

Zadanie 1 - założenie konta na GitHub

Załącz konto na platformie GitHub - wykorzystaj przy tym szkolny adres email. Jeśli miałeś już konto założone wcześniej - dodaj adres szkolny do istniejącego konta w ustawieniach.

W domu warto wygenerować klucz SSH i wykorzystywać go podczas uwierzytelnienia.

<https://docs.github.com/en/authentication/connecting-to-github-with-ssh>

Zadanie 2 - konfiguracja Git'a

1. Przy pierwszym użyciu może zaistnieć potrzeba konfiguracji Git'a.
2. Minimalny zestaw ustawień, który warto skonfigurować to nazwa użytkownika i email.
 1. `git config --global user.name "sxxxx"`
 2. `git config --global user.email "sxxxx@p.jwstk.edu.pl"`

Zadanie 2 - nowe repozytorium

1. Stwórz nowe repozytorium na GitHub.
2. Sklonuj powstałe repozytorium na pulpit.
3. Umieść odpowiedni plik gitignore w repozytorium. W naszym wypadku można wybrać jeden z predefiniowanych plików dla Visual Studio.
4. <https://github.com/github/gitignore>
5. Następnie dodaj do repozytorium nową aplikację konsolową .NET. Wykonaj commit o nazwie "Initial project" i wykonaj push do repozytorium online.
6. Następnie wykonaj 3 kolejne commit'y. W każdym commit'cie wprowadź modyfikację w kodzie. Dla każdego z commit'ów

podawaj nazwę "Modyfikacja 1", "Modyfikacja 2", "Modyfikacja 3".

7. Wszystkie commit'y powinny być widoczne online na GitHub.

Możesz również najpierw stworzyć repozytorium lokalnie, a później podpiąć je do repozytorium na GitHub.

Zadanie 3 - nowe zadanie

Założmy, że otrzymałeś nowe zadanie. Musisz stworzyć statyczną metodę, która przyjmuje tablicę int'ów i zwraca wyliczoną średnią.

1. Stwórz osobny branch o nazwie "feature-average".
2. Następnie umieść na tym branch'u commit'y, które implementują wymagania. Możesz umieścić jeden lub dowolną większą liczbą commit'ów.
3. Następnie zmerguj powstały brach z główną gałęzią main. Jaki domyślnie rodzaj merge'a zostanie wykonany przez git'a.
4. Sprawdź jak wygląda obecnie historia repozytorium poprzez komendę:

```
git log --oneline --graph
```

5. Zakładając, że branch "main" nie miał żadnych dodatkowych commit'ów - git powinien skorzystać z metody "fast-forward".

Zadanie 4 - rebase

W kolejnym zadanie mamy dodać statyczną metodę, która przyjmuje tablicę int'ów i zwraca maksymalną wartość.

1. Stwórz nowy branch feature-max
2. Następnie zaimplementuj opisaną funkcjonalność dodając commit'y na branch
3. Na koniec wykonaj merge swojej gałęzi do gałęzi main. Tym razem spróbuj wykonać merge wykorzystując komendę rebase -t

(warto skorzystać z flagi "interactive").

4. Z pomocą komendy git log sprawdź jak wygląda historia repozytorium
5. Wszystkie zmiany powinny zostać wypchnięte do repozytorium online

Zadanie 5 - konflikt

W tym zadaniu zasymulujemy powstanie konfliktu.

1. Stwórz nową gałąź feature-new
2. Następnie będąc na nowo powstałej gałęzi spróbuj zmodyfikować pętle pętli odpowiadającą za wyliczenie średniej. Możesz np. zmienić nazwę zmiennej wykorzystywanej w ramach pętli.
3. Następnie wykonaj commit na gałęzi feature-new.
4. W kolejnym kroku przełącz się na gałąź main i wykonaj inną modyfikację tej samej pętli. Możesz np. zmienić nazwę zmiennej na jeszcze inną metodę.
5. Wykonaj commit na gałęzi main.
6. W ten sposób obie gałęzie różnią się między sobą. Dodatkowo modyfikowaliśmy ten sam kod na obu gałęziach. Taka sytuacja powinna doprowadzić do konfliktu.
7. Spróbuj wykonać merge swojej gałęzi z gałęzią main. Rozwiąż konflikt. Wykonaj push zmian na GitHub.
8. Na koniec sprawdź historię swojego repozytorium poprzez komendę git log.