**Міністерство освіти і науки України**

**Івано-Франківський національний технічний університет нафти і газу**

**Кафедра Програмного забезпечення**

**автоматизованих систем**

**Лабараторна робота №2**

**Тема: Робота з системамою керування версіями Git**

**Виконав:**

**ст. групи ПІ-14-1**

**Дарчин Павло**

**Перевірив:**

**Бандура В.В.**

**Івано-Франківськ 2016**

**Мета:** здобути необхідні знання навички для роботи з системою керування версіями Git**.**

**Завдання**

1. Зареєструватися на GitHub.   
2. Завантажити ПЗ для роботи з Git.   
3. Встановити на робочий компютер.   
4. Створити репозиторій на GitHub.   
5. Створити локальний репозиторій.   
6. Додати в локальний репозиторій файл.   
7. Закомітати даний файл в лок.репозиторій.   
8. Закинути зміни з лок.репозиторія в репозиторій, який ви створили в п.4.   
9. Додати два файли в лок.репозиторій. Закомітайте і запушуйте на GitHub.   
10. Відновіться до стану, який був в п.8.   
11. Зробіть клон репозиторія з GitHub в іншу папку на вашому комп’ютері.   
12. Створіть нову гілку і додайте в неї новий файл.   
13. Закомітайте зміни в новій бренчі. Синхронізуйте з репозиторієм на GitHub.   
14. Об`єднайте створену гілку з основною.   
15. Змініть файл, який ви додали в п.12 в основній бренчі.   
16. Переключіться на гілку, яку ви створили в п.12 і також зробіть зміни в файлі, що додали в п.12.   
17. Об`єднайте гілку з основною. Конфлікт файлу вирішіть в ручну.

**Короткі теоретичні відомості.**

**GitHub** — один з найбільших [веб-сервісів](https://uk.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%B5%D1%80%D0%B2%D1%96%D1%81) для спільної розробки [програмного забезпечення](https://uk.wikipedia.org/wiki/%D0%9F%D0%97). Існують безкоштовні та платні тарифні плани користування сайтом. Базується на [системі керування версіями](https://uk.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D0%BA%D0%B5%D1%80%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F_%D0%B2%D0%B5%D1%80%D1%81%D1%96%D1%8F%D0%BC%D0%B8" \o "Система керування версіями) [Git](https://uk.wikipedia.org/wiki/Git" \o "Git) і розроблений на [Ruby on Rails](https://uk.wikipedia.org/wiki/Ruby_on_Rails" \o "Ruby on Rails) і [Erlang](https://uk.wikipedia.org/wiki/Erlang" \o "Erlang)компанією GitHub, Inc (раніше Logical Awesome).

Сервіс безкоштовний для проектів з відкритим вихідним кодом, з наданням користувачам усіх своїх можливостей (включаючи SSL), а для окремих індивідуальних проектів пропонуються різні платні тарифні плани. [21 вересня](https://uk.wikipedia.org/wiki/21_%D0%B2%D0%B5%D1%80%D0%B5%D1%81%D0%BD%D1%8F) [2011](https://uk.wikipedia.org/wiki/2011) року кількість користувачів стала більшою за мільйон

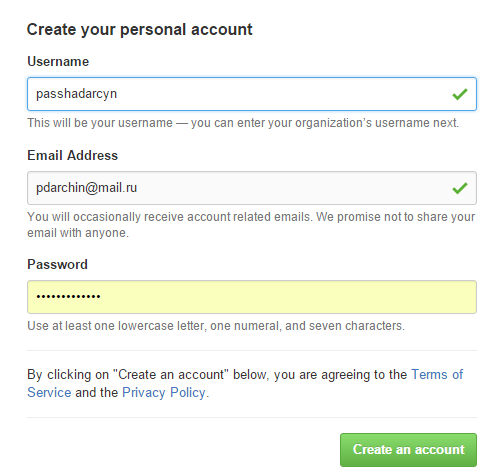
Сайт має елементи [соціальної мережі](https://uk.wikipedia.org/wiki/%D0%A1%D0%BE%D1%86%D1%96%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0_%D0%BC%D0%B5%D1%80%D0%B5%D0%B6%D0%B0) — крім розміщення коду розробники можуть спілкуватися, коментувати правки один одного, слідкувати за новинами знайомих. Також існує можливість об'єднувати різні репозиторії, в тому числі і такі, що мають різних власників.

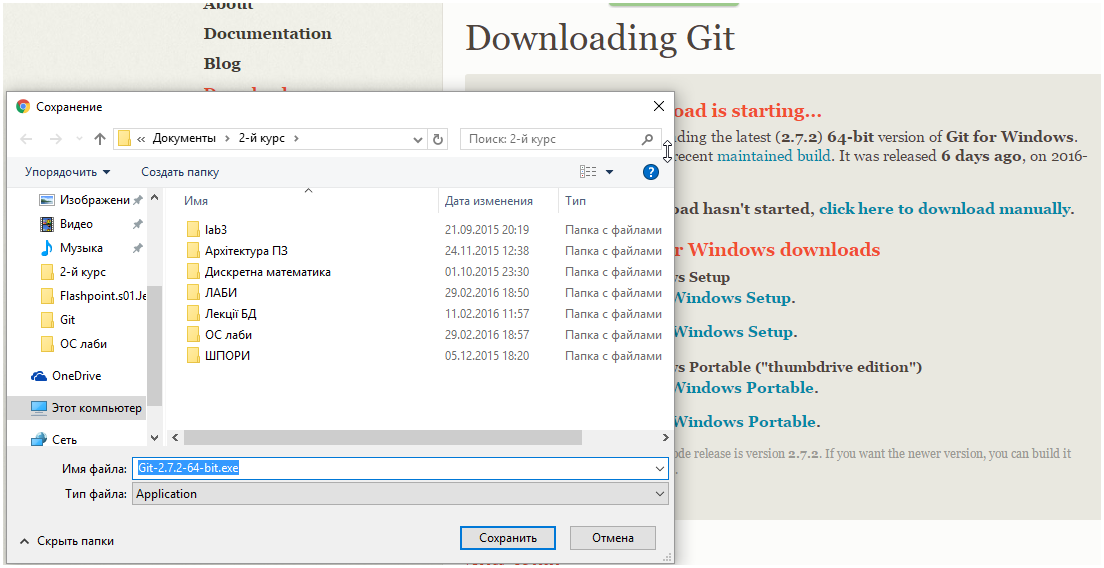
Для проектів є персональні сторінки, [Вікі](https://uk.wikipedia.org/wiki/%D0%92%D1%96%D0%BA%D1%96) та система відслідковування помилок. Прямо на сайті можна переглядати файли проектів, реалізовано підсвічування [синтаксису](https://uk.wikipedia.org/wiki/%D0%A1%D0%B8%D0%BD%D1%82%D0%B0%D0%BA%D1%81%D0%B8%D1%81" \o "Синтаксис) для більшості мов програмування. На платних тарифних планах можна працювати з сервісом через [SSL](https://uk.wikipedia.org/wiki/SSL" \o "SSL) та показувати репозиторій тільки визначеним користувачам. Сам код проектів можна отримати не тільки через Git, а і завантажити звичайний архівний файл.

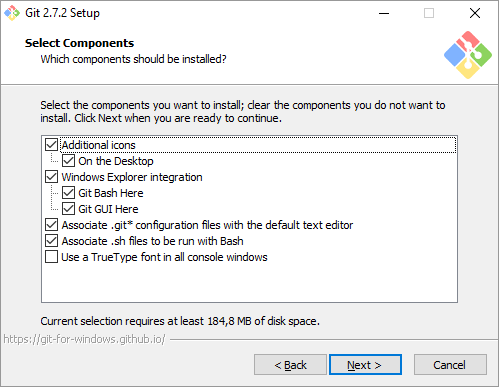
Також на сайті є [pastebin](https://uk.wikipedia.org/wiki/Pastebin)-сервіс [gist.github.com](https://gist.github.com/) для швидкої публікації фрагментів коду.

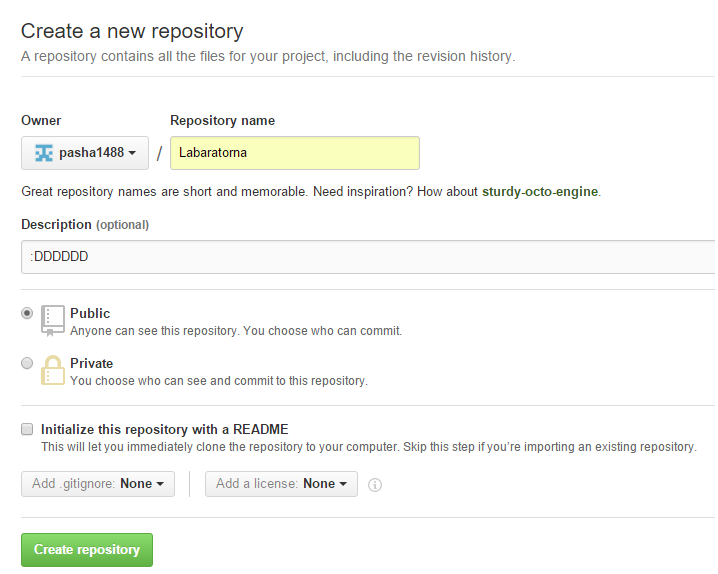
**Виконання**

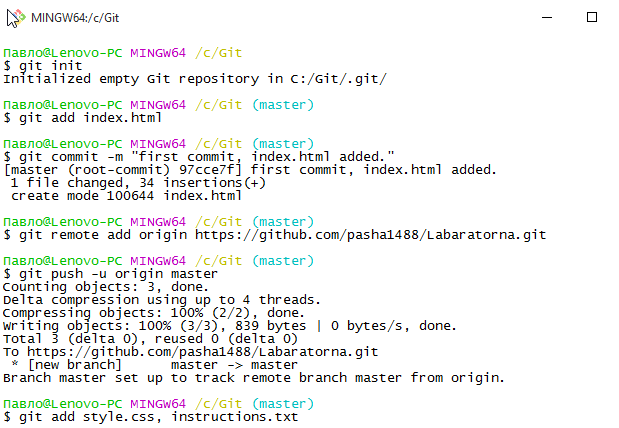
1. Зареєструватися на GitHub.

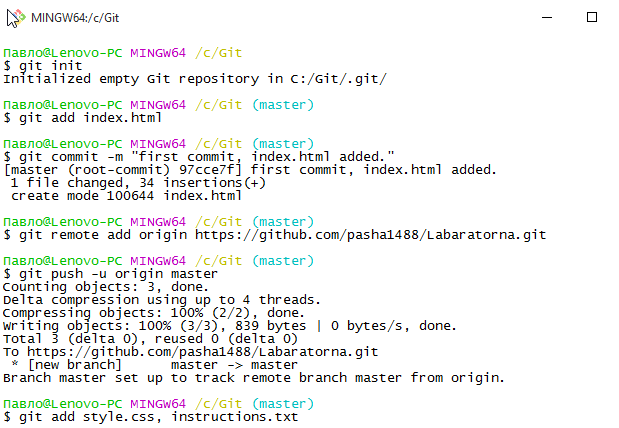
  
2. Завантажити ПЗ для роботи з Git.

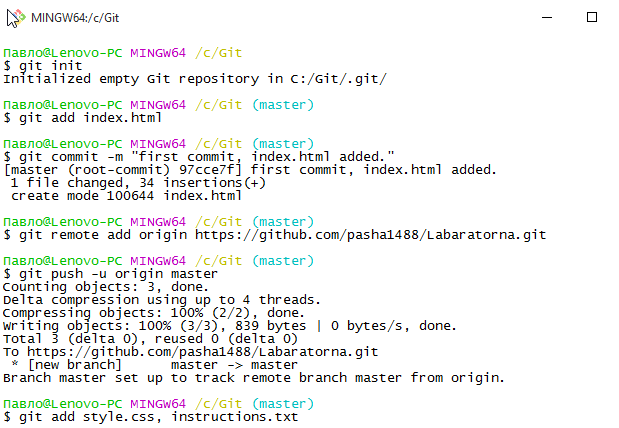
  
3. Встановити на робочий компютер.

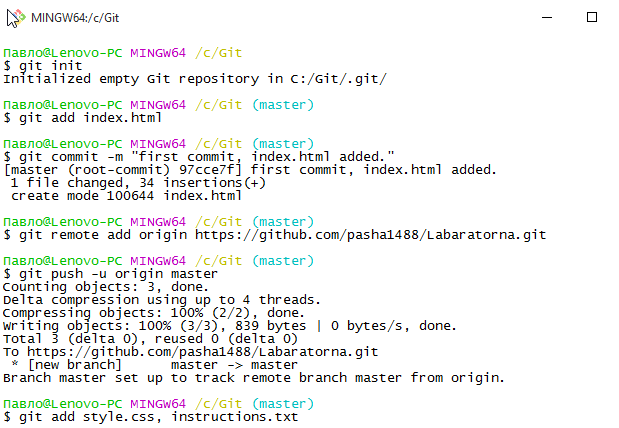
  
4. Створити репозиторій на GitHub.

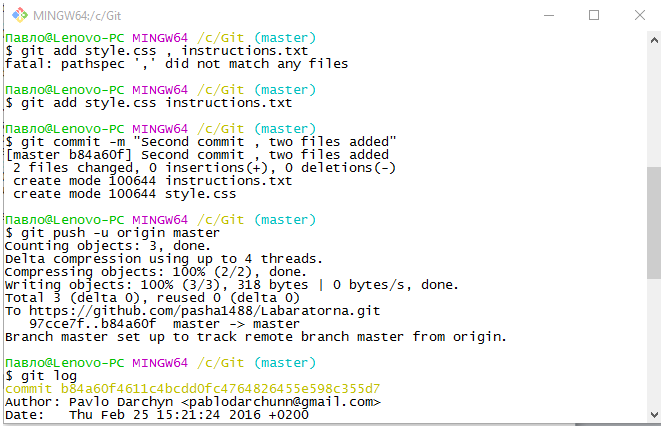
  
5. Створити локальний репозиторій.

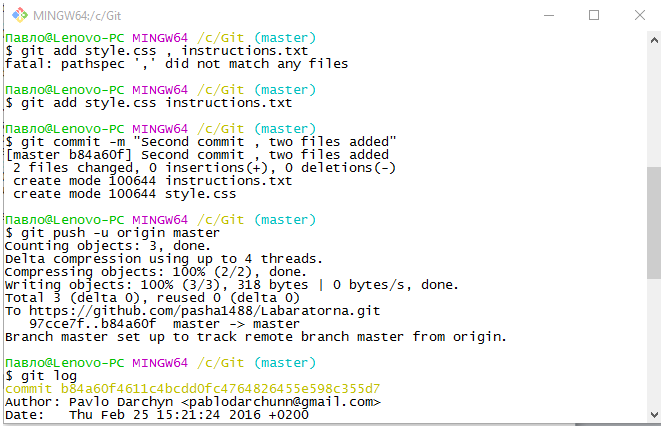
  
6. Додати в локальний репозиторій файл.

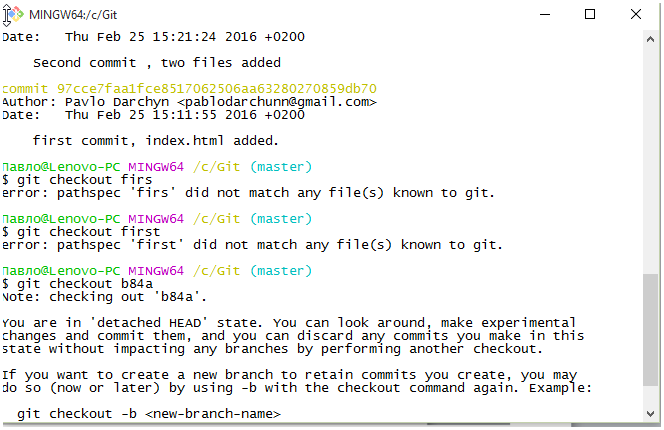
  
7. Закомітати даний файл в лок.репозиторій.

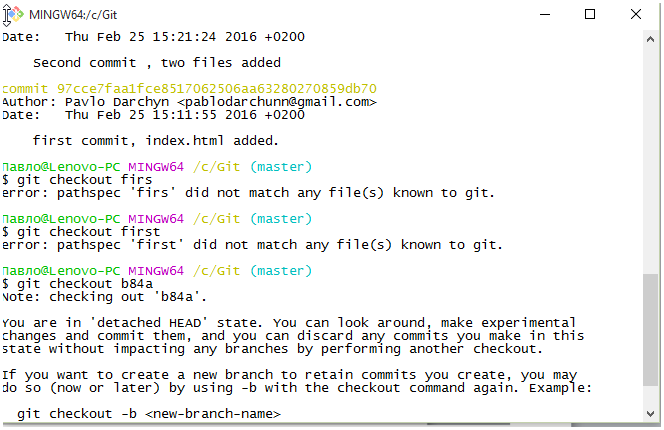
  
8. Закинути зміни з лок.репозиторія в репозиторій, який ви створили в п.4.

  
9. Додати два файли в лок.репозиторій. Закомітайте і запушуйте на GitHub.

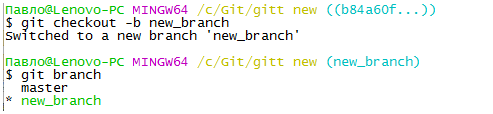
  
10. Відновіться до стану, який був в п.8.

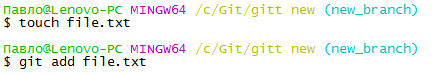


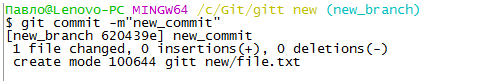


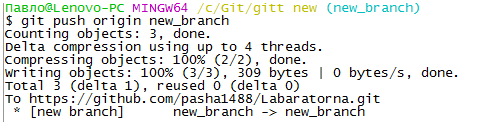
  
11. Зробіть клон репозиторія з GitHub в іншу папку на вашому комп’ютері.

  
12. Створіть нову гілку і додайте в неї новий файл.

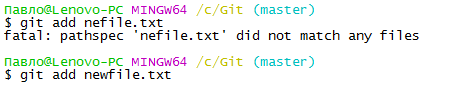


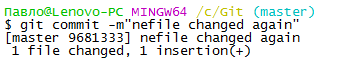
  
13. Закомітайте зміни в новій бренчі. Синхронізуйте з репозиторієм на GitHub.

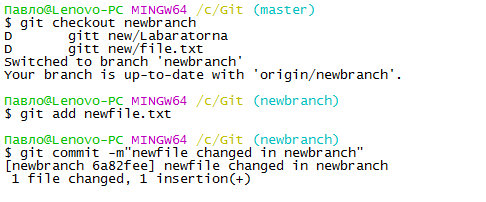


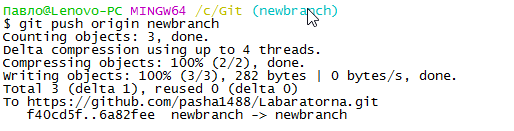
  
14. Об`єднайте створену гілку з основною.

  
15. Змініть файл, який ви додали в п.12 в основній бренчі.

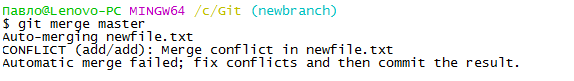


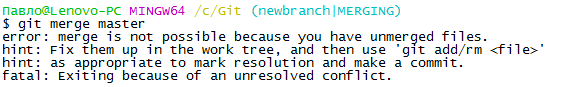
  
16. Переключіться на гілку, яку ви створили в п.12 і також зробіть зміни в файлі, що додали в п.12.





17. Об`єднайте гілку з основною. Конфлікт файлу вирішіть в ручну.





**Висновок:** на даній лабораторній роботі ми ознайомились з системою контролю версій Git.