**Звіт**

**до предмету «Технології програмування»**

Студента Київського Національного Університету ім. Тараса Шевченко

Пруніча Сергія Миколайовича

Факультет інформаційних технологій

Група МІТ-31

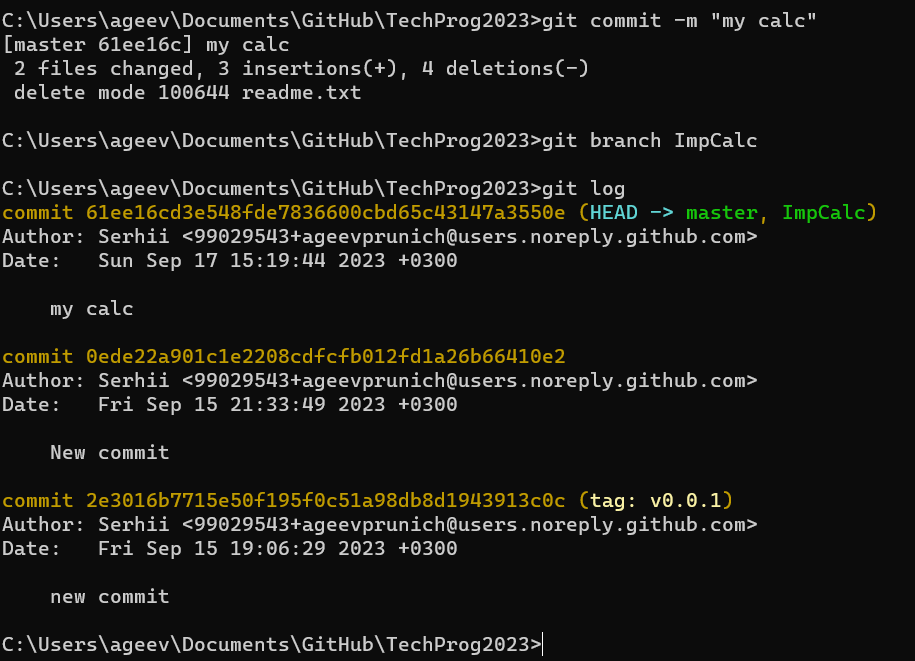


Лабораторна робота №1 (частина 2)

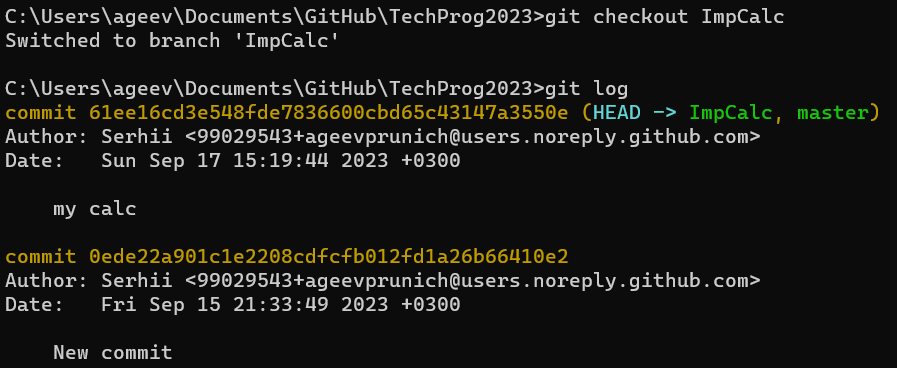
Тема: Робота з різними гілками програми в Git. Конфлікти в Git

Мета: «Навчитися працювати з різними гілками програми в Git, розглянути причини виникнення конфліктів зливання у Git та засоби боротьби з ними.»

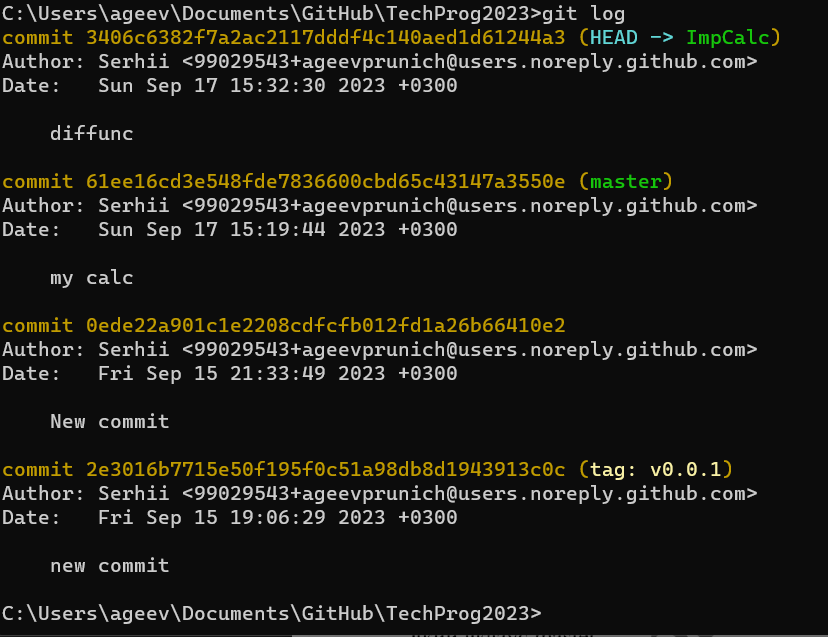
У файл main.py були внесені зміни, а саме створення примітивної функції додавання двох чисел. Файл було проіндексовано та зроблений комміт. Створюється нова гілка під назвою ImpCalc



Зроблений перехід у нову гілку

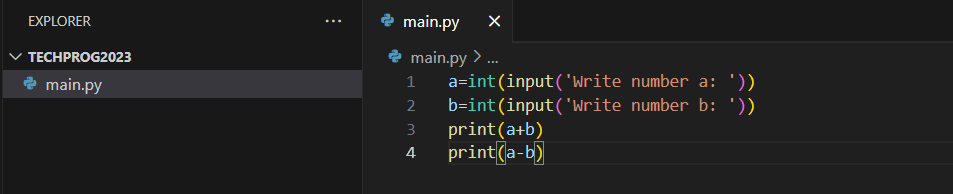


У коді додана функція віднімання, створено новий комміт. Тепер гілка ImpCalc посунулась вперед у порівнянні з master.

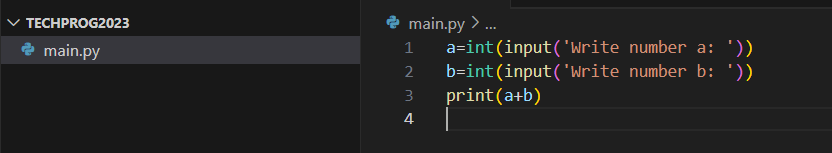


При переході назад у гілку master файл з кодом змінює свій вміст

Гілка ImpCalc

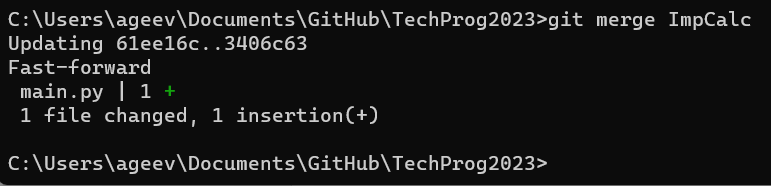


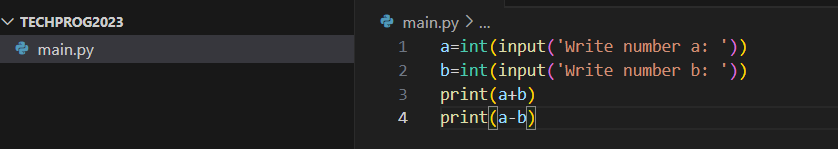
Гілка master



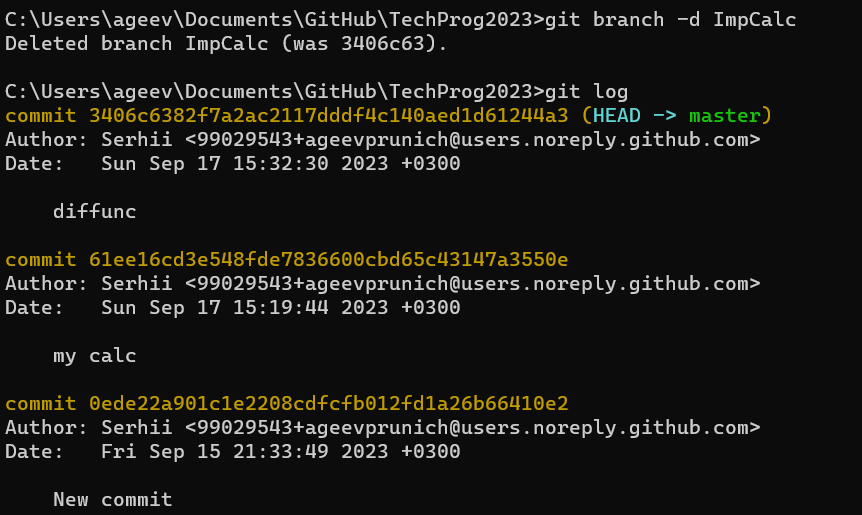
Тепер коли робота над змінами завершена, можна провести злиття гілок.

Для цього потрібно ввести команду git merge ImpCalc

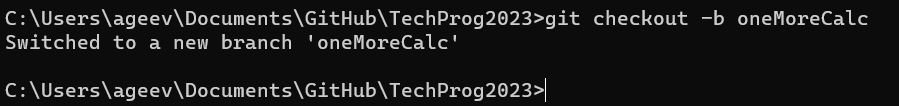




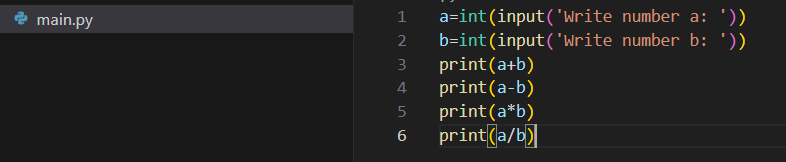
Гілки було об’єднано методом Fast-forward, або двухточкове злиття.

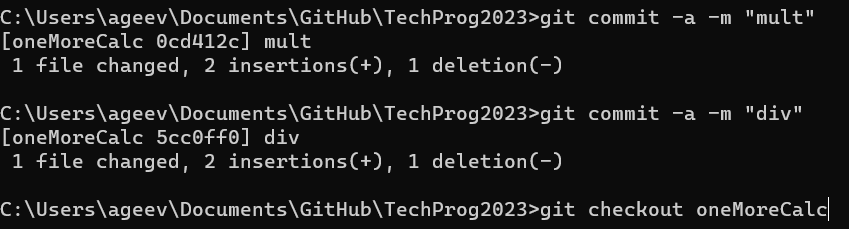
Тепер коли гілка ImpCalc більше не потрібна, її можна видалити за допомогою команди git branch -d ImpCalc 

Для того щоб розглянути інший вид злиття, знову створемо нову гілку і одразу перейдемо в неї

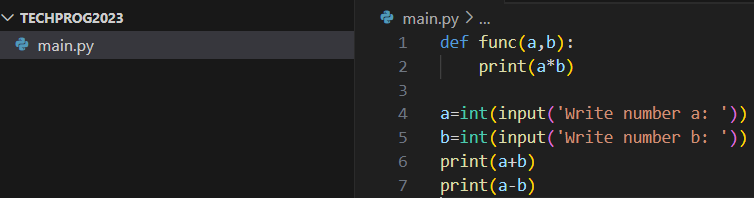


Додаймо функцію множення, створимо комміт і одразу додаємо функцію ділення і знову створюємо комміт.



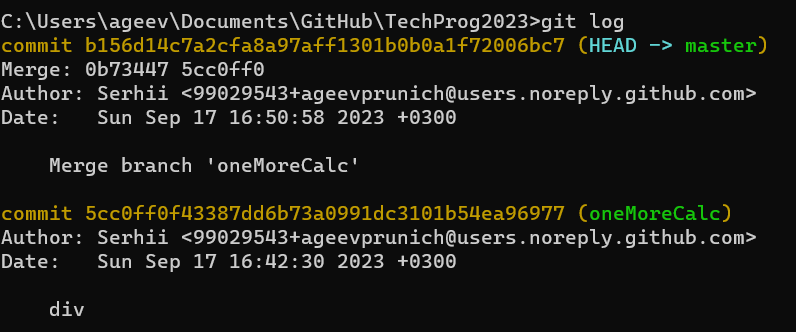


А в гілці master додаємо функцію і також зробимо комміт.

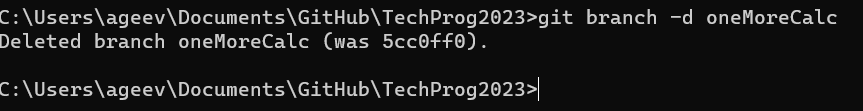




Тепер проводимо злиття гілок тією самою командою.

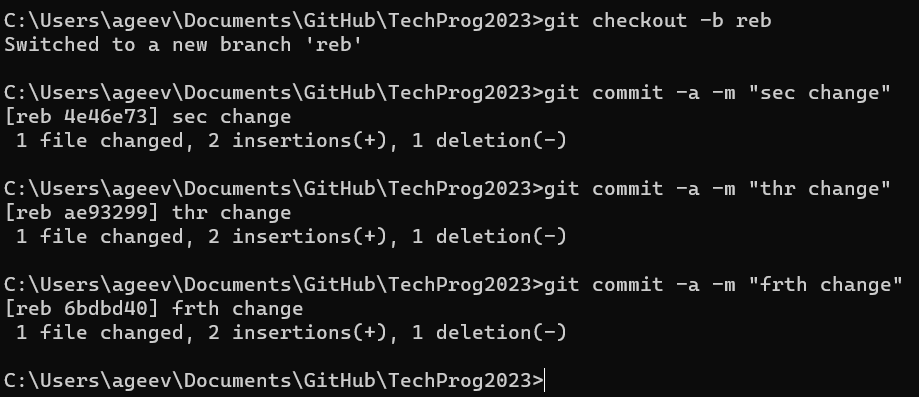


І тепер непотрібну гілку можна видалити

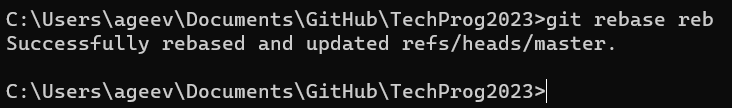


Також розглянемо інший метод поєднання гілок, а саме rebase.

Для цього створюємо нову гілку та робимо в ній декілька коммітів.



Після переходимо назад у гілку master та виконуємо команду git rebase.

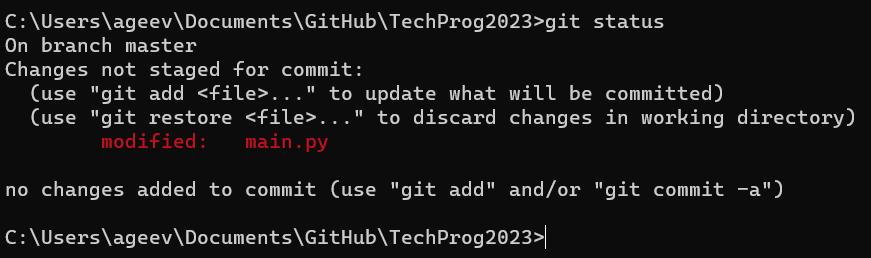


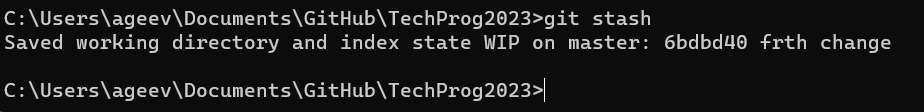
Rebase вважається більш «чистим» методом поєднання гілок.

Тепер гілку можна видалити.

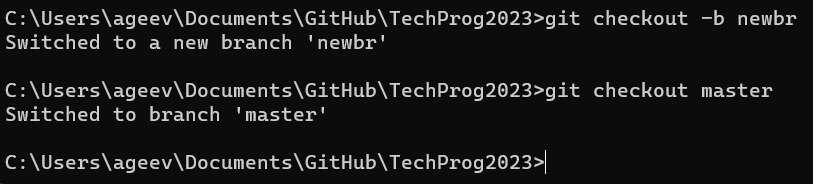
Команда git stash використовується коли ви зробили деякі зміни у файлах і хочете перемкнутися на іншу гілку, але втрачати зміни немає бажання. В такому випадку використовується дана команда. Вона зберігає напрацьовані зміни і дозволяє перемкнутися на іншу гілку (це не комміт).

Наприклад у мене є деякі напрацювання, і мені потрібно зробити перехід на іншу гілку.

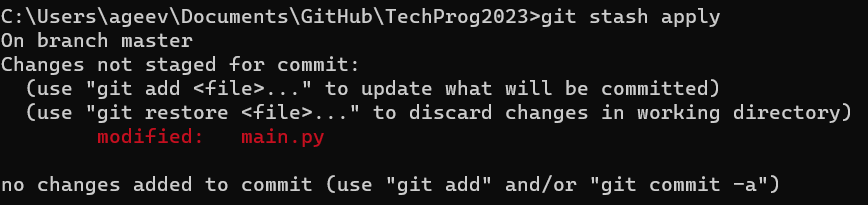




Усі напрацювання збережено, можна виконувати перехід.



Виконавши перехід повертаємось на гілку мастер та бачимо що всі напрацювання зникли. Для того щоб їх повернути необхідно виконати команду git stash apply.



Всі збереженні напрацювання повернулися в робочу область.

Може бути збережено декілька stash-ей, в такому випадку щоб обрати потрібний необхідно в кінці команди додати його назву, наприклад: git stash apply stash@{2}

**Висновки:** «Я навчився працювати з різними гілками програми в Git, розглянув причини виникнення конфліктів зливання у Git та засоби боротьби з ними.»