

Практична робота №1

Тема: Підписання персонального документа за допомогою кваліфікованого електронного підпису (КЕП).

Мета: Набування навичок підписання особистої документації з використанням кваліфікованого електронного підпису.

Очікувані результати навчання: уміння підписувати особисту документацію з використанням кваліфікованого електронного підпису за допомогою різних сервісів і додатків.

Короткі теоретичні відомості

Кваліфікований електронний підпис (КЕП) має таку ж юридичну силу, як і власноручний підпис, та має презумпцію його відповідності власноручному підпису. Наразі в Україні використовується термін «кваліфікований електронний підпис» (КЕП), який прийшов на зміну терміну «електронний цифровий підпис» (ЕЦП) після набрання чинності Закону України «Про електронні довірчі послуги» у 2018 році.

Кваліфікований електронний підпис базується на кваліфікованому сертифікаті відкритого ключа та створюється за допомогою спеціалізованих програмних або апаратно-програмних засобів, що забезпечують найвищий рівень безпеки.

У стандартах Європейського інституту стандартів телекомунікацій (ETSI), визначено три формати розширеного підпису та один формат контейнера підпису:

- удосконалений (advanced) електронний підпис XML (XAdES), заснований на підписах XML;
- удосконалений (advanced) електронний підпис PDF (PAdES), заснований на підписах PDF;
- удосконалений (advanced) електронний підпис CMS (CAdES), заснований на Cryptographic Message Syntax (CMS);
- Асоційований контейнер підпису (ASiC) на основі формату ZIP та який підтримує формати підпису XAdES і CAdES.

Ці формати електронного підпису мають визнаватися як українськими так і європейськими органами державного сектору.

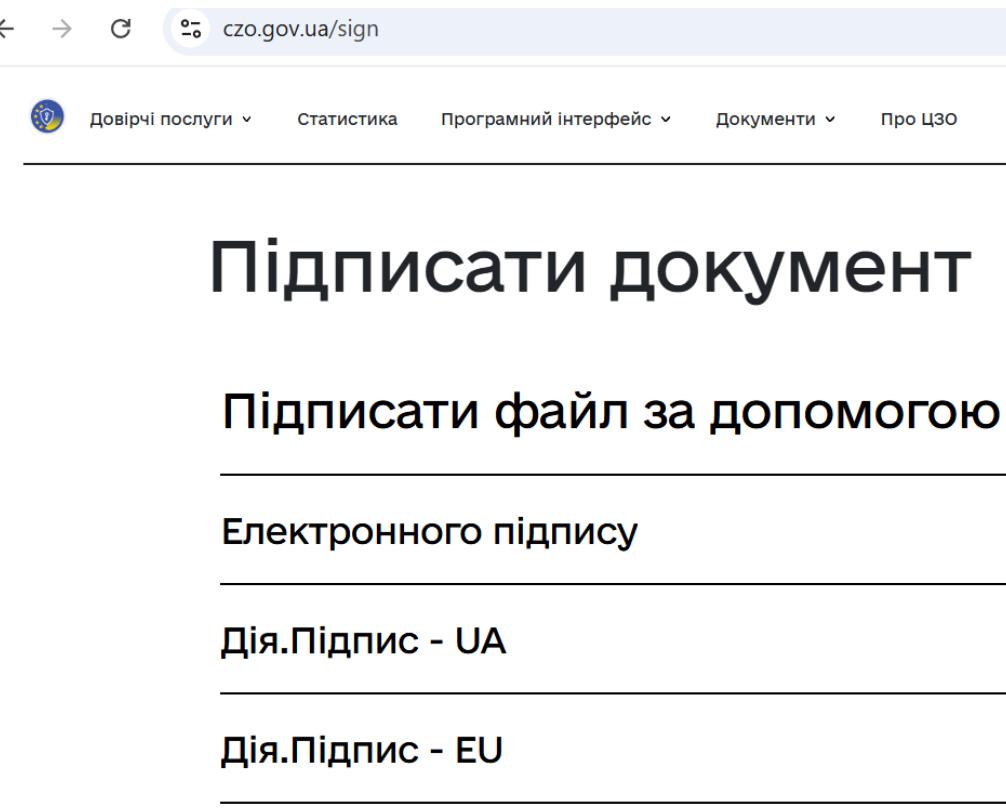
Перевірити накладання підпису на е-документ можна на сайті центрального засвідчувального органу Міністерства цифрової трансформації України <https://czo.gov.ua/verify> або за допомогою спеціального програмного забезпечення для звітності та електронного документообігу.

Завдання.

Створити документ формату *.pdf. В цьому документі написати кілька речень з фактами про себе. Наприклад: «Я Микола і я маю кота Димчика. А ще я обожню баскетбол». Підписати цей документ за допомогою кваліфікованого цифрового підпису (КЕП), використовуючи безкоштовні сервісів - <https://sign.diiia.gov.ua/> чи Дія. Результат виконання надати викладачеві для перевірки.

Приклад щодо накладання кваліфікованого електронного підпису.

Крок 1. Переходимо за посиланням: <https://czo.gov.ua/> (Центральний засвідчувальний орган Міністерство цифрової трансформації України). Далі переходимо на сторінку підписання документації <https://czo.gov.ua/sign>.



← → ⌂ czo.gov.ua/sign

Довірчі послуги Статистика Програмний інтерфейс Документи Про ЦЗО

Підписати документ

Підписати файл за допомогою

Електронного підпису

Дія.Підпис - UA

Дія.Підпис - EU

Рис. 1.1 – Сайт Міністерства цифрової трансформації України

а) Обираєте спосіб підписання. Наприклад за допомогою електронного ключа, який ви створили завчасно на сайті ПриватБанку (рис. 1.2).

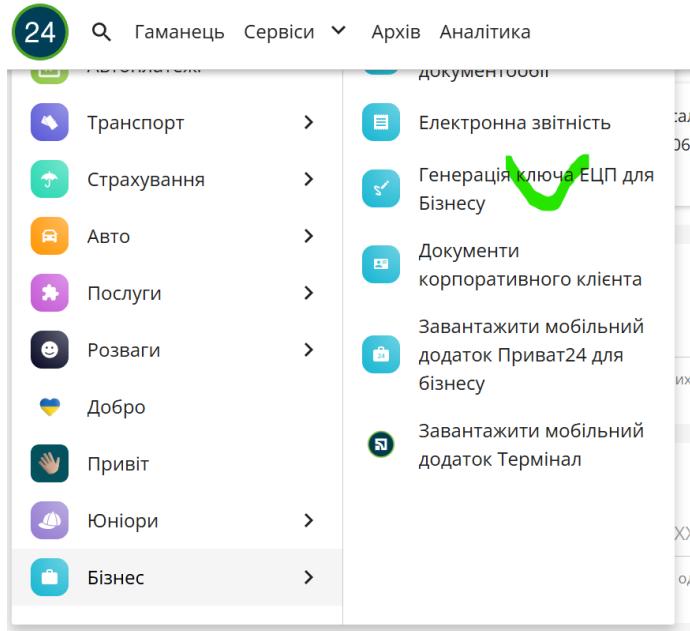


Рис. 1.2 – Генерація ЕЦП

б) Виконуємо дії по ідентифікації електронного ключа: перетягнути файл ключа та ввести пароль від файла ключа.

Підписати документ

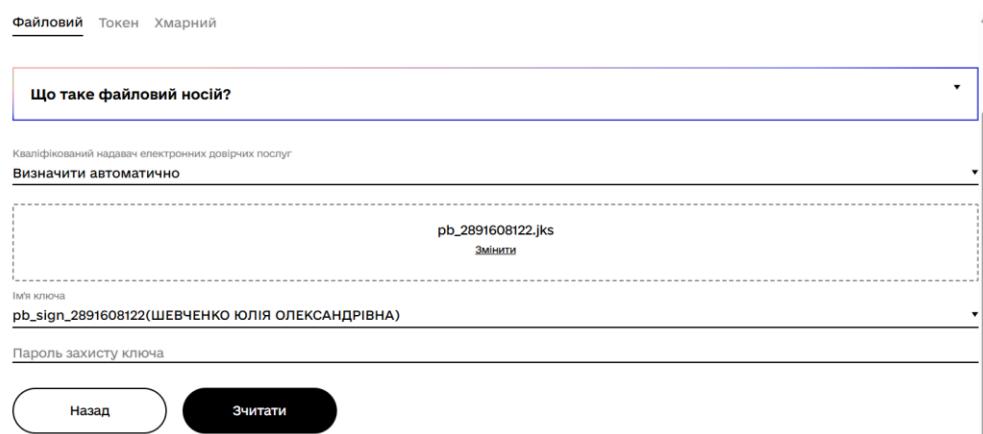


Рис. 1.3 – Ідентифікації електронного ключа

- в) Натискаємо кнопку «Зчитати ключ».
г) Виконуємо підписання документа (**Крок 2**).

Крок 2. Обираємо формат документа, який буде згенерований після підписання документа.

Підписати документ

Крок 3 з 4

Підписати та зберегти

Що таке ASiC?

ASiC – структурований контейнер, що дозволяє зберігати набір файлових об'єктів з пов'язаними е-підписами та/або е-позначками часу, що відповідає специфікації ZIP.

ASiC-S дозволяє зберігати один файловий об'єкт з пов'язаним е-підписом та в подальшому додавати нові. Також дає можливість додавати файли для захисту е-позначок часу.

ASiC-E дозволяє зберігати один або кілька файлових об'єктів з пов'язаними е-підписами та в подальшому додавати файлові об'єкти, файли е-підпису та е-позначки часу.

👉 Рекомендуємо підписувати документи у форматі ASiC-E.

Це уніфікований формат електронного документообігу, який гарантує, що ваші документи прийматимуть всі держоргані.

Підписати та зберегти

Що таке ASiC?

👉 Рекомендуємо підписувати документи у форматі ASiC-E.

Це уніфікований формат електронного документообігу, який гарантує, що ваші документи прийматимуть всі держоргані.

Так, підписати в форматі ASiC-E

Ні, обрати інший формат

Рис. 1.4 – Обрання формату документа з КЕП

Обираєте документ для підписання і натискаємо «**Так, підписати...**»:

Перетягніть сюди файл(и) для підпису
або завантажте його зі свого носія

(doc, pdf, docx та інші)

Рис. 1.5 – Обрання файлу для підписання

Підписати документ

👍 Документ підписано

⬇ Завантажити все архівом

⬇ Файл з підписом
Apz1_Шевченко124м-253-1.pdf.asice
1.9 MB

⬇ Файл(и) без підпису
Apz1_Шевченко124м-253-1.pdf
2.2 MB

⬇ Протокол створення та перевірки кваліфікованого електронного підпису від 29.09.2025
Apz1_Шевченко124м-253-1_Validation_Report.pdf
50.5 KB

Рис. 1.6 Результат підписання на документ КЕП

У результат підписання створюється файл із розширенням *.asice. Саме цей файл має ваш КЕП і його треба надати викладачу на перевірку. Для цього натискаємо значок завантажити у правому куті вікна.

Файл Apz1_Шевченко124м-253-1.pdf.asice збережено до директорії "Завантаження"

⬇ Завантажити все архівом

⬇ Файл з підписом
Apz1_Шевченко124м-253-1.pdf.asice
1.9 MB



Рис. 1.7 Завантаження файлу з КЕП

Для перевірки файлу з КЕП ви можете скористатися сайтом <https://czo.gov.ua/verify>.

Контрольні питання

1. Що таке ЕЦП?
2. Навіщо використовують КЕП?
3. Які сервіси і застосунки використовуються для створення ЕЦП?
4. Який формат має документ після підписання ЕЦП?

Практична робота №2

Тема: Створення і налаштування профілю у системі Git.

Мета: Набування навичок при реєстрації та налаштуванню облікового запису (account) на хостінгу GitHub.

Очікувані результати навчання: уміння створити, налаштувати та підтримувати власний профіль на найбільшому хостингу для сховищ Git.

Короткі теоретичні відомості

Git – це система для керування та контролю версіями. Це найпопулярніший та безкоштовний інструмент, в якому зберігається код та історія його змін.

До основних завдань Git належить збереження коду та історії змін, збереження інформації про користувачів, які змінюють код, можливість відкотити код до будь-якої версії, можливість об'єднувати різні версії, зміни версій, підготовка кінцевого коду до релізу.

GitHub – це інструмент для зберігання та управління репозиторіями Git.

GitHub дозволяє:

- зберігати код;
- використовувати інструменти для спільної роботи;
- оцінювати роботи інших розробників;
- створювати приватні та публічні репозиторії (за приватні стягується плата – користування від трьох і більше осіб).

Завдання.

Створити власний репозиторій в GitHub. В подальшому усі результати своїх практичних робіт необхідно завантажувати у цей репозиторій. В репозиторії створити дляожної практичної роботи окрему папку і розмістить звіт. Якщо ви ще не маєте обліковий запис, будь ласка, в професійному світі, запис повинен виглядати, як поєднання першої літери ім’я та прізвище в повному виді. Наприклад Микола Єфремов буде «myefremov», а Тарас Шевченко «tshevchenko».

Приклад виконання завдання.

Для створення облікового запису у системі GitHub, необхідно виконати декілька простих дій:

- 1) Перейдіть на сайт <https://github.com/>
- 2) Натисніть кнопку “Sign up” у верхньому правому куті сторінки.
- 3) Заповніть реєстраційну форму, вказавши ваше ім’я користувача, адресу електронної пошти та пароль.
- 4) Натисніть “Create account” та дотримуйтесь інструкцій на екрані для завершення реєстрації.

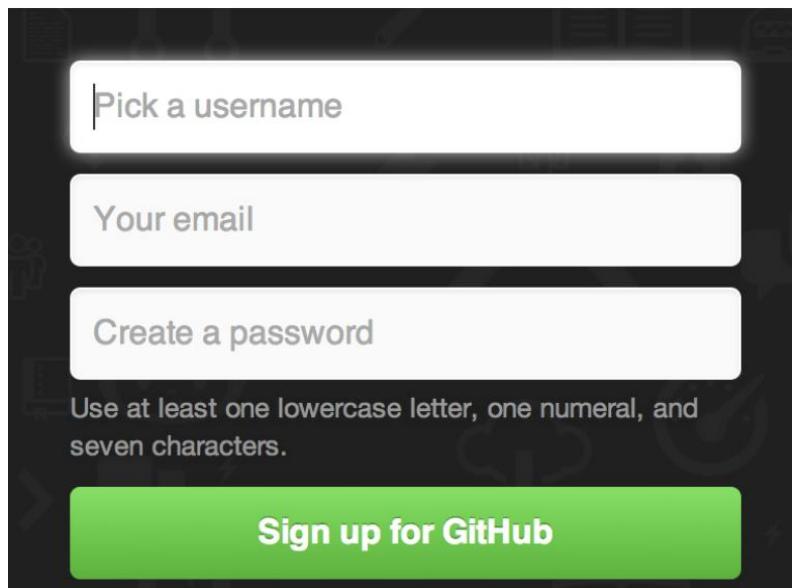


Рис. 2.1 - Форма реєстрації на сайті GitHub

Далі необхідно створити нового проект (репозиторій) для розміщення всіх практичних робіт в ньому.

Репозиторій — це місце може містити файли коду, документацію, зображення і будь-які інші файли, які стосуються проекту.

Для того, щоб створити **новий репозиторій** на GitHub, виконайте наступні дії:

- 1) Увійдіть у ваш обліковий запис на GitHub - “Sign up” у верхньому правому куті сторінки.
- 2) Натисніть на кнопку “New” у верхньому правому куті сторінки, щоб створити новий репозиторій.

- 3) Введіть назву вашого репозиторію і, за бажанням, його опис.
- 4) Виберіть, щоб ваш репозиторій був репозиторій **публічним (Public)**.

Публічний репозиторій і всі файли в ньому зможе побачити будь-яка людина в інтернеті, в той час, як приватний буде доступний лише тим людям, кому ви безпосередньо надасте доступ.

- 5) Натисніть кнопку “Create repository”.

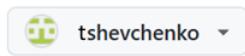
Create a new repository

Repositories contain a project's files and version history. Have a project elsewhere? [Import a repository](#).

Required fields are marked with an asterisk (*).

General

Owner *



Repository name *

/ apz

apz is available.

Great repository names are short and memorable. How about [musical-doodle](#)?

Description

0 / 350 characters

Configuration

Choose visibility *

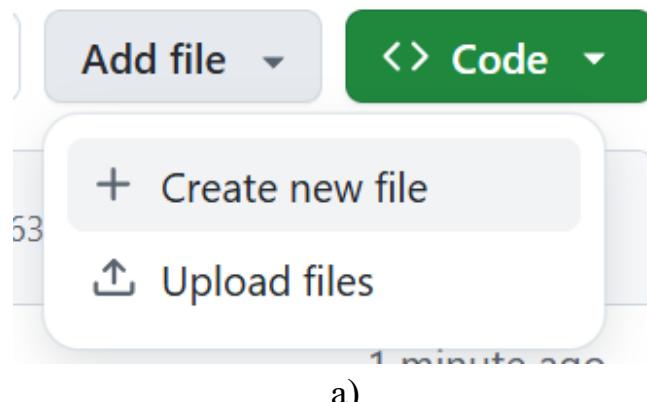
Choose who can see and commit to this repository

Public

Рис. 2.2 – Створення нового публічного репозиторію на GitHub для розміщення практичних робіт

Далі для кожної лабораторної робото зробіть окрему папку і розмістить в ній звіт для перевірки викладачем:

- 1) Створити новий файл, наприклад текстовий документ (рис. 2.3), який потім при необхідності можна видалити чи замінити на звіт з практичної роботи.



<> Code Issues 1 Pull requests Actions

apz / pr1 / about1 in main

Edit Preview

1 Практична робота 1
2 підпис файлу з використарням КЦП

б)



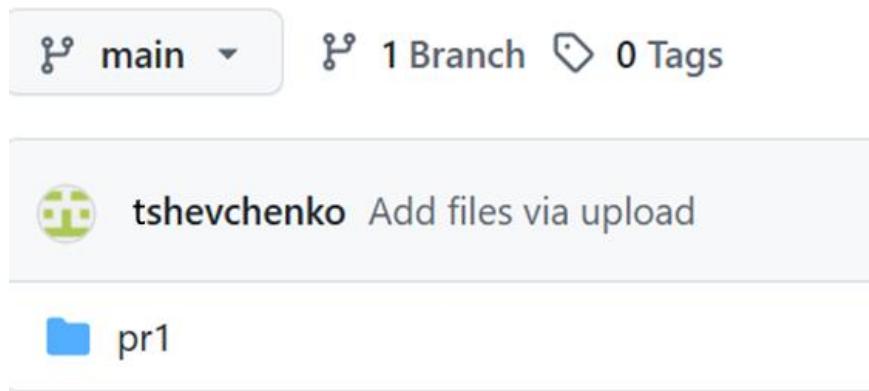


Рис. 2.3 – Послідовність створення окремих папок (а,б,в) у репозиторії для кожної практичної роботи

2) В кожну окрему папку додаємо файл звіту, скориставшись «Upload files» у верхньому куті екрану.

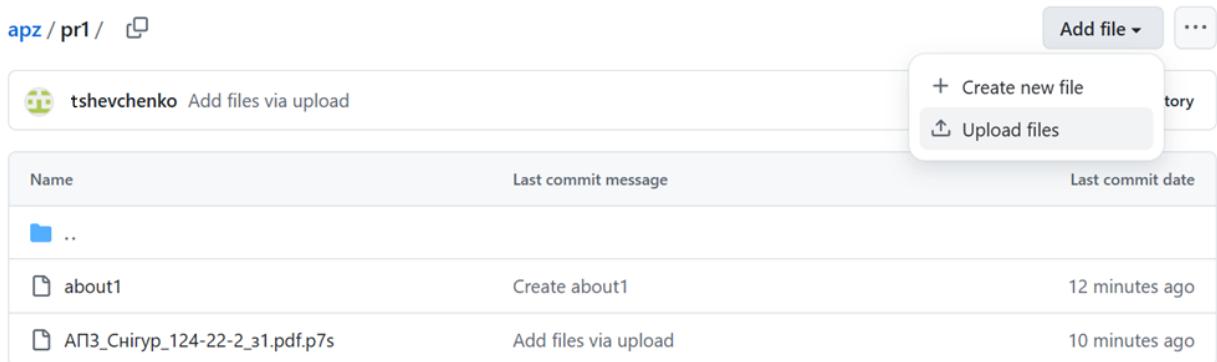


Рис. 2.4 – Додавання звітів з практичних робіт у репозиторії

Контрольні питання

1. Що таке GIT?
2. Що таке репозиторій у GIT?
3. Які переваги використання GIT?
4. Яка мова використовується в GIT?
5. Як можна створити репозиторій у Git?
6. Яка команда використовується для видалення гілки?
7. Що таке контроль версій GIT?
8. Як можна виправити несправний комміт?
9. Як ви дізнаєтесь у GIT, чи гілку вже об'єднано в master?

Аналіз програмного забезпечення (АПЗ)
<https://do.nmu.org.ua/course/view.php?id=2877>

Мінєєв Олександр Сергійович, доцент, канд. техн. наук, доцент, mineev.o.s@nmu.one
Шевченко Юлія Олександрівна – асистент, shevchenko.yu.o@nmu.one

Практична робота №3

Тема: Написання тест-кейсів (Test Case).

Мета: Набування навичок у написанні тест-кейсів різних пристройів.

Очікувані результати навчання: уміння підписувати особисту документацію з використанням єдиного цифрового підпису за допомогою різних сервісів і додатків.

Короткі теоретичні відомості

Тест-кейс – це професійна документація тестирувальника, це послідовність дій, спрямована на перевірку будь-якого функціоналу, що описує як прийти до очікуваного результату.

Тема кейса – описова назва тесту, яка спрощує його пошук та розуміння його змісту.

У темі тест-кейсі не повинно бути залежностей від інших тест-кейсів і також нечітких формулювань та зайвої деталізації.

Навички, необхідні для написання тест-кейсів:

- Уміння збирати та аналізувати вимоги до продукту. Якщо немає формально описаних вимог (специфікацій) – потрібно вміти їх збирати у розробників, аналітиків, користувачів, тощо.
- Уміння розділяти систему на складові, тобто робити декомпозицію. Тобто, потрібно не тільки вміти бачити систему як ціле, а й уміти розкладати її на складові частини.
Це дуже корисна навичка для проведення функціонального тестирування, де перевіряється кожна складова продукту.
- Уміння розставляти пріоритети. Тест-дизайнер повинен уміти відрізняти важливіше від менш важливого, а також розставляти пріоритети тестирування.
- Уміння формулювати свої думки (письмово та усно). Це вміння важливе для тестирувальника в принципі. Тест-дизайнеру воно дуже допомагає при створенні тест-кейсів.
- Знання технік тест-дизайну, а також уміння застосовувати їх на практиці.

Результатом виконання тест кейсу можуть бути:

- Позитивний результат (**pass**) – фактичний результат дорівнює очікуваному результату.
- Негативний результат (**fail**) – фактичний результат не дорівнює очікуваному результату. У цьому випадку знайдено помилку.
- Виконання тесту блоковано (**blocked**) – після одного з кроків продовження тесту неможливе. У цьому випадку також знайдено помилку.

Важливі моменти при складанні тест-кейса:

Розуміння вимоги, за якою складається кейс

Скласти й описати ефективний тест-кейс не вийде, якщо немає розуміння, яким саме чином повинен бути реалізований той чи інший функціонал програми. Ефективніше буде витратити сили та час на уточнення неточностей і в підсумку написати «робочий» тест-кейс, ніж спершу скласти абсолютно неправильний, зрозуміти в процесі тестування, що він не спрацював як треба, знову витратити час на уточнення, і тільки після цього повністю переробити та отримати робочий варіант.

Простота та легкість для розуміння

Важливо, щоб тест-кейс був описаний зрозумілими словами, без використання спеціальної термінології та складних мовних конструкцій, так як його можуть проходити фахівці різного рівня кваліфікації та спеціалізації (досвідчені тестувальники та початківці, керівники проекту, розробники, аналітики та навіть співробітники замовника, не завжди володіють достатнім рівнем знань у галузі комп'ютерних технологій).

Окрема увага приділяється наборам тестових даних

Якщо тест передбачає використання будь-яких даних, слід звернути увагу на правильність їх складання та подальші вказівки. В іншому випадку такий тест є марним

Завдання.

Придумайте об'єкт тестування. Це повинен буде будь-який об'єкт, який складається мінімум з 5 частин. (годинник, скейт, велосипед, монітор і т.п.) Напишіть не менше 20 тест кейсів до цього об'єкту, що зможуть оцінити його якість з точки зору продукту. Створіть *.pdf документ, де кратко описіть ваш об'єкт та потім опишіть ваші тест-кейси.

Приклад Test Case: «Обчислення $1 + 1$ в калькуляторі».

Назва: Перевірка обчислення $1 + 1$.

Pre-condition: Відкрито стандартний калькулятор, система готова до введення виразів.

Кроки:

- Натиснути кнопку “1”.
- Натиснути кнопку “+”.
- Натиснути кнопку “1”.
- Натиснути кнопку “=”.

Expected Result: Відображається число 2

Post-condition: Екран калькулятора показує результат, система готова до наступного обчислення.

Контрольні питання

1. Навіщо потрібні тест-кейси?
2. Основні атрибути Test Case?
3. Типи тест-кейсів.
4. Що таке негативний тест-кейс?
5. Що повинен знати тестувальник?
6. Скільки основних принципів тестування?