

Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

Лабораторна робота №1
з дисципліни
«Розробка мобільних застосунків під Android»

Виконала:
студентка 3 курсу ФІОТ
групи ІО-23
Чепенюк Анастасія

Київ 2025

Тема: дослідження роботи з елементами керування.

Мета роботи: дослідити створення простого застосунку під платформу Андроїд та набути практичні навички з використання елементів керування інтерфейсу, мов програмування Java чи Kotlin.

Завдання: Написати програму під платформу андроїд, яка має інтерфейс для введення Або/та вибору даних згідно варіанту і відображає результат взаємодії з цим Інтерфейсом у деяке текстове поле цього інтерфейсу.

Передбачити наступне: якщо не Всі дані введені або обрані, а користувач натискає кнопку для отримання результату, То відобразити вікно, що спливає, з повідомленням завершити введення всіх даних.

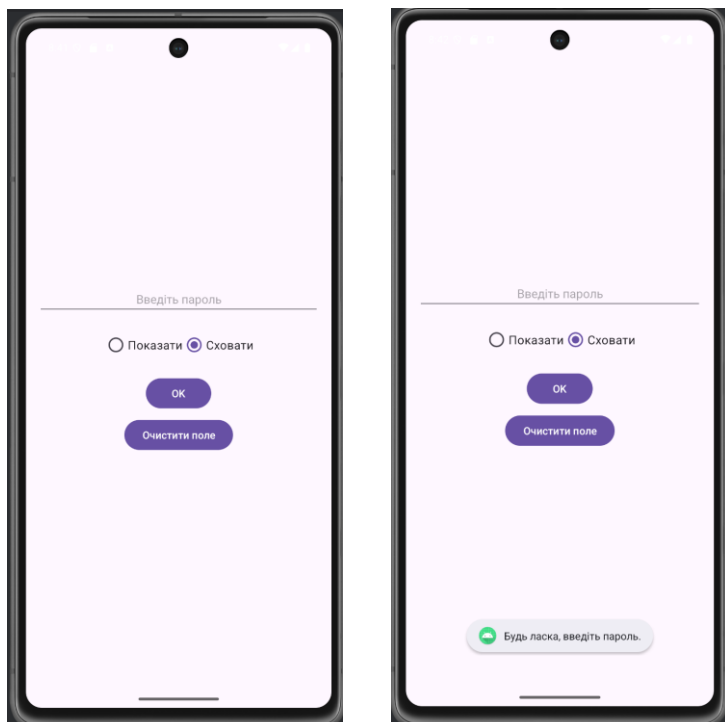
Мій номер в списку: 159

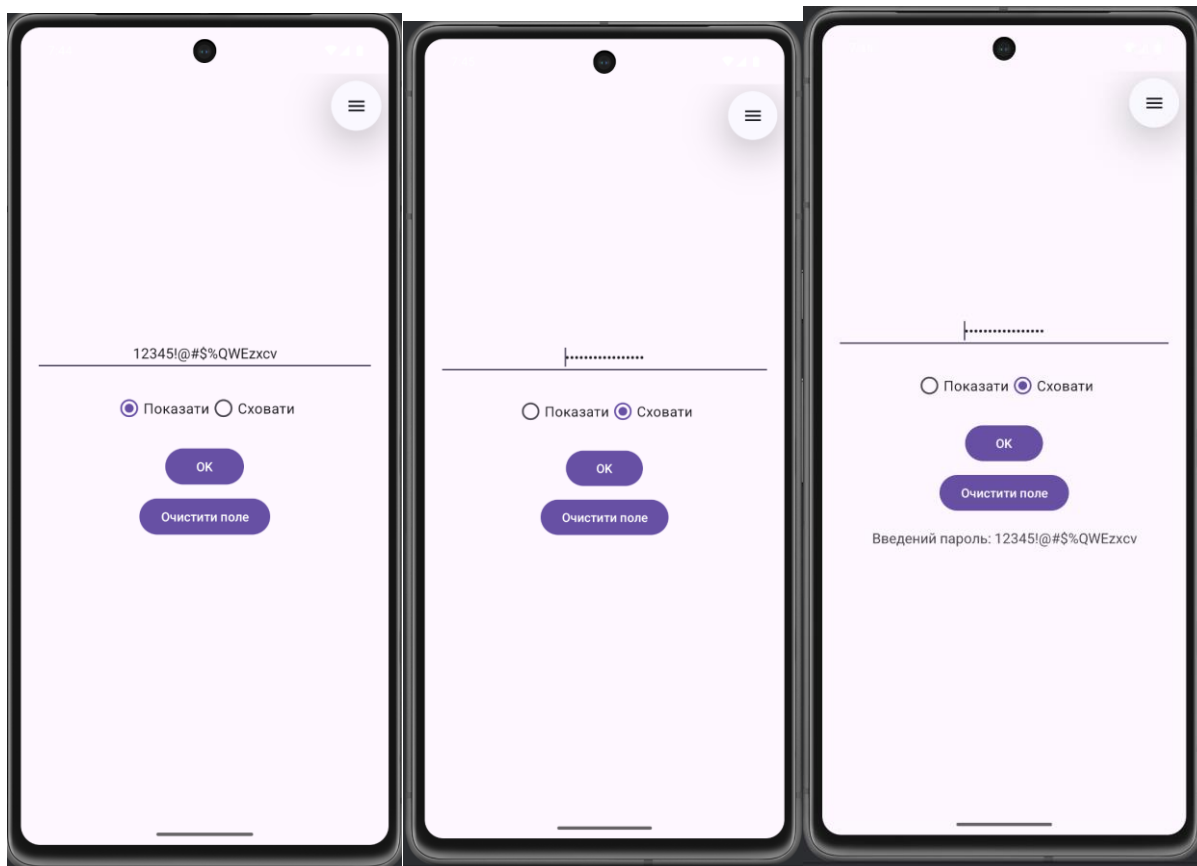
Варіант: 6

6.	Вікно введення паролю містить: текстове поле, дві опції (радіо-батони) режиму введення (відображати введені символи або відображати зірочки) та кнопку «ОК». Вивести введенний пароль при натисканні на кнопку «ОК» у деяке текстове поле.
----	---

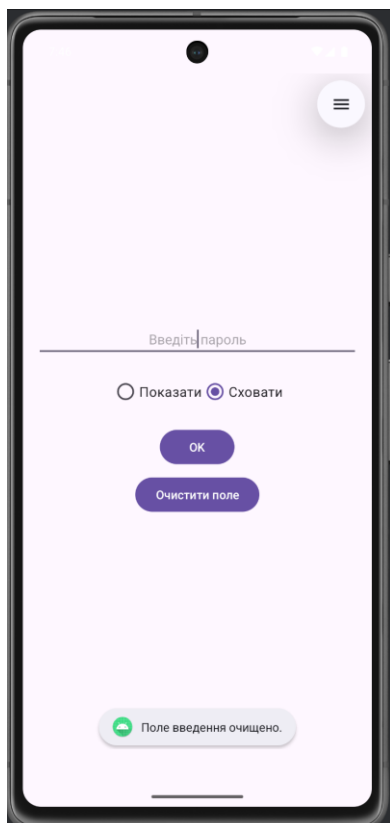
[Посилання на код](#)

За замовчуванням у мене пароль вводиться прихований. При натисканні кнопки «ОК» коли поле вводу пусте відображається спливаюче повідомлення.





Також для зручності я додала кнопку, яка очищує поле вводу



КОНТРОЛЬНІ ПИТАННЯ

1. Архітектура застосунку під платформу Андроїд.

Архітектура Android додатків базується на фреймворку Android, що надає класи та інтерфейси для розробки. Вона має унікальну систему управління пам'яттю, підтримує шаблон MVVM та використовує XML для опису інтерфейсу.

2. Загальний огляд компонентів застосунку під Андроїд.

Activity (Діяльність) – представляє один екран інтерфейсу.

Service (Служба) – працює у фоні без інтерфейсу.

Content Provider (Контент-провайдер) – дає доступ до даних між застосунками.

Broadcast Receiver (Приймач широкомовних намірів) – реагує на системні або користувацькі повідомлення.

Intent (Намір) – механізм комунікації між компонентами.

3. Життєвий цикл компоненту «Діяльність».

Метод onCreate() викликається один раз при створенні Діяльності (проводиться початкове налаштування інтерфейсу, зв'язування даних, налаштування служб та потоків, тощо).

Метод onStart() викликається для переходу Діяльності до взаємодії з користувачем (стає видимою).

Метод onResume() передає весь фокус введення Діяльності (для ініціалізації компонентів, реєстрації будь-яких широкомовних приймачів або інших процесів).

Метод onPause() викликається, коли користувач вирішує перейти до роботи з новим вікном (зупинити анімацію та інші дії, які завантажують процесор; зафіксувати не збережені дані; звільнити системні ресурси, наприклад, обробку даних від GPS).

Метод onStop() викликається, коли вікно стає невидимим для користувача (якщо була запущена інша Діяльність, що перекрила вікно поточної активності; можна зберігати дані: для призупинення складної анімації, потоків, відстеження показань датчиків, запитів до GPS, таймерів, служб

або інших процесів, які потрібні виключно для оновлення інтерфейсу користувача).

Метод `onRestart()` викликається після того, як Діяльність знову була запущена користувачем (для спеціальних дій, які повинні виконуватися лише при повторному запуску).

Метод `onDestroy()` викликається після закінчення роботи. Діяльність (видаляє всі статичні дані активності. Віддає/звільняє всі використовувані ресурси).

4. Життєвий цикл компоненту «Служба».

Якщо компонент запускає Службу викликом методу `startService()` (який у результаті призведе до виклику методу `onStartCommand()`), то Служба буде працювати, доки не зупинить сама себе методом `stopSelf()`, або поки інший компонент не зупинить її викликом методу `stopService()`.

Якщо компонент викликав метод `bindService()` для створення Служби (і метод `onStartCommand()` не викликався), то Служба працює, поки хоча б один компонент залишився прив'язаним до неї (як тільки прибрані всі прив'язки клієнтів до Служби, система її знищить).

5. Опис процесів платформи Андроїд.

Активний процес (процес переднього плану) – це процес програми:

- з яким користувач взаємодіє в даний момент;
- виконує Службу, пов'язану з Діяльністю, з якою взаємодіє користувач;
- що містить Службу та виконує метод зворотного виклику, який визначений для цієї служби;
- що містить Приймач широкомовних намірів і виконує його метод зворотного виклику для прийому наміру.

Видимий процес - це процес, який:

- має Діяльність, видиму кінцевому користувачеві в даний момент часу (діяльність втратила фокус введення, але ще видна користувачеві);

- має Службу, пов'язану на даний момент з Діяльністю, яка перебуває на передньому плані (або частково перекрита).

Сервісний (службовий) процес - це процес, що містить, виконувану на даний момент Службу, яка не відноситься до попередніх типів.

Фоновий процес - це процес, що містить Діяльність, яку не видно користувачеві (є безліч фонових процесів, робота яких завершується за принципом "останній запущений закривається останнім").

Порожній процес - це процес, який не містить жодних активних компонентів програми і використовується як «кеш» для зменшення витрат під час виклику компонента.

6. Яким чином активуються компоненти застосунку.

Компоненти активуються через **Intent (Намір)**. Є два види Intent:

Явний (Explicit Intent) – викликає конкретний компонент.

Неявний (Implicit Intent) – використовується для виклику компонентів інших застосунків.

7. Призначення файлу маніфесту та його структура.

Файл маніфесту AndroidManifest.xml

Призначення: надає системі основну інформацію про програму:

- визначає ім'я пакета програми (унікальний ідентифікатор для програми);
- описує компоненти програми Activities, Services, Broadcast Receivers та Content Providers (визначає імена класів, що реалізують кожен із компонентів та оголошує можливості);
- містить список необхідних дозволів для звернення до захищених частин API та взаємодії з іншими програмами;
- оголошує дозволи, які сторонні додатки повинні мати для взаємодії з компонентами цієї програми;
- оголошує мінімальний рівень API Android, необхідний для роботи додатку;
- перераховує зв'язані бібліотеки.

Для визначення компонентів використовуються:

<activity> для Activity (діяльності)

<service> для Service (служби)

<receiver> для Broadcast Receiver (приймача широкомовних повідомлень)

<provider> для Content Providers (постачальники даних)

8. Поняття ресурсу та яким чином визначаються ресурси.

Використання ресурсів дає можливість змінювати деякі частини програми без модифікації вихідного коду, а також дозволяє оптимізувати програму для різних пристроїв (з різною мовою інтерфейсу або розміром екрану)

Типи ресурсів:

- Зображення;
- Шари GUI (XML файли);
- Оголошення меню (XML файли);
- Текстові рядки.