Rozpoczęcie zajęć (zadanie wykonywane jednorazowo w ramach semestru)

Aby przystąpić do laboratorium każdy student musi posiadać konto na platformie GitHub. Wypełniając formularz zapisu na zajęcia w polu ***Uwagi*** należy podać nazwę użytkownika zarejestrowaną na platformie GitHub.
Prowadzący zajęcia wyśle, z poziomu platformy, zaproszenie do użytkownika po podanej nazwie do tzw. Organizacji, która będzie reprezentowała zajęcia (np. mto-2017). Student, aby móc przystąpić do zajęć musi zaakceptować zaproszenie (wysyłane jest ono domyślnie na adres mailowy powiązany z kontem GitHub).

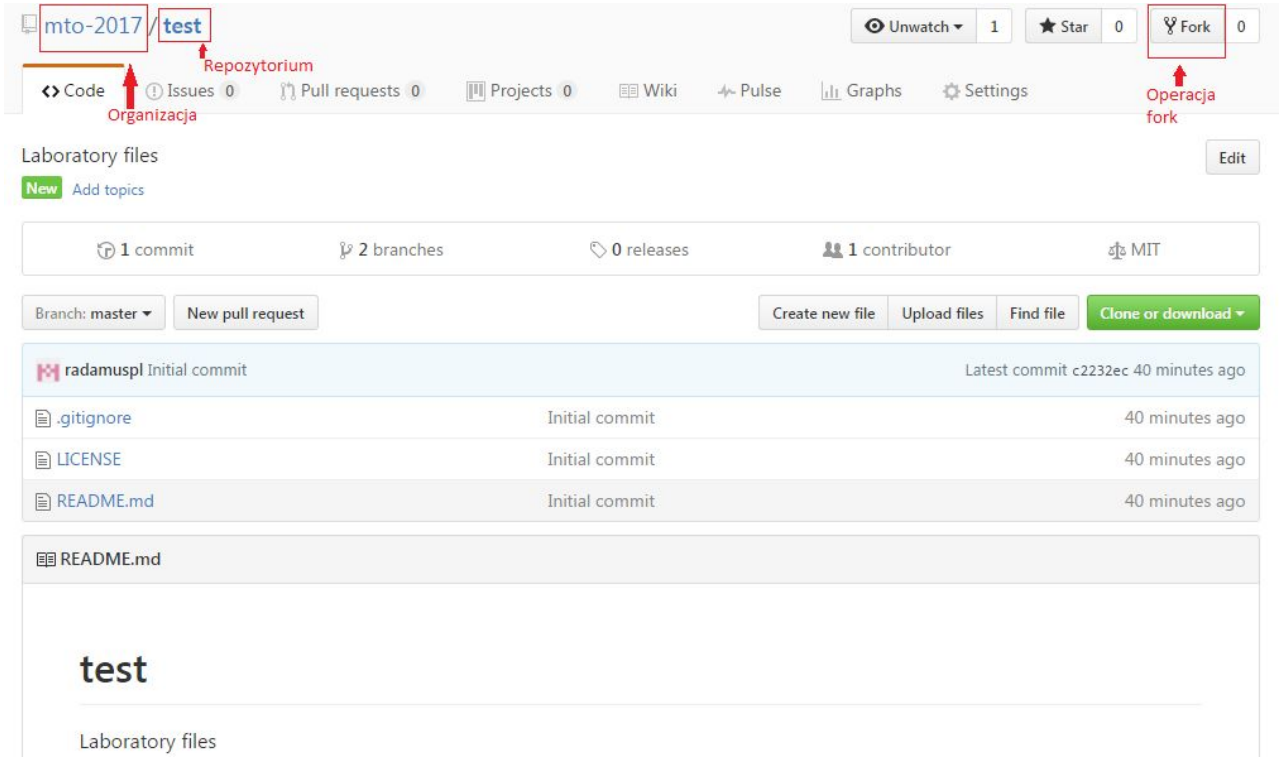
Należy zapoznać się z regułami formatowania kodu oraz konwencjami i praktykami

programistycznymi w języku Java opisanymi w dokumencie Java-rules oraz sposobem konfiguracji środowiska Eclipse IDE pozwalającego na automatyzację reguł formatowania (Automated Code Formatting).

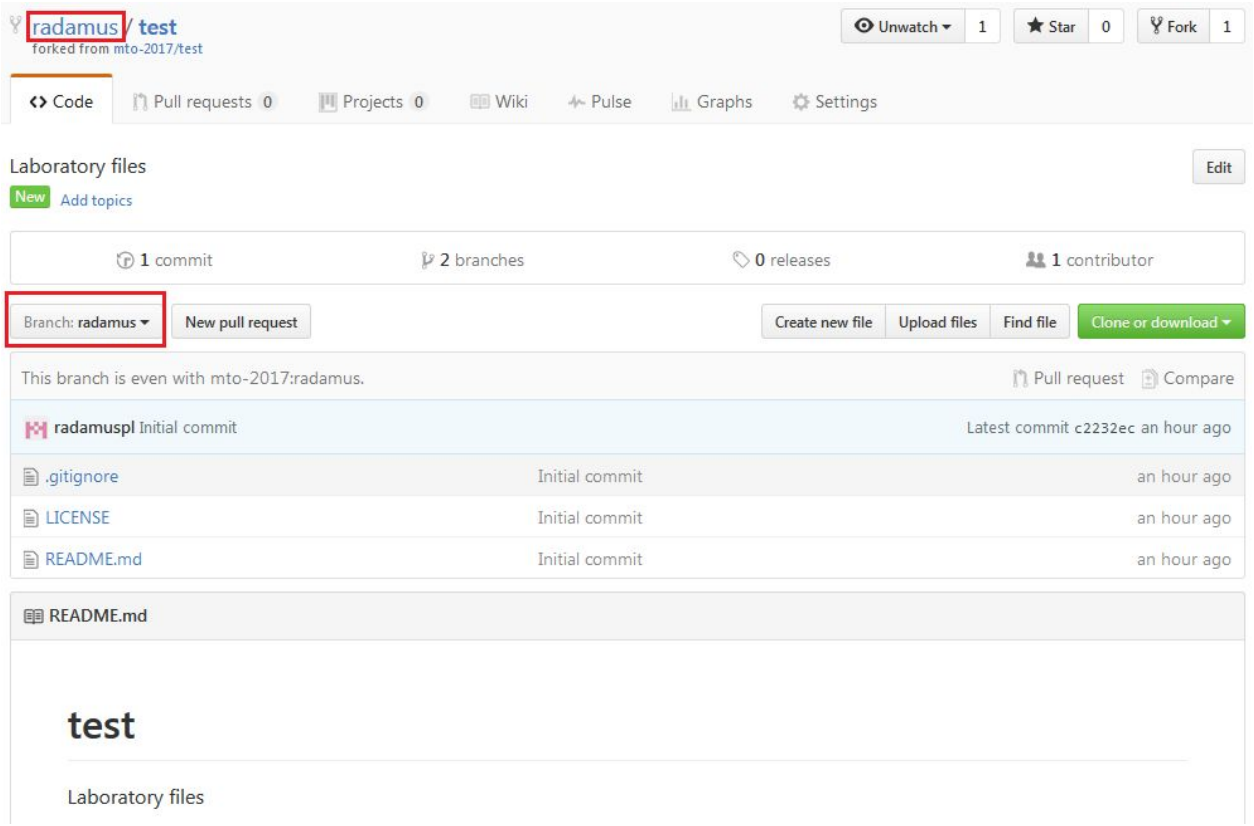
2. Przygotowanie do wykonania zadania

Zadanie laboratoryjne, które oceniane będzie na podstawie analizy kodu źródłowego utworzone zostanie repozytorium (w obrębie Organizacji).
Prowadzący zajęcia utworzy gałęzie dla studentów (jedną lub więcej w zależności od wymagań zadania).

Każdy student klonuje repozytorium na swoje konto Github (operacja fork na platformie GitHub).



Rysunek1: Oryginalne repozytorium



Rysunek 2: Repozytorium na koncie studenta po wykonaniu operacji fork

3. Rozpoczęcie pracy

Repozytorium reprezentujące zadanie laboratoryjne należy sklonować (ze swojego konta Github) na lokalny komputer w celu wykonania zadań. W konsoli operacja:

```
git clone adres_repozytorium
```

Uwaga! Należy ustawić w konfiguracji systemu Git na komputerze lokalnym aby zatwierdzane zmiany sygnowane były nazwą i adresem email reprezentującym studenta (z poziomu folderu projektu):

```
git config user.name "nazwisko.imie"
git config user.email "nrindeksu@edu.p.lodz.pl"
```

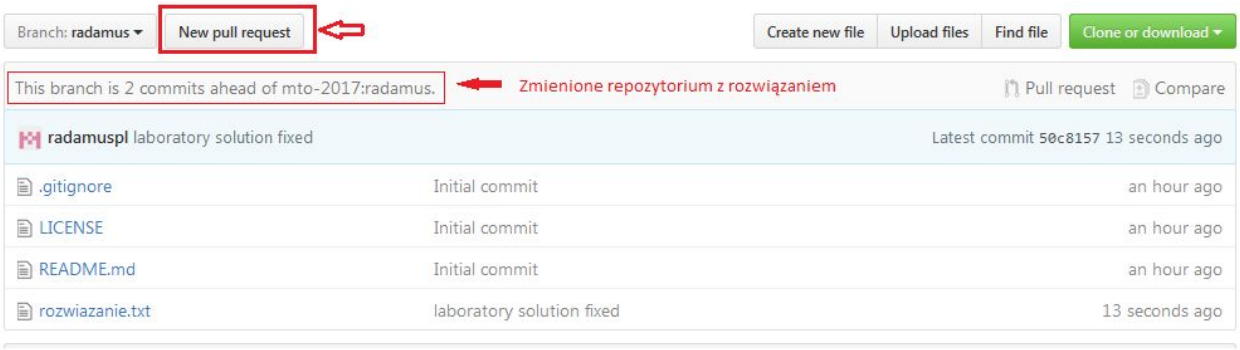
W celu wykonania zadań laboratoryjnych należy zaimportować projekt do edytora IDE (np. Eclipse).

4. Praca w trakcie laboratorium

Zmiany należy zatwierdzać w odpowiednich gałęziach repozytorium.
W trakcie wykonywania zadań należy zatwierdzać pośrednie efekty prac w repozytorium. Nie jest możliwe zaliczenie laboratorium jeżeli całość pracy będzie zatwierdzona pojedynczą zmianą na koniec laboratorium.
Ostateczne efekty pracy należy wypchnąć do repozytorium na swoim koncie GitHub.

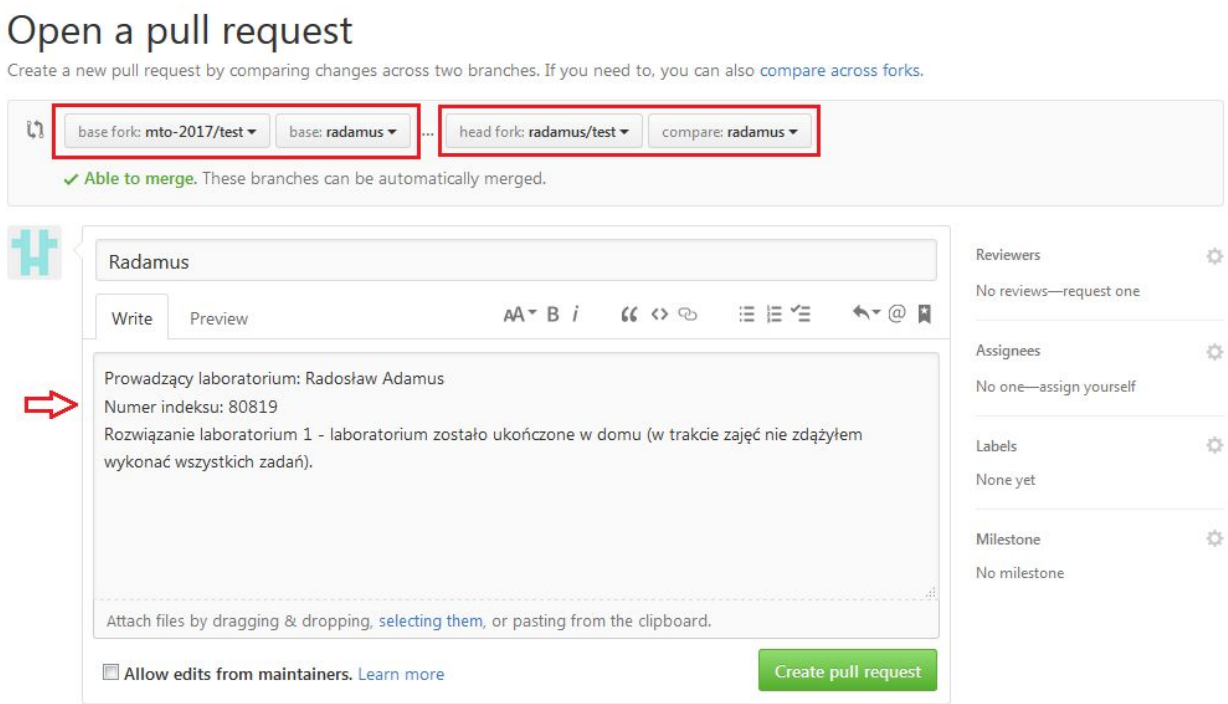
Zaliczenie laboratorium

Przesłanie pracy do sprawdzenia odbywa się **wyłącznie(!)** za pomocą operacji `pull request`: **Nie należy** przesyłać zmian bezpośrednio do repozytorium oryginalnego.



Rysunek 3: Operacja pull request w interfejsie repozytorium GitHub

Jako bazę operacji należy wybrać repozytorium oryginalne (base fork) i gałąź studenta (base). Jako wersję do złączenia należy wybrać repozytorium na koncie studenta (head fork) i gałąź do której zatwierdzane były zmiany (compare). Komentarz do operacji pull request można wykorzystać do przesłania dodatkowych informacji ułatwiających prowadzącemu sprawdzenie pracy. W celu uniknięcia pomyłek należy przynajmniej wpisać swój numer indeksu oraz nazwisko prowadzącego laboratorium.



Rysunek 4. Edycja pull request'a.