

Zarządzanie projektami i usługami

Instrukcja do laboratorium 2

Bartłomiej Kizielewicz

27 listopada 2022

1 Zasady ogólne

Ta instrukcja jest przeznaczona na 6 godzin lekcyjnych. Do wykonania są 3 zadania. Ocena za laboratorium zależy od liczby dobrze zrobionych zadań, skala ocen jest pokazana w Tabeli 1.

Tabela 1: Skala ocen.	
Liczba zrobionych zadań domowych	Ocena
1	3.0
2	4.0
3	5.0

W przypadku nie oddania zadania w terminie ustawionym w systemie Moodle, uzyskana ocena jest zmniejszana o 0,5 za każdy zaczęty tydzień opóźnienia. Jako termin oddania liczy się data wgrania pliku z kodem źródłowym na platformie Moodle.

2 Zakres tematyczny

- Rozwiązywanie konfliktów,
 - <https://git-scm.com/book/pl/v2/Ga%C5%82%C4%99zie-Gita-Podstawy-rozga%C5%82%C4%99ziania-i-scalania>
- Podgląd historii,
 - <https://git-scm.com/book/pl/v2/Podstawy-Gita-Podgl%C4%85d-historii-rewizji>
- Cofanie zmian,
 - <https://git-scm.com/book/pl/v2/Podstawy-Gita-Cofanie-zmian>

3 Zadania do wykonania

1. Zadanie pierwsze:

- Utwórz na pulpicie katalog zadanie_drugie (konsola Git)
cd /Desktop
mkdir zadanie_drugie
cd zadanie_drugie
- Utwórz repozytorium (git init)
- Stwórz plik index.html z następującą zawartością:

```
1 <!doctype html>
2 <html>
3   <head>
4     <meta charset="UTF-8" />
5     <title>Strona przykładowa</title>
6   </head>
7   <body>
8     </body>
9 </html>
```

- Dodaj plik do repozytorium (git add, git commit) - po 'git commit -m' o treści 'pierwsza modyfikacja'
- Stwórz branch **development**
- Na branchy **development** edytuj plik index.html w następujący sposób:

```
1 ...
2     <title>Zmieniona nazwa strony</title>
3 ...
```

- Dodaj i zatwierdź zmiany (git add, git commit)
- Przełącz się na branch **master**
- Edytuj plik index.html w następujący sposób:

```
1 ...
2     <title>Poprawna nazwa strony</title>
3 ...
```

- Dodaj i zatwierdź zmiany (git add, git commit)
- Branch **master** zmerguj z branchą **development** (git merge)
- Rozwiąż konflikt występujący w pliku index.html, akceptując zmiany pochodzące z branchy **master**
- Dodaj i zatwierdź zmiany (git add, git commit)
- Usuń branch **development**
- Dodaj repozytorium do swojego konta na serwisie GitHub

2. Zadanie drugie:

- W tym zadaniu będziemy pracować na repozytorium stworzonym w zadaniu pierwszym
- Utwórz branch **fixes**
- Na branchy **fixes** zedytuj plik index.html w następujący sposób:

```
1 ...
2     <title>Nowa nazwa strony</title>
3     <p>Losowy tekst</p>
4 ...
```

- Dodaj i zatwierdź zmiany (git add, git commit)
- Przełącz się na branch **master**
- Usuń plik index.html (rm -rf index.html)
- Dodaj i zatwierdź zmiany (git add, git commit)
- Branch **master** zmerguj z branchą **development** (git merge)
- Zachowaj zmiany z usunięciem pliku (git rm, git commit)
- Dodaj zmiany do repozytorium na GitHub

3. Zadanie trzecie:

- W tym zadaniu będziemy pracować na repozytorium stworzonym w zadaniu drugim
- Utwórz plik README.md
- Dodaj i zatwierdź zmiany (git add, git commit) - treść commitu: 'Dodanie pliku README.md'
- Utwórz plik LICENCE.txt
- Dodaj i zatwierdź zmiany (git add, git commit) - treść commitu: 'Dodanie pliku LICENCE.txt'
- Zobacz historię repozytorium (git log --oneline)
- Zmień treść ostatniego commitu (git commit --amend -m "") - treść commitu: "Dodanie licencji"
- Przejdź na commit o treści 'Dodanie pliku README.md' (git reset hash_commitu)
- Zobacz historię repozytorium (git log --oneline)
- Zobacz status repozytorium (git status)
- Dodaj i zatwierdź zmiany (git add, git commit) - treść commitu: 'Dodanie pliku LICENCE.txt'
- Usuń wszystkie commity i pliki aż do commitu z treścią 'Dodanie pliku README.md' (git reset --hard hash_commitu)
- Zobacz historię repozytorium (git log --oneline)
- Zobacz historię zmian w HEAD repozytorium (git reflog)
- Przywróć usunięte commity (git reset --hard hash_commitu)