

Menedżerska Akademia Nauk Stosowanych w Warszawie

Imię i nazwisko studenta: Mateusz Plata

Imię i nazwisko prowadzącego: Jakub Słowik

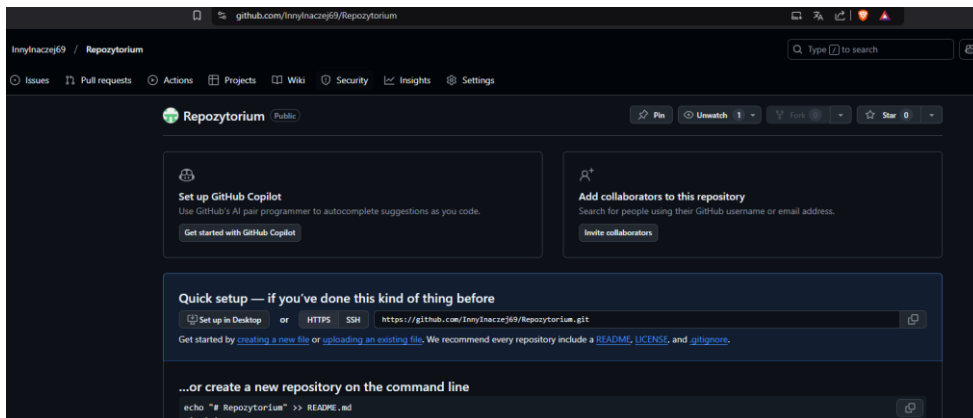
Nazwa przedmiotu: Zaawansowane techniki tworzenia

oprogramowania

Numer grupy: 55DP

Data: 31.01.2025

1. Stworzenie repozytorium



2. Aby wykonać to zadanie, najpierw zainicjowałem lokalne repozytorium przy pomocy komendy `git init`. Następnie stworzyłem plik `README.md` za pomocą polecenia `echo "Pierwszy plik" > README.md` i dodałem go do systemu kontroli wersji przy użyciu `git add README.md`. Kolejnym krokiem było zapisanie zmian w lokalnym repozytorium poprzez wykonanie pierwszego commita komendą `git commit -m "Initial commit"`.

Ustawiłem nazwę głównej gałęzi na `main`, korzystając z komendy `git branch -M main`, co jest standardową praktyką w nowych repozytoriach. Aby połączyć lokalne repozytorium ze zdalnym, dodałem zdalne repozytorium za pomocą `git remote add origin https://github.com/InnyInaczej69/Repozytorium.git`. Następnie wypchnąłem zmiany do zdalnego repozytorium za pomocą `git push -u origin main`, co wymagało zalogowania się w przeglądarce. Po wykonaniu tej operacji lokalna gałąź `main` została powiązana ze zdalną gałęzią `origin/main`.

W ten sposób zakończyłem konfigurację lokalnego i zdalnego repozytorium, zapewniając pełną synchronizację między nimi

```
innyi@Mateusz MINGW64 ~ (main)
$ git init
Reinitialized existing Git repository in C:/Users/innyi/.git/

innyi@Mateusz MINGW64 ~ (main)
$ echo "Pierwszy plik" > README.md

innyi@Mateusz MINGW64 ~ (main)
$ git add README.md
warning: in the working copy of 'README.md', LF will be replaced by CRLF the next time Git touches it

innyi@Mateusz MINGW64 ~ (main)
$ git commit -m "Initial commit"
[main (root-commit) 153eal6] Initial commit
1 file changed, 1 insertion(+)
create mode 100644 README.md

innyi@Mateusz MINGW64 ~ (main)
$ git branch -M main

innyi@Mateusz MINGW64 ~ (main)
$ git remote add origin https://github.com/InnyInaczej69/Repozytorium
error: remote origin already exists.

innyi@Mateusz MINGW64 ~ (main)
$ git push -u origin main
info: please complete authentication in your browser...
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 234 bytes | 234.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/InnyInaczej69/Repozytorium
 * [new branch]    main -> main
branch 'main' set up to track 'origin/main'.

innyi@Mateusz MINGW64 ~ (main)
$
```

```

innymi@Mateusz MINGW64 ~ (main)
$ git checkout main
Already on 'main'
Your branch is up to date with 'origin/main'.

innymi@Mateusz MINGW64 ~ (main)
$ echo "Linia z main" > plik.txt

innymi@Mateusz MINGW64 ~ (main)
$ git add plik.txt
warning: in the working copy of 'plik.txt', LF will be replaced by CRLF the next time Git touches
it

innymi@Mateusz MINGW64 ~ (main)
$ git commit -m "Dodano plik.txt z linią z main"
[main 4a8f594] Dodano plik.txt z linią z main
1 file changed, 1 insertion(+)
create mode 100644 plik.txt

innymi@Mateusz MINGW64 ~ (main)
$ git checkout -b branch1
Switched to a new branch 'branch1'

innymi@Mateusz MINGW64 ~ (branch1)
$ echo "Linia z branch1" > plik.txt

innymi@Mateusz MINGW64 ~ (branch1)
$ git add plik.txt
warning: in the working copy of 'plik.txt', LF will be replaced by CRLF the next time Git touches
it

innymi@Mateusz MINGW64 ~ (branch1)
$ git commit -m "Zmiana w pliku.txt na branch1"
[branch1 d149fe7] Zmiana w pliku.txt na branch1
1 file changed, 1 insertion(+), 1 deletion(-)

innymi@Mateusz MINGW64 ~ (branch1)
$ git checkout main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 1 commit.
(use "git push" to publish your local commits)

innymi@Mateusz MINGW64 ~ (main)
$ echo "Linia z main zmieniona" > plik.txt

innymi@Mateusz MINGW64 ~ (main)
$ git add plik.txt
warning: in the working copy of 'plik.txt', LF will be replaced by CRLF the next time Git touches
it

innymi@Mateusz MINGW64 ~ (main)
$ git commit -m "Zmiana w pliku.txt na main"
[main 47688f0] Zmiana w pliku.txt na main
1 file changed, 1 insertion(+), 1 deletion(-)

innymi@Mateusz MINGW64 ~ (main)
$ git merge branch1
Auto-merging plik.txt
CONFLICT (content): Merge conflict in plik.txt
Automatic merge failed; fix conflicts and then commit the result.

innymi@Mateusz MINGW64 ~ (main|MERGING)
$ git add plik.txt

innymi@Mateusz MINGW64 ~ (main|MERGING)
$ git commit -m "Rozwiązano konflikt w pliku.txt"
[main 1cfc634] Rozwiązano konflikt w pliku.txt

innymi@Mateusz MINGW64 ~ (main)
$ git remote -v

```

3. Aby wykonać zadanie, rozpocząłem od dodania pliku plik.txt w głównej gałęzi main, do którego wpisałem treść „Linia z main”. Następnie dodałem plik do systemu kontroli wersji (git add plik.txt) i zapisałem zmiany w repozytorium jako commit z opisem „Dodano plik.txt z linią z main”.

Później utworzyłem nową gałąź o nazwie branch1, przełączając się na nią za pomocą komendy git checkout -b branch1. W tej gałęzi edytowałem zawartość pliku plik.txt, wpisując „Linia z branch1”. Po edycji zapisałem plik w systemie kontroli wersji jako commit z opisem „Zmiana w pliku.txt na branch1”.

Po powrocie do głównej gałęzi main (git checkout main) ponownie edytowałem plik plik.txt, wpisując „Linia z main zmieniona”. Te zmiany także zostały zapisane w repozytorium jako commit.

Następnie podjąłem próbę połączenia gałęzi branch1 z main przy użyciu komendy git merge branch1. Operacja zakończyła się konfliktem, ponieważ plik plik.txt został zmodyfikowany zarówno w gałęzi main, jak i branch1. Konflikt został rozwiązany ręcznie poprzez edycję pliku, a następnie zmiany zostały zapisane (git add plik.txt) i potwierdzone jako commit z opisem „Rozwiązano konflikt w pliku.txt”.

Na końcu sprawdziłem konfigurację zdalnego repozytorium za pomocą git remote -v, aby upewnić się, że repozytorium jest poprawnie połączone ze zdalnym serwerem. Dzięki tym krokom zadanie zostało pomyślnie wykonane.

4. Na zdjęciu przedstawiłem realizację kilku operacji w systemie kontroli wersji Git. Wykonano następujące kroki: najpierw wyświetlono zdalne repozytoria za pomocą komendy `git remote -v`, aby sprawdzić poprawność konfiguracji zdalnego połączenia. Następnie pobrano zmiany z zdalnego repozytorium, używając komendy `git fetch`, co pozwoliło na synchronizację bez automatycznego łączenia. Kolejnym krokiem było wysłanie lokalnych zmian do zdalnego repozytorium za pomocą `git push`, co zapewniło zapisanie wprowadzonych modyfikacji. Następnie, za pomocą komendy `git pull`, pobrano i połączono zmiany z zdalnym repozytorium, synchronizując lokalny stan zdalnymi modyfikacjami. Na koniec wyświetlono listę gałęzi przy użyciu komendy `git branch`, co pozwoliło na weryfikację istniejących gałęzi w repozytorium. Wszystkie kroki zostały wykonane poprawnie i z sukcesem zrealizowano synchronizację oraz zarządzanie repozytorium

```
innyi@Mateusz MINGW64 ~ (main)
$ git remote -v
origin  https://github.com/InnyInaczej69/Repozytorium (fetch)
origin  https://github.com/InnyInaczej69/Repozytorium (push)

innyi@Mateusz MINGW64 ~ (main)
$ git fetch

innyi@Mateusz MINGW64 ~ (main)
$ git push
Enumerating objects: 13, done.
Counting objects: 100% (13/13), done.
Delta compression using up to 28 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (12/12), 1.10 KiB | 1.10 MiB/s, done.
Total 12 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/InnyInaczej69/Repozytorium
    153ea16..1cfc634  main -> main

innyi@Mateusz MINGW64 ~ (main)
$ git pull
Already up to date.

innyi@Mateusz MINGW64 ~ (main)
$ git branch
  branch1
* main
```

5. Utworzenie nowej gałęzi i przetączenie na nią

```
innyi@Mateusz MINGW64 ~ (main)
$ git branch nowa_galaz

innyi@Mateusz MINGW64 ~ (main)
$ git checkout nowa_galaz
Switched to branch 'nowa_galaz'
```

6. Wykonanie fast-forward merge

Na zdjęciu przedstawiono proces wykonania operacji *fast-forward merge* w systemie kontroli wersji Git. Na początku utworzono nową gałąź o nazwie *feature* za pomocą komendy `git checkout -b feature`. Następnie dodano nowy plik `feature_file.txt` zawierający tekst "Fast forward merge", co zrealizowano poleceniami

```
innyi@Mateusz MINGW64 ~ (nowa_galaz)
$ git checkout -b feature
Switched to a new branch 'feature'

innyi@Mateusz MINGW64 ~ (feature)
$ echo "Fast forward merge" > feature_file.txt

innyi@Mateusz MINGW64 ~ (feature)
$ git add feature_file.txt
warning: in the working copy of 'feature_file.txt', LF will be replaced by CRLF the next time Git
touches it

innyi@Mateusz MINGW64 ~ (feature)
$ git commit -m "Dodano plik do gałęzi feature"
[feature 58fef83] Dodano plik do gałęzi feature
1 file changed, 1 insertion(+)
create mode 100644 feature_file.txt

innyi@Mateusz MINGW64 ~ (feature)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

innyi@Mateusz MINGW64 ~ (main)
$ git merge feature
Updating 1cf634..58fef83
Fast-forward
 feature_file.txt | 1 +
1 file changed, 1 insertion(+)
create mode 100644 feature_file.txt
```

7. Wykonanie 3-way merge

Wykonałem 3-way merge, zaczynając od utworzenia nowej gałęzi `branch2`, na której dodałem zmiany w pliku `branch2_file.txt`:

Utworzyłem gałąź `branch2` za pomocą komendy `git checkout -b branch2`.

Dodałem zmiany do pliku `branch2_file.txt`, zapisując tekst "Zmiana w branch2", a następnie zatwierdziłem te zmiany używając `git commit -m "Commit na branch2"`.

Następnie wróciłem na gałąź główną `main`, a za pomocą komendy `git merge branch2` połączyłem zmiany z gałęzi `branch2` do `main`. Merge został pomyślnie wykonany, a wszystkie zmiany zostały połączone bez konfliktów.

```
innyi@Mateusz MINGW64 ~ (main)
$ git checkout -b branch2
Switched to a new branch 'branch2'

innyi@Mateusz MINGW64 ~ (branch2)
$ echo "Zmiana w branch2" > branch2_file.txt

innyi@Mateusz MINGW64 ~ (branch2)
$ git add branch2_file.txt
warning: in the working copy of 'branch2_file.txt', LF will be replaced by CRLF the ne
touches it

innyi@Mateusz MINGW64 ~ (branch2)
$ git commit -m "Commit na branch2"
[branch2 4b05d30] Commit na branch2
1 file changed, 1 insertion(+)
create mode 100644 branch2_file.txt

innyi@Mateusz MINGW64 ~ (branch2)
$ git checkout main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 1 commit.
(use "git push" to publish your local commits)

innyi@Mateusz MINGW64 ~ (main)
$ git merge branch2
Updating 58fef83..4b05d30
Fast-forward
 branch2_file.txt | 1 +
1 file changed, 1 insertion(+)
create mode 100644 branch2_file.txt
```

8. Dodanie do Slowiczek do repozytorium i dodanie zdalnego repozytorium raport pokazujący wyniki Ćwiczenia 8.

