

Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»



ЗВІТ
про виконання практичних робіт
з дисципліни «АПЗ»

Виконав:
студент гр. 122-22-6
Хміль О.Р.

Дніпро
2025

Об'єкт тестування: Наручні годинники

Опис: Наручні годинники — пристрій для відображення часу, що складається з таких частин: циферблат (дисплей для показу часу), стрілки (годинна, хвилинна, секундна), ремінець (для фіксації на зап'ясті), корпус (захист механізму), кнопки керування (для налаштування часу/функцій), батареяка (джерело живлення). Годинники призначені для точного відображення часу, стійкості до щоденних умов та зручності використання.

Тест-кейси

Тест-кейс 1: Перевірка відображення поточного часу

- **Назва:** Перевірка коректності відображення часу.
- **Pre-condition:** Годинники увімкнені, батареяка заряджена, час встановлено.
- **Кроки:**
 1. Одягнути годинники на зап'ястя.
 2. Подивитися на циферблат.
 3. Порівняти відображений час із еталонним (наприклад, зі смартфоном).
- **Очікуваний результат:** Час на годинниках збігається з еталонним (± 1 хвилина).
- **Post-condition:** Годинники продовжують показувати час.

Тест-кейс 2: Перевірка руху секундної стрілки

- **Назва:** Перевірка плавності руху секундної стрілки.
- **Pre-condition:** Годинники увімкнені, батареяка заряджена.
- **Кроки:**
 1. Спостерігати за секундною стрілкою протягом 60 секунд.
- **Очікуваний результат:** Стрілка рухається плавно, без зупинок, проходячи повне коло за 60 секунд.
- **Post-condition:** Годинники в робочому стані.

Тест-кейс 3: Перевірка встановлення часу

- **Назва:** Перевірка функції встановлення часу.
- **Pre-condition:** Годинники увімкнені, батареяка заряджена.
- **Кроки:**
 1. Натиснути кнопку налаштування часу.
 2. Прокрутити кнопки для встановлення годин і хвилин (наприклад, 14:30).
 3. Підтвердити встановлення (натиснути кнопку "ОК" або аналог).
 4. Перевірити відображений час.
- **Очікуваний результат:** Годинники показують встановлений час (14:30).
- **Post-condition:** Годинники готові до подальшого використання.

Тест-кейс 4: Перевірка міцності ремінця

- **Назва:** Перевірка стійкості ремінця до розтягнення.
- **Pre-condition:** Годинники з надітим ремінцем, застібка закрита.
- **Кроки:**
 1. Потягнути ремінець із помірною силою (еквівалентною вазі 1 кг).
 2. Перевірити цілісність ремінця та застібки.
- **Очікуваний результат:** Ремінець не рветься, застібка залишається закритою.
- **Post-condition:** Ремінець у початковому стані.

Тест-кейс 5: Перевірка водонепроникності корпусу

- **Назва:** Перевірка водонепроникності при зануренні.
- **Pre-condition:** Годинники увімкнені, корпус герметичний.
- **Кроки:**
 1. Занурте годинники у воду на глибину 1 м на 30 хвилин.
 2. Витягти годинники та перевірити їхню роботу.

- **Очікуваний результат:** Годинники продовжують працювати, циферблат без запотівання.
- **Post-condition:** Годинники сухі та функціональні.

Тест-кейс 6: Перевірка точності ходу за 24 години

- **Назва:** Перевірка точності часу за добу.
- **Pre-condition:** Годинники синхронізовані з еталонним часом.
- **Кроки:**
 1. Залишити годинники увімкненими на 24 години.
 2. Порівняти час із еталонним.
- **Очікуваний результат:** Відхилення не більше ± 15 секунд.
- **Post-condition:** Годинники продовжують працювати.

Тест-кейс 7: Перевірка реакції кнопок керування

- **Назва:** Перевірка відгуку кнопок налаштування.
- **Pre-condition:** Годинники увімкнені.
- **Кроки:**
 1. Натиснути кожну кнопку (налаштування часу, підсвітка тощо) по 5 разів.
 2. Перевірити, чи реагують вони.
- **Очікуваний результат:** Кожна кнопка спрацьовує без залипань.
- **Post-condition:** Годинники в початковому стані.

Тест-кейс 8: Перевірка підсвітки циферблата

- **Назва:** Перевірка роботи підсвітки в темряві.
- **Pre-condition:** Годинники увімкнені, у темній кімнаті.
- **Кроки:**
 1. Натиснути кнопку активації підсвітки.
 2. Спостерігати за циферблатом 10 секунд.
- **Очікуваний результат:** Підсвітка вмикається, час чітко видно.
- **Post-condition:** Підсвітка вимикається автоматично.

Тест-кейс 9: Перевірка стійкості корпусу до падіння

- **Назва:** Перевірка міцності корпусу при падінні.
- **Pre-condition:** Годинники увімкнені, ремінець знятий.
- **Кроки:**
 1. Скинути годинники з висоти 1 м на дерев'яну поверхню.
 2. Перевірити цілісність корпусу та роботу годинників.
- **Очікуваний результат:** Корпус без тріщин, годинники працюють.
- **Post-condition:** Годинники в робочому стані.

Тест-кейс 10: Перевірка роботи при низькій температурі

- **Назва:** Перевірка роботи годинників у холоді.
- **Pre-condition:** Годинники увімкнені.
- **Кроки:**
 1. Помістити годинники в холодильник (0°C) на 1 годину.
 2. Витягти та перевірити роботу.
- **Очікуваний результат:** Годинники показують час коректно.
- **Post-condition:** Годинники в робочому стані.

Тест-кейс 11: Негативний тест — встановлення некоректного часу

- **Назва:** Перевірка обробки некоректного часу.
- **Pre-condition:** Годинники увімкнені.
- **Кроки:**
 1. Спробувати встановити час 25:00.

2. Підтвердити встановлення.

- **Очікуваний результат:** Годинники не приймають значення, показують помилку або скидають час.
- **Post-condition:** Годинники в початковому стані.

Тест-кейс 12: Перевірка роботи без батарейки

- **Назва:** Перевірка реакції на відсутність батарейки.
- **Pre-condition:** Годинники вимкнені, батарейку витягнуто.
- **Кроки:**
 1. Відкрити задню кришку та витягти батарейку.
 2. Спробувати увімкнути годинники.
- **Очікуваний результат:** Годинники не вмикаються.
- **Post-condition:** Батарейку повернуто, годинники працюють.

Тест-кейс 13: Перевірка застібки ремінця

- **Назва:** Перевірка надійності застібки.
- **Pre-condition:** Годинники з надітим ремінцем.
- **Кроки:**
 1. Застебнути ремінець.
 2. Потягнути з силою, еквівалентною 0.5 кг.
- **Очікуваний результат:** Застібка не розстібається.
- **Post-condition:** Ремінець у початковому стані.

Тест-кейс 14: Перевірка читабельності циферблата при яскравому світлі

- **Назва:** Перевірка читабельності при сонячному світлі.
- **Pre-condition:** Годинники увімкнені.
- **Кроки:**
 1. Помістити годинники під пряме сонячне світло.
 2. Перевірити читабельність цифр і стрілок.
- **Очікуваний результат:** Час чітко видно без відблисків.
- **Post-condition:** Годинники в робочому стані.

Тест-кейс 15: Перевірка стійкості до вібрації

- **Назва:** Перевірка роботи при вібрації.
- **Pre-condition:** Годинники увімкнені.
- **Кроки:**
 1. Помістити годинники на віброуючу поверхню (наприклад, поруч із телефоном у режимі вібрації) на 10 хвилин.
 2. Перевірити роботу.
- **Очікуваний результат:** Годинники продовжують працювати коректно.
- **Post-condition:** Годинники в робочому стані.

Тест-кейс 16: Негативний тест — занурення понад допустиму глибину

- **Назва:** Перевірка при зануренні на 50 м.
- **Pre-condition:** Годинники увімкнені, заявлена водонепроникність до 30 м.
- **Кроки:**
 1. Занурте годинники на глибину 50 м (у симуляторі тиску).
 2. Перевірити роботу.
- **Очікуваний результат:** Годинники не працюють або запотівають.
- **Post-condition:** Годинники потребують ремонту.

Тест-кейс 17: Перевірка ергономіки ремінця

- **Назва:** Перевірка зручності носіння.
- **Pre-condition:** Годинники з надітим ремінцем.

- **Кроки:**
 1. Носити годинники на зап'ясті 8 годин.
 2. Перевірити шкіру під ремінцем.
- **Очікуваний результат:** Немає подразнення або дискомфорту.
- **Post-condition:** Годинники в початковому стані.

Тест-кейс 18: Перевірка заміни батарейки

- **Назва:** Перевірка можливості заміни батарейки.
- **Pre-condition:** Годинники вимкнені.
- **Кроки:**
 1. Відкрити задню кришку.
 2. Замінити батарейку на нову.
 3. Закрити кришку та увімкнути годинники.
- **Очікуваний результат:** Годинники вмикаються та показують час.
- **Post-condition:** Годинники в робочому стані.

Тест-кейс 19: Перевірка стійкості скла циферблата

- **Назва:** Перевірка скла на подряпини.
- **Pre-condition:** Годинники увімкнені.
- **Кроки:**
 1. Провести по склу металевим предметом із легким натиском.
 2. Перевірити скло на наявність подряпин.
- **Очікуваний результат:** Скло без подряпин.
- **Post-condition:** Годинники в робочому стані.

Тест-кейс 20: Перевірка тривалості роботи батарейки

- **Назва:** Перевірка часу роботи батарейки.
 - **Pre-condition:** Годинники з новою батарейкою.
 - **Кроки:**
 1. Увімкнути годинники та залишити працювати 6 місяців.
 2. Перевірити, чи працюють годинники.
 - **Очікуваний результат:** Годинники продовжують працювати.
 - **Post-condition:** Годинники в робочому стані.
-

Відповіді на контрольні питання до ПРЗ

1. **Навіщо потрібні тест-кейси?** Тест-кейси потрібні для систематичної перевірки функціональності, якості та відповідності продукту вимогам. Вони допомагають:
 - Структурувати процес тестування.
 - Забезпечити покриття всіх сценаріїв використання.
 - Виявити дефекти та помилки.
 - Документувати результати тестування для аналізу та подальшого вдосконалення продукту.
 - Забезпечити повторюваність тестів для різних версій продукту.
2. **Основні атрибути Test Case?** Основні атрибути тест-кейсу включають:
 - **Назва (Title):** Короткий і зрозумілий опис тесту, що відображає його суть.
 - **Pre-condition (Передумови):** Умови, які повинні бути виконані перед початком тесту (наприклад, пристрій увімкнено, налаштування виконані).
 - **Кроки (Steps):** Покрокова послідовність дій для виконання тесту.
 - **Очікуваний результат (Expected Result):** Опис того, що має статися після виконання тесту (pass, fail або blocked).
 - **Post-condition (Післяумови):** Стан системи після виконання тесту (опціонально). Додатково можуть бути: ідентифікатор тесту, пріоритет, тестові дані, опис дефектів.
3. **Типи тест-кейсів.** Тест-кейси поділяються за різними критеріями:
 - **За функціональністю:**
 - Функціональні (перевірка основних функцій продукту).
 - Нефункціональні (перевірка продуктивності, безпеки, ергономіки).

- **За типом сценарію:**
 - Позитивні (перевірка коректної роботи при правильному використанні).
 - Негативні (перевірка поведінки при некоректному використанні).
 - **За рівнем деталізації:**
 - Високорівневі (загальний опис сценарію).
 - Низькорівневі (детальні кроки).
 - **За етапом тестування:**
 - Модульні (тестування окремих компонентів).
 - Інтеграційні (перевірка взаємодії компонентів).
 - Системні (тестування системи в цілому).
 - Регресійні (перевірка після змін у продукті).
4. **Що таке негативний тест-кейс?** Негативний тест-кейс — це тест, який перевіряє поведінку системи при некоректних, нестандартних або граничних умовах. Наприклад, введення некоректних даних, перевищення допустимих параметрів або використання продукту не за призначенням. Мета — виявити, як система обробляє помилки, чи видає коректні повідомлення про помилки, чи залишається стабільною. Приклад: Спроба встановити на годинниках час 25:00 (Тест-кейс 11 із попередньої відповіді).
5. **Що повинен знати тестувальник?** Тестувальник повинен:
- **Розуміти вимоги до продукту:** Збирати, аналізувати та уточнювати вимоги у розробників, аналітиків чи замовників.
 - **Володіти техніками тест-дизайну:** Використовувати методи, як-от розбиття на класи еквівалентності, аналіз граничних значень тощо.
 - **Вміти декомпонувати систему:** Розкласти продукт на складові для тестування кожної частини.
 - **Розставляти пріоритети:** Визначати, які тести важливіші, залежно від ризиків і вимог.
 - **Формулювати думки:** Чітко описувати тест-кейси та звіти про помилки письмово та усно.
 - **Знати основи тестування:** Принципи, типи тестів, життєвий цикл тестування.
 - **Мати базові технічні знання:** Наприклад, робота з інструментами тестування, основи програмування чи SQL (залежно від проєкту).
 - **Розуміти домен продукту:** Специфіку продукту (наприклад, годинники, ПЗ, апаратне забезпечення).
6. **Скільки основних принципів тестування?** Існує 7 основних принципів тестування (за ISTQB):
- **Тестування показує наявність дефектів:** Тести можуть підтвердити, що дефекти є, але не можуть довести їх відсутність.
 - **Вичерпне тестування неможливе:** Неможливо перевірити всі сценарії, тому потрібен аналіз ризиків і пріоритетів.
 - **Раннє тестування економить час і кошти:** Виявлення дефектів на ранніх етапах зменшує витрати на їх виправлення.
 - **Скупчення дефектів:** Більшість дефектів зосереджена в окремих модулях системи.
 - **Парадокс пестициду:** Повторення тих самих тестів знижує їх ефективність, потрібні нові тести.
 - **Тестування залежить від контексту:** Підходи до тестування залежать від типу продукту (наприклад, годинники чи ПЗ).
 - **Помилка відсутності помилок:** Відсутність знайдених дефектів не означає, що продукт якісний.
-

Відповіді на контрольні питання до ПР2

1. **Що таке GIT?** Git — це розподілена система контролю версій, яка використовується для відстеження змін у вихідному коді або інших файлах під час розробки проєктів. Вона дозволяє кільком розробникам працювати над одним проєктом одночасно, зберігаючи історію змін, забезпечуючи можливість повернення до попередніх версій і керування гілками для паралельної розробки.
2. **Що таке репозиторій у GIT?** Репозиторій у Git — це сховище, де зберігаються всі файли проєкту разом із їхньою історією змін. Це директорія, яка містить файли, папки та спеціальну теку .git, де Git зберігає метадані та історію версій (коміти, гілки, теги тощо). Репозиторій може бути локальним (на комп'ютері користувача) або віддаленим (наприклад, на GitHub).
3. **Які переваги використання GIT?**
 - **Контроль версій:** Відстежує всі зміни, дозволяючи повернутися до будь-якої версії файлу.

- **Розподілена система:** Кожен розробник має повну копію репозиторію, що дозволяє працювати офлайн.
 - **Гілкування та злиття:** Легке створення гілок для паралельної розробки та їх об'єднання.
 - **Спільна робота:** Підтримує одночасну роботу кількох розробників із конфліктним вирішенням.
 - **Гнучкість:** Підходить для будь-яких типів проєктів (код, документація, тести).
 - **Інтеграція:** Сумісність із платформами, як-от GitHub, GitLab, Bitbucket.
 - **Відновлення даних:** Можливість відновити видалені файли чи зміни.
4. **Яка мова використовується в GIT?** Git не прив'язаний до конкретної мови програмування, оскільки це інструмент для контролю версій, який працює з будь-якими файлами (текстовими, бінарними тощо). Сам Git написаний переважно на мові C (для швидкості та портативності), з елементами shell-скриптів, Perl і Python для деяких утиліт. Команди Git вводяться через командний рядок (CLI) або графічні інтерфейси.
5. **Як можна створити репозиторій у Git?** Є два основні способи створення репозиторію:
- **Локальний репозиторій:**
 1. Відкрийте термінал у папці проєкту.
 2. Виконайте команду: `git init`.
 3. Це створить теку `.git` у папці, ініціалізувавши новий репозиторій.
 - **Віддалений репозиторій (наприклад, на GitHub):**
 1. Увійдіть на GitHub (або іншу платформу, наприклад, GitLab).
 2. Натисніть "New repository".
 3. Вкажіть назву, опис, тип (Public/Private), виберіть "Add a README file" (опціонально).
 4. Натисніть "Create repository".
 5. Склонуйте репозиторій локально: `git clone <URL-репозиторію>`.
6. **Яка команда використовується для видалення гілки?**
- Для видалення **локальної гілки**:

bash

```
git branch -d <назва гілки>
```

(Опція `-d` видаляє гілку, якщо вона вже об'єднана. Для примусового видалення використовуйте `-D`.)

- Для видалення **віддаленої гілки**:

bash

```
git push origin --delete <назва гілки>
```

7. **Що таке контроль версій GIT?** Контроль версій у Git — це процес відстеження та керування змінами у файлах проєкту. Git зберігає історію всіх змін у вигляді комітів (знімків стану файлів), дозволяючи:
- Відстежувати, хто, коли і які зміни вніс.
 - Повернутися до попередніх версій файлів чи проєкту.
 - Працювати паралельно через гілки.
 - Вирішувати конфлікти при злитті змін. Це забезпечує прозорість, безпеку даних і співпрацю між розробниками.
8. **Як можна виправити несправний коміт?** Для виправлення несправного коміту є кілька способів залежно від ситуації:
- **Виправлення останнього коміту (без push):**
 1. Внесіть зміни у файли.
 2. Додайте їх до індексу: `git add <файли>`.
 3. Виконайте:

bash

```
git commit --amend
```

Це додасть зміни до останнього комміту, дозволяючи також змінити повідомлення.

- **Виправлення після push (відалений репозиторій):**

1. Виправте файли та створіть новий комміт: `git commit -m "Виправлення"`.
2. Виконайте примусовий push:

bash

```
git push --force
```

(Обережно: це може вплинути на інших розробників).

- **Скасування комміту:**

1. Для скасування останнього комміту (збереження змін):

bash

```
git reset --soft HEAD~1
```

2. Для повного видалення комміту та змін:

bash

```
git reset --hard HEAD~1
```

9. Як ви дізнаєтесь у GIT, чи гілку вже об'єднано в master? Щоб перевірити, чи гілку об'єднано в master:

- Виконайте команду:

bash

```
git branch --merged master
```

Це покаже список гілок, які вже об'єднані в master. Якщо ваша гілка (наприклад, feature-branch) є в списку, вона була об'єднана.

- Альтернативно:

1. Перейдіть на гілку master: `git checkout master`.
2. Перегляньте історію комітів: `git log`.
3. Якщо коміти з вашої гілки присутні в master, це означає, що гілку об'єднано.

- На GitHub: Перейдіть до репозиторію → вкладка "Pull Requests". Якщо для гілки є закритий Pull Request із позначкою "Merged", гілку об'єднано.

Відповіді на контрольні питання до ПР1

1. **Що таке ЕЦП?** ЕЦП (електронний цифровий підпис) — це електронний підпис, який створюється за допомогою криптографічних алгоритмів і використовується для підтвердження автентичності документа, ідентифікації особи, яка його підписала, та забезпечення цілісності даних. ЕЦП є аналогом власноручного підпису в цифровому середовищі, що дозволяє підписувати електронні документи, забезпечуючи їх юридичну силу.
2. **Навіщо використовують КЕП?** КЕП (кваліфікований електронний підпис) — це вдосконалений вид ЕЦП, який відповідає суворим вимогам законодавства (наприклад, Закону України "Про електронні довірчі послуги"). КЕП використовується для:
 - **Підтвердження особи:** Гарантує, що документ підписала конкретна людина.

- **Забезпечення цілісності:** Захищає документ від несанкціонованих змін.
- **Юридична сила:** Надає електронним документам таку ж силу, як паперовим із власноручним підписом.
- **Автоматизація процесів:** Використовується в електронному документообігу, поданні звітності, укладанні договорів, доступі до державних сервісів (наприклад, податкова, держреєстри).
- **Безпека:** Захищає від підробки документів.

3. **Які сервіси і застосунки використовуються для створення ЕЦП?** Для створення та використання ЕЦП/КЕП в Україні застосовуються такі сервіси та застосунки:

- **Дія.Підпис:** Мобільний застосунок від Мінцифри для створення та використання КЕП (інтегровано з ID-картками та біометричними паспортами).
- **Приват24:** Сервіс ПриватБанку для генерації та використання КЕП для клієнтів банку.
- **Кваліфіковані надавачі електронних довірчих послуг (КНЕДП):**
 - АЦСК "Україна".
 - АЦСК ДПС (для податкової звітності).
 - АЦСК "ПриватБанк".
 - АЦСК "Державна казначейська служба".
- **Програмне забезпечення:**
 - "ІТ Користувач ЦСК" (для роботи з ключами ЕЦП/КЕП).
 - "М.Е.Дос" (для підписання звітів та документів).
 - "СОТА" (вебсервіс для подання звітності з підтримкою КЕП).
- **Апаратні засоби:** Токени (SecureID, Aladdin, тощо) для зберігання ключів КЕП.
- **Хмарні сервіси:** Хмарний КЕП від КНЕДП, який зберігає ключі на віддалених серверах (наприклад, у "Дія" або "Приват24").

4. **Який формат має документ після підписання ЕЦП?** Після підписання ЕЦП документ може мати різні формати залежно від способу підписання:

- **Формат P7S:** Найпоширеніший формат для електронного підпису. Документ залишається у своєму початковому форматі (наприклад, PDF, DOC), а підпис додається як окремий файл із розширенням .p7s (на основі стандарту PKCS#7).
- **Вбудований підпис у PDF:** У PDF-документах ЕЦП може бути інтегровано безпосередньо в файл (видимий або невидимий підпис), зберігаючи розширення .pdf.
- **Формат CAdES або XAdES:** Використовується для XML-документів або інших спеціалізованих форматів, де підпис вбудовано в структуру документа.
- **ASiC (Associated Signature Containers):** Контейнер, який об'єднує документ і підпис (наприклад, .asice, .asics). У більшості випадків в Україні після підписання КЕП створюється додатковий файл .p7s, який містить інформацію про підпис, а оригінальний документ залишається незмінним.