Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра автоматики та управління в технічних системах

**Лабораторна робота №2**

по дисципліні «Основи клієнтської розробки»

Тема: «Робота з гілками. Shelving в системі контролю версій Git»

|  |  |
| --- | --- |
| Виконав:  студент групи ІА-94  ПІБ: Хотінь К.Ю.  Дата здачі: 17.09.2020  Захищено з балом \_\_\_\_\_\_\_ | Перевірила:  ст. вик. кафедри АУТС  Хмелюк М.С. |

Київ 2020

**Лабораторна робота №2**

1. **Тема:** «Робота з гілками. Shelving в системі контролю версій Git»
2. **Завдання:**

* навчитися створювати гілки, працювати в різних гілках (вносити зміни), здійснювати переходи між гілками. Для проекту створити 3 гілки. У кожній з них реалізувати різний функціонал .Робота з командами commit, update;
* навчитися працювати з функцією Shelve (відкладання змін в системі контролю версій). Додати файл, внести в нього зміни, відкласти на поличку, дістати з полички;
* написати звіт: титульний аркуш, тема роботи, хід виконання роботи зі скриншотами, висновки.

1. **Теоретичні матеріали:**

*Команди:*

*$ git branch [name]*

*$ git checkout [name]*

*$ git merge [name branch]*

*$ git branch –d [name]*

*$ git stash*

*$ git stash pop*

*$ git tag –a [name tag] -m[message]*

*$ git tag*

*$ git status*

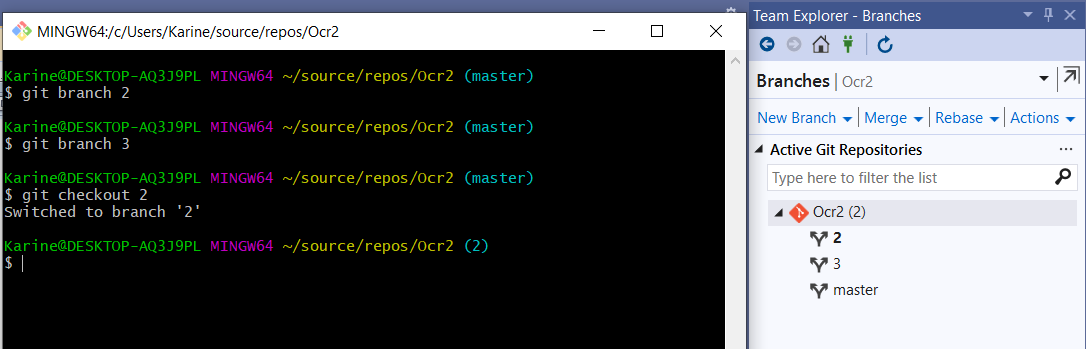
*$ git add --all*

*$ git commit -m "-----"*

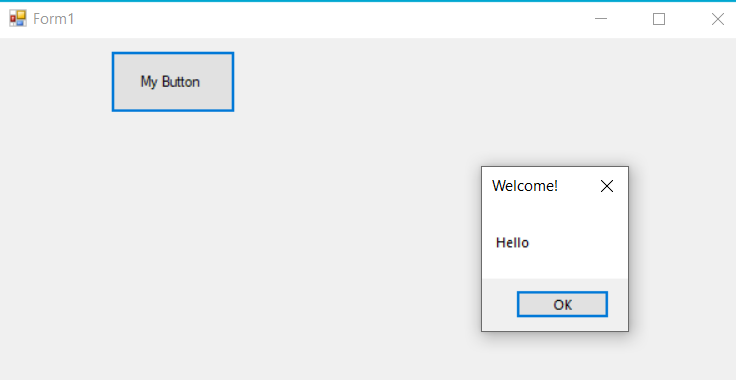
1. **Хід роботи:**

У Git Bash відкриваємо проект, який вже створений у Visual Studio. У гілці master на формі по кнопці відображається напис ОКР. Створюємо у Git Bash за допомогою команди *$ git branch [name]* дві нові гілки. Далі командою *$ git checkout [name]* переходимо на обрану гілку.

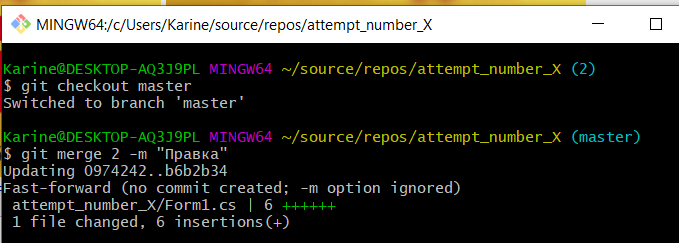
У Visual Studio в блоці Team Explorer графічно можно перевірити створення двох нових гілок, а саме гілка 2 та гілка 3. Далі додаємо нову форму 2 на якій при ініціалізації форми з’являтиметься повідомлення.



Нижче можна побачити результат роботи першої форми.



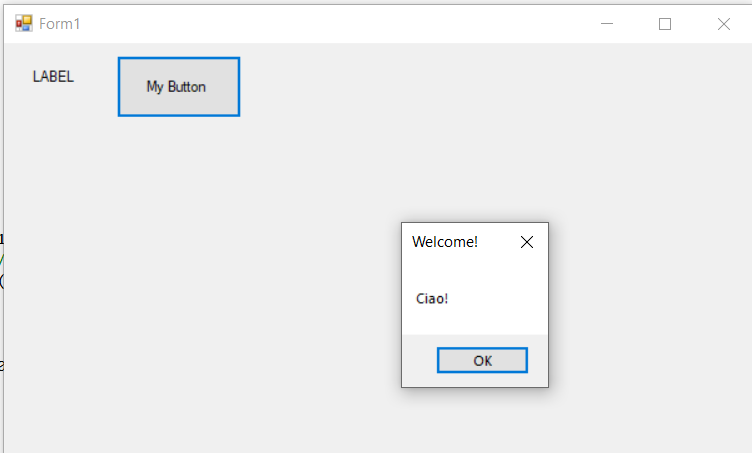
Cторюємо інший функціонал цієї форми в гілці 2 : додали лейбел та змінили повідомлення. В git bash повертаємося до головної гілки, щоб злити функціонал. Використовуємо команду *$ git merge [name branch].*



Після злиття двох гілок допоміжну (тобто 2 ) можна видалити. У цьому випадку використовуємо команду *$ git branch –d [name]*

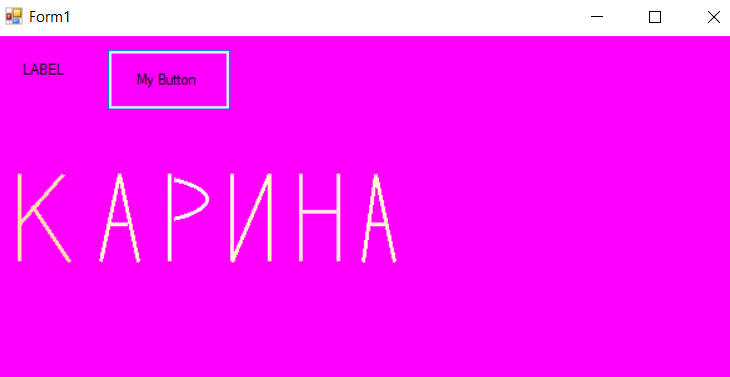


Результат після злиття.

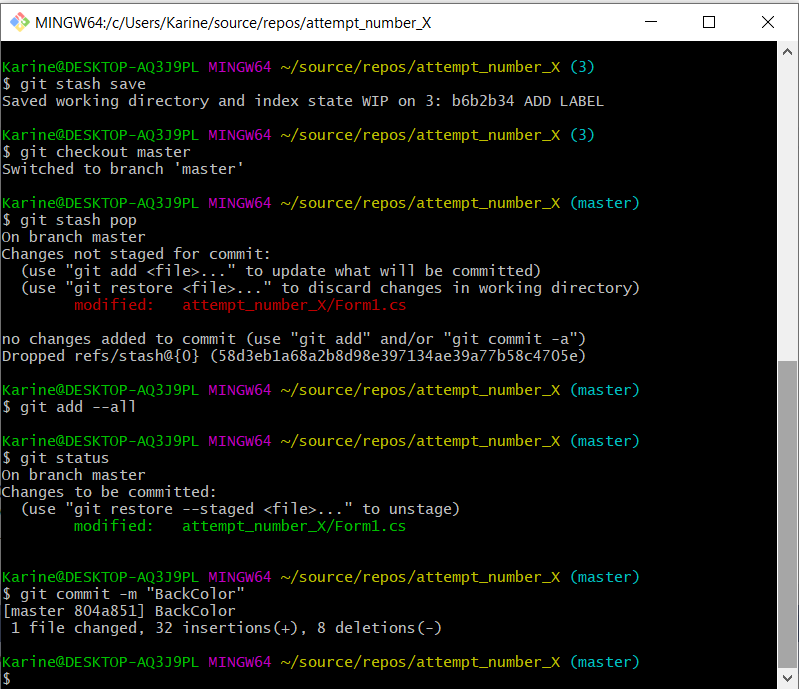


За допомогою *$ git stash* можна тимчасово зберігати всі незакоммічені зміни без необхідності коммітити незавершену роботу.

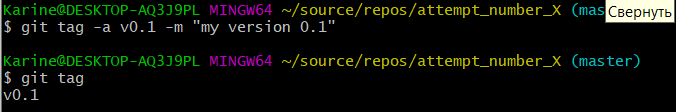
Тому напочатку планувалося змінити колір форми та зробити вивід слова, але потім було обрано робити зміни на другій гілці. Коли злитя другої та головної гілки закінчилося за допомогою комади *$ git stash pop* дістала зміни початкові та закоммітила їх.



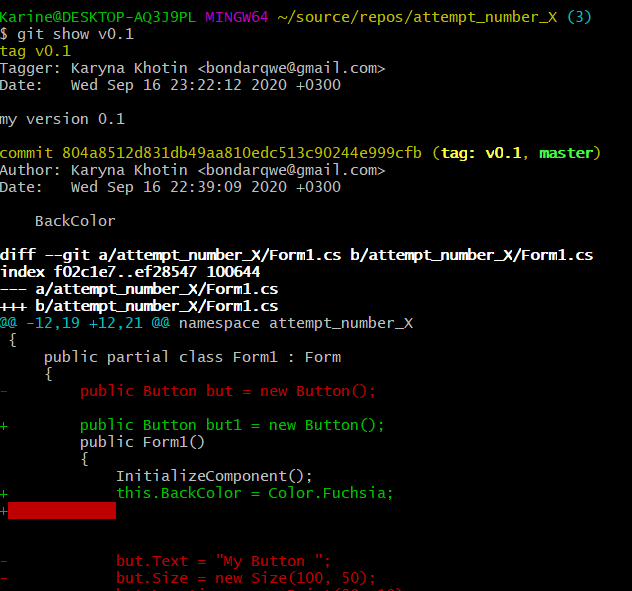
Хід виконання команд



Після усіх змін останньому комміту створено тег для подальшої комфортної робити над проектом.



Перегляд комміту по тегу.



1. **Висновки:** я дізналася про роботу з гілками у Git. Навчилася працювати з основними командами для гілок та $ git stash, $ git stash pop у Git Bash. Cтворювала функціонал та перевіряла роботу команд у Visual Studio.