

Log:

Technika 1:

```
Command Prompt
remote:
To https://github.com/s27531/apbd16-cw1.2.git
* [new branch]      feature-1 -> feature-1
branch 'feature-1' set up to track 'origin/feature-1'.

C:\Users\Magzik\Documents\GitHub\apbd16-cw1.2>git status
On branch feature-1
Your branch is up to date with 'origin/feature-1'.

nothing to commit, working tree clean

C:\Users\Magzik\Documents\GitHub\apbd16-cw1.2>git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

C:\Users\Magzik\Documents\GitHub\apbd16-cw1.2>git merge feature-1
Updating 697b44d..1618846
Fast-forward
  Zad2/Zad2/Program.cs | 6 +++++
  1 file changed, 6 insertions(+)

C:\Users\Magzik\Documents\GitHub\apbd16-cw1.2>git log --graph --oneline
* 1618846 (HEAD -> main, origin/feature-1, feature-1) Used Add().
* f7de65f Implemented Add().
* 97e570c Added method Add().
* 697b44d (origin/main, origin/HEAD) Prepared main().
* a0569fc Added C# project.
* 4d7d305 Initial commit

C:\Users\Magzik\Documents\GitHub\apbd16-cw1.2>
```

- * 1618846 (HEAD -> main, origin/feature-1, feature-1) Used Add().
- * f7de65f Implemented Add().
- * 97e570c Added method Add().
- * 697b44d (origin/main, origin/HEAD) Prepared main().
- * a0569fc Added C# project.
- * 4d7d305 Initial commit

Technika 2:

```
1 file changed, 1 insertion(+)
C:\Users\Magzik\Documents\GitHub\apbd16-cw1.2>git add .
C:\Users\Magzik\Documents\GitHub\apbd16-cw1.2>git commit -m "change 3."
[feature-2 e0547db] change 3.
1 file changed, 1 insertion(+)
C:\Users\Magzik\Documents\GitHub\apbd16-cw1.2>git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
C:\Users\Magzik\Documents\GitHub\apbd16-cw1.2>git merge --squash feature-2
Updating 697b44d..e0547db
Fast-forward
Squash commit -- not updating HEAD
Zad2/Zad2/Program.cs | 3 +++
1 file changed, 3 insertions(+)
C:\Users\Magzik\Documents\GitHub\apbd16-cw1.2>git commit -m "Squash"
[main 52502d9] Squash
1 file changed, 3 insertions(+)
C:\Users\Magzik\Documents\GitHub\apbd16-cw1.2>git log --oneline --graph
* 52502d9 (HEAD -> main) Squash
* 697b44d (origin/main, origin/HEAD) Prepared main().
* a0569fc Added C# project.
* 4d7d305 Initial commit
C:\Users\Magzik\Documents\GitHub\apbd16-cw1.2>
```

- * 52502d9 (HEAD -> main) Squash
- * 697b44d (origin/main, origin/HEAD) Prepared main().
- * a0569fc Added C# project.
- * 4d7d305 Initial commit

Technika 3:

```
C:\Users\Magzik\Documents\GitHub\apbd16-cw1.2>git commit -m "c3"
[feature-3 bd54bac] c3
1 file changed, 1 insertion(+)
C:\Users\Magzik\Documents\GitHub\apbd16-cw1.2>git checkout feature-3
Already on 'feature-3'
C:\Users\Magzik\Documents\GitHub\apbd16-cw1.2>git rebase main
Current branch feature-3 is up to date.
C:\Users\Magzik\Documents\GitHub\apbd16-cw1.2>git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
C:\Users\Magzik\Documents\GitHub\apbd16-cw1.2>git merge feature-3
Updating 697b44d..bd54bac
Fast-forward
Zad2/Zad2/Program.cs | 3 +++
1 file changed, 3 insertions(+)
C:\Users\Magzik\Documents\GitHub\apbd16-cw1.2>git log --oneline --graph
* bd54bac (HEAD -> main, feature-3) c3
* 3267770 c2
* 7790b12 c1
* 697b44d (origin/main, origin/HEAD) Prepared main().
* a0569fc Added C# project.
* 4d7d305 Initial commit
C:\Users\Magzik\Documents\GitHub\apbd16-cw1.2>
```

- * bd54bac (HEAD -> main, feature-3) c3
- * 3267770 c2

- * 7790b12 c1
- * 697b44d (origin/main, origin/HEAD) Prepared main().
- * a0569fc Added C# project.
- * 4d7d305 Initial commit

```
C:\Users\Magzik\Documents\GitHub\apbd16-cw1.2>git log --oneline --graph --all
* bd54bac (HEAD -> main, feature-3) c3
* 3267770 c2
* 7790b12 c1
| * e0547db (feature-2) change 3.
| * cc8e223 change 2.
| * 4e9b850 Change 1
|/
| * 1618846 (origin/feature-1, feature-1) Used Add().
| * f7de65f Implemented Add().
| * 97e570c Added method Add().
|/
* 697b44d (origin/main, origin/HEAD) Prepared main().
* a0569fc Added C# project.
* 4d7d305 Initial commit

C:\Users\Magzik\Documents\GitHub\apbd16-cw1.2>
```

Podsumowanie:

Zwykły merge – Zachowuje pełną historię commitów i tworzy merge commit. Dobrze sprawdza się w zespołach, gdy historia musi być czytelna.

Squash merge – Kompresuje historie w jeden commit. Przydatny do utrzymania czystej historii na branch głównym.

Rebase i merge – Zachowuje liniową historię, bez merge commitów, Dobrze sprawdza się w repozytoriach z kontrolowaną historią.