

Звіт з лабораторної роботи №1

за курсом «Програмування та алгоритмічні мови»

студентки групи ПС-24-1

Біляєвої Олени

кафедри обчислюваної математики та математичної кібернетики, ДНУ

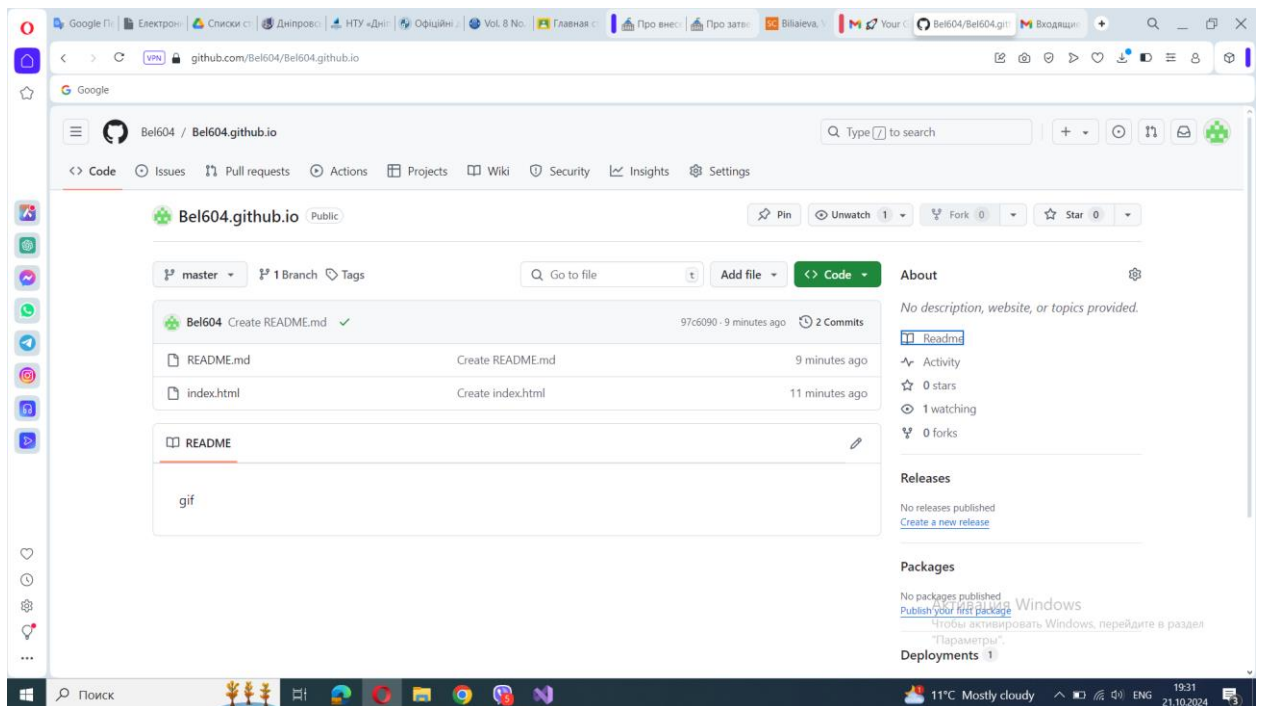
2024/2025

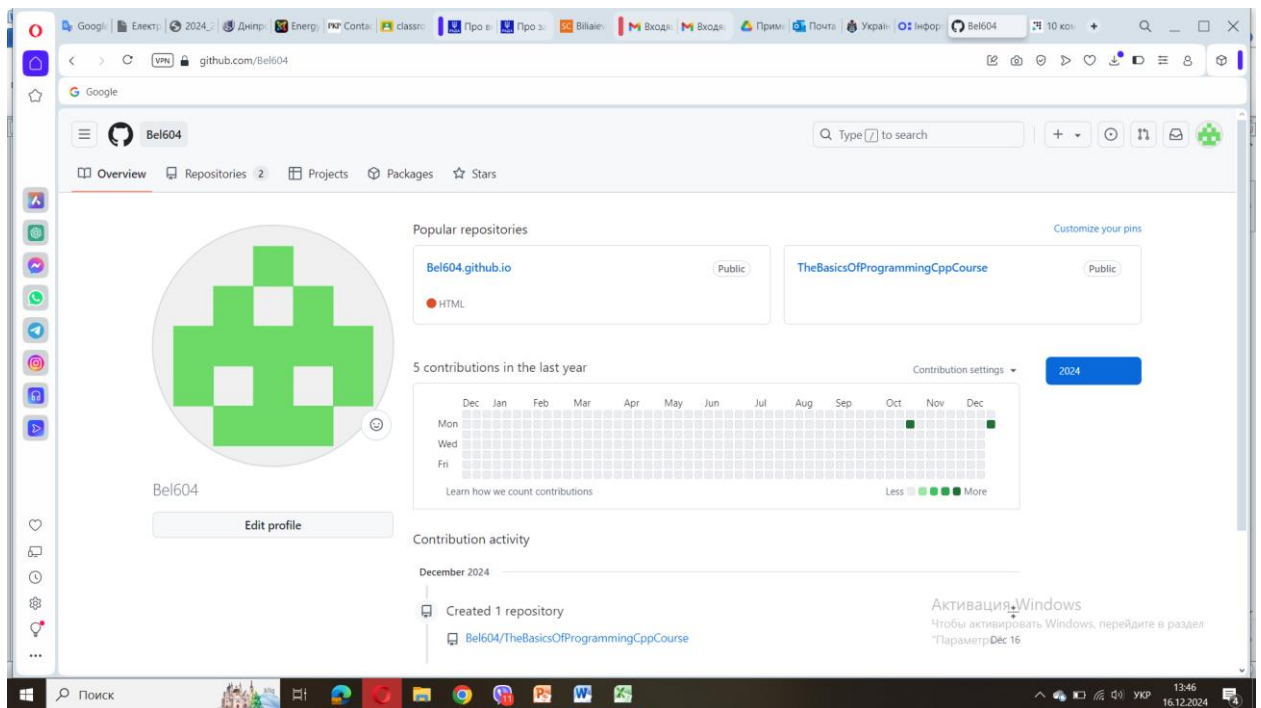
Тема: «Знайомство з «GitHub»

1. Посилання на GitHub репозиторій:

<https://github.com/Bel604/Bel604.github.io?tab=readme-ov-file#readme>

2. Скріни репозиторію:





3. Основні команди **git**:

git init

Команда `git init` дозволяє створити нове сховище Git. Це перша команда для запуску нового проекту в репозиторії GitHub. Перейдіть до каталогу, який містить файли вашого проекту, і виконайте команду `git init`. До нього буде додано прихований підкаталог `.git`.

Використання

\$ git init

Ви також можете вказати назву сховища за допомогою команди `git init`.

\$ git init <your repository name>

git clone

`git clone` створює локальну робочу копію вихідного коду з віддаленого сховища. Коли ви клонуєте репозиторій, код буде автоматично завантажено на вашу машину.

Ця команда додасть початкове розташування як віддалене місце, щоб ви могли отримувати та надсилати зміни з нього, якщо у вас є дозвіл.

Використання

\$ git clone <git-repo-url>

git branch

git branch дозволяє нам додати нову гілку до існуючої гілки, переглянути всі існуючі гілки та видалити їх.

Використання

Створіть нову гілку локально:

\$ git branch <branch-name>

Перелічіть усі локальні та віддалені філії:

\$ git branch -a

Перегляньте всі гілки та подивіться, над якою гілкою ви зараз працюєте:

\$ git branch or \$ git branch –list

Видалити гілку:

\$ git branch -d <branch-name>

Перейменувати гілку:

git branch -m <branch-name> <new-branch-name>

git checkout

Команда git checkout дозволяє перейти до існуючої гілки або створити та перейти до нової гілки. Щоб досягти цього, гілка, на яку ви хочете перейти, має бути присутньою у вашій локальній системі, а зміни у вашій поточній гілці мають бути зафіксовані або сховані до того, як ви переключитесь. Цю команду також можна використовувати для перевірки файлів.

Використання

Перейти до існуючої гілки:

git checkout <branch-name>

Створіть та перейдіть до нової гілки:

git checkout -b <new-branch-name>

git add

Команда `git add` додає ваші зміни у файл до робочої області, де ви можете порівняти свою локальну версію та версію у віддаленому сховищі.

Перш ніж зафіксувати новий або змінений файл, його слід додати до робочої області за допомогою команди `git add`.

Використання

Додайте конкретні файли:

```
$ git add <file-name-1> <file-name-2>
```

Додайте всі нові, змінені та видалені файли:

```
$ git add -A
```

Додайте змінені та видалені файли:

```
$ git add -u
```

git commit

`git commit` зберігає зміни у вашому локальному сховищі. Щоразу, коли ви фіксуєте зміни коду, ви повинні включити короткий опис внесених змін. Це повідомлення про фіксацію допомагає іншим зрозуміти внесені зміни.

Використання

Зафіксуйте всі файли, додані за допомогою команди `git add`, а також файли, які були змінені з того часу:

```
$ git commit -a
```

Після виконання наведеної вище команди відкриється редактор за замовчуванням, який надасть повідомлення про фіксацію.

Закріпити файли з повідомленням:

```
$ git commit -m "<commit-message>"
```

Ви можете замінити дві попередні команди наступною командою:

```
$ git commit -am "<commit-message>"
```

Змініть останній комміт останніми змінами як новий комміт:

```
$ git commit --amend -m "<commit-message>"
```

git push

Команда `git push` надсилає зафіксовані зміни файлу з локального репозиторію у віддалений репозиторій, щоб інші могли їх використовувати. Він також створить іменовану гілку у віддаленому сховищі, якщо вона не існує.

Використання

\$ git push або **\$ git push <remote> <branch-name>**

Якщо ваша гілка є новоствореною, вам потрібно натиснути гілку за допомогою такої команди:

git push –set-upstream <remote> <branch-name> або

git push -u origin <branch-name>

git pull

`git pull` отримує останні завантажені зміни з віддаленого сервера в локальне сховище, щоб ви могли отримувати останні оновлення від ваших товаришів по команді.

Використання

\$ git pull or **\$ git pull <remote>** або **\$ git pull <remote> <branch-name>**

git merge

Команда `git merge` об'єднує вашу гілку з батьківською гілкою. Батьківська гілка може бути розробкою або головною гілкою залежно від вашого робочого процесу.

Команда автоматично створить новий комміт, якщо немає конфліктів. Перш ніж запустити `git merge`, ви повинні бути на конкретній гілці, яку хочете об'єднати з батьківською.

Використання

Перейдіть до гілки, у яку ви хочете об'єднати все:

\$ git checkout <branch>

Перед злиттям вам слід оновити локальну розробку/основну гілку. Це потрібно зробити для того, щоб ваші товариші по команді могли об'єднатися в батьківську гілку, поки ви працюєте над своєю.

Команда для отримання останніх оновлень:

\$ git pull або git fetch

Якщо під час отримання оновлень немає конфліктів, ви можете об'єднати свою гілку з батьківською.

Об'єднайте свою гілку:

\$ git merge <branch-name>

git status

git status надає огляд поточного стану вашого сховища.

Використання

\$ git status

Він надає наступну інформацію про вашу гілку:

- Поточна назва філії.
- Незалежно від того, актуальна ваша поточна філія чи ні.
- Зміни на етапах, які мають бути зафіксовані.
- Зміни, які не підготовлені для фіксації.
- Файли, які не відстежуються.