

# ВИПУСКНИЙ КВАЛІФІКАЦІЙНИЙ ПРОЕКТ

на тему:

## «Розробка Android-додатку для імітації тачпаду ПК»

Студента 2м курсу, 6 групи,  
спеціальності 121 «Інженерія  
програмного забезпечення»  
спеціалізації «Інженерія  
програмного забезпечення»

\_\_\_\_\_

підпис студента

Шевченко  
Юлія Олександрівні

Науковий керівник  
доктор технічних наук,  
професор, завідувач кафедри  
інженерії програмного  
забезпечення та кібербезпеки

\_\_\_\_\_

підпис керівника

Криворучко Олена  
Володимирівна

Гарант освітньої програми  
доктор технічних наук,  
професор, завідувач кафедри  
інженерії програмного  
забезпечення та кібербезпеки

\_\_\_\_\_

підпис гаранта

Криворучко Олена  
Володимирівна

# Київський національний торговельно-економічний університет

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення та кібербезпеки

Освітній ступінь магістр

Спеціальність 121 «Інженерія програмного забезпечення»

**Затверджую**

Зав. кафедри інженерії програмного  
забезпечення та кібербезпеки

Криворучко О. В.

8 жовтня 2019 р.

## **Завдання**

### **на випускний кваліфікаційний проект студентів**

Шевченко Юлія Олександрівні

(прізвище, ім'я, по батькові)

1. Тема випускного кваліфікаційного проекту «Розробка Android-додатку для імітації тачпаду ПК»

Затверджена наказом ректора від "07" листопада 2019 р. № 185

2. Строк здачі студентом закінченої проекту 01 грудня 2020

3. Цільова установка та вихідні дані до проекту

Мета проекту усунення ризиків, пов'язаних з уразливістю комп'ютерної мишки шляхом створення її альтернативи у вигляді Android-додатка, що імітуватиме тачпад.

Об'єкт дослідження процес розробки мобільного додатка

Предмет дослідження розробка мобільного додатка для імітації тачпаду ПК на платформі Android

4. Консультанти проекту із зазначенням розділів, які консультують:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв

5. Зміст випускного кваліфікаційного проекту (перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ЗАСТОСУВАННЯ МОБІЛЬНОГО ДОДАТКУ

1.1. Характеристика роботи комп'ютерних маніпуляторів

1.2. Обґрунтування вибору ОС Android

1.3. Середовище розробки Android Studio

1.4. Висновок до розділу 1

РОЗДІЛ 2. ВИБІР ПРОГРАМНИХ ЗАСОБІВ ТА ПРОЕКТУВАННЯ МОБІЛЬНОГО ДОДАТКУ

2.1. Мова програмування Java

2.2. Розмітка XML

2.3. Gradle

2.4. Архітектура додатку

2.5. Висновок до розділу 2

РОЗДІЛ 3. РЕАЛІЗАЦІЯ МОБІЛЬНОГО ДОДАТКУ

3.1. Перелік програмних модулів

3.2. Моделювання руху мишки

3.3 Інтерфейс та функціонал додатку

3.4. Висновок до розділу 3

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТКИ





## АНОТАЦІЯ

Відповідно до мети дослідження робота присвячена розробці мобільного додатка для імітації тачпада ПК, що дозволить уникнути ризиків у випадках несправності комп'ютерної мишки.

В результаті порівняльного аналізу аналогічних рішень визначено вагомі недоліки подібних Android-додатків, що були прийняті до уваги та усунені у нашій розробці.

Розробка серверної частини виконана у середовищі розробки IntelliJ IDEA, а клієнтської – в Android Studio. Обрана мова програмування в обох випадках – Java. Окремі елементи мобільного додатка розроблені із використанням XML, відповідно до стандартів розробки Android-додатків. Збирання проектів виконувалося за допомогою інструмента автоматизації Gradle.

Готовий програмний комплекс MobileMouse було успішно протестовано відповідно до функціональних вимог.

**Ключові слова:** Android, мобільний додаток, комп'ютерна миша, тачпад, клавіатура.

## ABSTRACT

According to the purpose the research is devoted to developing a mobile app to simulate a PC's touchpad, which will help to avoid the risk in case of computer mouse malfunctions.

Comparative analysis of similar solutions identified major shortcomings applications that were taken into account and eliminated in our solution.

The server side development was done in IDE IntelliJ IDEA and the client partition – in Android Studio. The chosen programming language in both cases is Java. Separate elements of the mobile application were designed using XML, in

accordance with the standards of Android applications development. Projects were built using the Gradle automation tool.

The MobileMouse software package was successfully tested according to functional requirements.

**Keywords:** Android, mobile application, computer mouse, touchpad, keyboard.

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ПЗ – програмне забезпечення

ПК – персональний комп'ютер

ОС – операційна система

RAM (англ. Random Access Memory) – оперативна пам'ять, об'єм ОЗП

USB (англ. Universal Serial Bus) – універсальний послідовний інтерфейс для з'єднання комп'ютерів і периферійних пристроїв

IDE (англ. Integrated Development Environment) – інтегроване середовище розробки

SDK (англ. Software Development Kit) – набір із засобів розробки, утиліт і документації до певної технології або платформи

АРК (англ. Application Package Kit) – формат архівних виконуваних файлів-додатків для Android

XML (англ. Extensible Markup Language) – розширювана мова розмітки

					<i>КНТЕУ 121 063-19.МР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
Зав. каф.		Криворучко О.В.		19.10.20	Розробка Android- додатку для імітації тачпаду ПК	<i>Стадія</i>	<i>Арку</i>	<i>Аркушів</i>
Керівник		Криворучко О.В.		19.10.20		<i>ПС</i>	2	50
Гарант		Криворучко О.В.		19.10.20		<i>Факультет інформаційних технологій 2м курс, 6з група</i>		
Розробив		Шевченко Ю.О.		19.10.20				
					<i>Перелік умовних скорочень</i>			



## ЗМІСТ

<b>ВСТУП.....</b>	<b>3</b>
<b>РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ЗАСТОСУВАННЯ МОБІЛЬНОГО ДОДАТКУ .....</b>	<b>6</b>
1.1. Характеристика роботи комп'ютерних маніпуляторів .....	6
1.2. Обґрунтування вибору ОС Android .....	10
1.3. Середовище розробки Android Studio.....	14
1.4 Висновки до розділу 1.....	20
<b>РОЗДІЛ 2 ВИБІР ПРОГРАМНИХ ЗАСОБІВ ТА ПРОЕКТУВАННЯ МОБІЛЬНОГО ДОДАТКУ .....</b>	<b>21</b>
2.1. Мова програмування Java.....	21
2.2. Розмітка XML .....	23
2.3. Gradle .....	27
2.4. Архітектура додатку.....	30
2.5. Висновки до розділу 2.....	33
<b>РОЗДІЛ 3 РЕАЛІЗАЦІЯ МОБІЛЬНОГО ДОДАТКУ .....</b>	<b>34</b>
3.1. Перелік програмних модулів.....	34
3.2. Моделювання руху мишки .....	40
3.3. Інтерфейс та функціонал додатку .....	43
3.4. Висновок до розділу 3 .....	45
<b>ВИСНОВКИ ТА ПРОПОЗИЦІЇ.....</b>	<b>46</b>
<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....</b>	<b>47</b>
<b>ТЕХНІЧНЕ ЗАВДАННЯ .....</b>	<b>51</b>
<b>ТЕСТУВАННЯ ДОДАТКА MOBILEMOUSE.....</b>	<b>54</b>

					<i>КНТЕУ 121 063-19.МР</i>			
Зм.	Аркуш	№ докум.	Підпис	Дата				
Зав. каф.		Криворучко О.В.		19.10.20	Розробка Android- додатку для імітації тачпаду ПК	Стадія	Арку	Аркушів
Керівник		Криворучко О.В.		19.10.20		3	3	50
Гарант		Криворучко О.В.		19.10.20		Факультет інформаційних технологій		
Розробив		Шевченко Ю.О.		19.10.20		2м курс, 6з група		
					Зміст			

## ВСТУП

*Актуальність.* Стрімкий розвиток інформаційних технологій на кінці ХХ - початку ХХІ століття призвів до того, що сьогодні дуже важко знайти людину, яка б щодня не користувалася смартфоном. Смартфон - це мультифункціональний пристрій, що поєднує у собі мобільний телефон і справжній кишеньковий комп'ютер, та дозволяє здійснювати у будь-який момент широкий спектр дій, таких як: вихід в Інтернет, робота з документами різних форматів, користування соцмережами через зручні додатки, онлайн-шопінг, онлайн-банкінг, фото- та відео-зйомка тощо.

Та, незважаючи на це, можливості смартфона є дещо обмеженими порівняно з настільним комп'ютером - перш за все, через різницю у розмірах екрану, по-друге - апаратні конфігурації (потужність процесора та доступний об'єм ОЗП), і по-третє - більшість процесів, що стосуються ручної обробки великих масивів інформації, здійснюються за допомогою комп'ютерного маніпулятора - мишки, і без неї така робота стає достатньо складною, незручною та довготривалою, або навіть неможливою.

Станом на сьогодні мишка є основним комп'ютерним маніпулятором та є на столі у кожного користувача ПК, як для побутових потреб, так і в робочих цілях. Однак, будь-яка мишка, незалежно від типу і технології роботи, має вразливі місця, вплив на які може призвести до її виходу з ладу та спричинити низку негативних наслідків, пов'язані з перериванням процесу роботи за комп'ютером, особливо у випадку термінового проекту. Отже, вважаємо за доцільне спроектувати та розробити альтернативне рішення, що дозволить замінити мишку у випадку її непрацездатності.

					<i>КНТЕУ 121 063-19.МР</i>			
Зм.	Аркуш	№ докум.	Підпис	Дата				
Зав. каф.		Криворучко О.В.		24.01.20	Розробка Android-додатку для імітації тачпаду ПК	Стадія	Аркуш	Аркушів
Керівник		Криворучко О.В.		24.01.20		В	4	50
Гарант		Криворучко О.В.		24.01.20	<i>Вступ</i>	Факультет інформаційних технологій 2м курс, 6з група		
Розробив		Шевченко Ю.О.		24.01.20				

Дослідженням питань архітектури комп'ютера займалися такі вітчизняні науковці як Голотенко О.С., Єфименко В.В., Жихаревич В.В., Закладний О.М., Матвієнко М.П., Мельник А.О., Рассамакін В.Я., Розен В.П. та ін. Зокрема, питанням пристроїв вводу-виводу присвячено праці Проценко С.І., Симотюк І.В., Степаненко О.О., Тищенко К.В., Шпетного І. О. тощо.

Незважаючи на це, питання у розрізі комп'ютерних маніпуляторів, та мишки зокрема, є недостатньо розкритими і потребують більш глибокого дослідження.

Альтернатива комп'ютерній мишці має бути доступною для пересічного користувача ПК, тому нами було прийнято рішення реалізувати її шляхом розробки Android-додатку, що використовуватиме сенсорний екран пристрою у якості тачпада, що й замінить мишку за потреби. Прийняте рішення зумовило визначення об'єкта і предмета дослідження.

*Мета дослідження:* усунення ризиків, пов'язаних з уразливістю комп'ютерної мишки шляхом створення її альтернативи у вигляді Android-додатка, що імітуватиме тачпад.

*Об'єкт дослідження:* процес розробки мобільного додатка.

*Предмет дослідження:* розробка мобільного додатка для імітації тачпада ПК на платформі Android.

У відповідності з метою дослідження поставлені наступні завдання:

- 1) визначити слабкі місця комп'ютерних маніпуляторів для вибору раціонального способу роботи альтернативного рішення;
- 2) дослідити вимоги користувачів комп'ютерної мишки та сформувані функціональний блок технічного завдання на розробку додатка;
- 3) обґрунтувати вибір мобільної операційної системи, яка слугуватиме платформою для майбутнього додатка;
- 4) проаналізувати та обґрунтувати вибір програмного забезпечення та технологій, що використовуватимуться у процесі розробки;

						КНТЕУ 121 06з-19.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			6

5) провести тестування готового рішення на реальних пристроях та розробити інструкцію користувача.

*Методи дослідження*, що були використані у роботі: аналіз, абстрагування, порівняння, графічний метод і моделювання.

*Наукова новизна дослідження* полягає в удосконаленій концепції мобільного додатка, що імітує комп'ютерні маніпулятори, порівняно з аналогічними рішеннями.

*Практичне значення дослідження*: готове альтернативне рішення у вигляді Android-дodatка, що здатний замінити комп'ютерну мишку за потреби, та доступний широкому загалу користувачів ПК.

					<i>КНТЕУ 121 06з-19.МР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		6

## РОЗДІЛ 1

# АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ЗАСТОСУВАННЯ МОБІЛЬНОГО ДОДАТКУ

### 1.1. Характеристика роботи комп'ютерних маніпуляторів

Пристрої введення/виведення – клас пристроїв в типовій архітектурі ПК, що надають комп'ютеру можливість взаємодіяти з зовнішнім світом (з користувачами, а також з іншими комп'ютерами).

До основних сучасних пристроїв введення інформації можна віднести: клавіатура, комп'ютерна миша, голосове введення, сканер тощо [1].

Зважаючи на те, що переважна більшість користувачів ПК у повсякденному житті використовують графічно-орієнтовані операційні системи (на відміну від командних ОС, орієнтованих на управління шляхом введення відповідних команд), ключове місце серед пристроїв вводу займає маніпулятор «комп'ютерна миша».

Перш за все, маніпулятор «миша» призначається для керування положенням курсора на екрані монітора і вибору режимів роботи програми [2]. Однак, комп'ютерна миша є не єдиним маніпулятором ПК та має альтернативи, що з'явилися у процесі еволюції комп'ютерних маніпуляторів.

**Трекбол** – це особливий пристрій для управління вказівником миші, що складається з кульки, зафіксованої за допомогою спеціальних сенсорів, що постійно відслідковують її поворот навколо своєї осі у декількох (частіше двох) напрямках. Кульку можна рухати за допомогою великого пальця, інших

Зм.	Аркуш	№ докум.	Підпис	Дата	<i>КНТЕУ 121 06з-19.МР</i>			
Зав. каф.		Криворучко О.В.		25.06.20	Розробка Android-додатку для імітації тачпаду ПК	Стадія	Аркуш	Аркушів
Керівник		Криворучко О.В.		25.06.20		РІ	7	50
Гарант		Криворучко О.В.		25.06.20		Факультет інформаційних технологій 2м курс, 6 група		
Розробив		Шевченко Ю.О.		25.06.20				
					<i>Аналіз предметної області застосування мобільного додатку</i>			

пальців, а також долоні, тим самим рухаючи вказівник миші на екрані монітора [3].

**Комп'ютерна миша** – пристрій, що здійснює управління курсором на екрані шляхом передачі даних про своє переміщення по робочій поверхні.

Мишка сприймає своє переміщення в робочій площині (зазвичай на частині поверхні стола) і передає цю інформацію комп'ютеру. Програма є в комп'ютері, у відповідь на переміщення миші виконує на екрані дію, яка відповідає напрямку і відстані цього переміщення. В універсальних інтерфейсах (наприклад, у віконному) за допомогою мишки користувач керує спеціальним курсором — вказівником — маніпулятором елементами інтерфейсу. Інколи використовується введення команд мишею без участі видимих елементів інтерфейсу програми: за допомогою аналізування рухів миші. Такий спосіб отримав назву «жести мишкою» (англ. mouse gestures) [4].

У доповнення до датчика переміщення, мишка має від однієї до трьох і більше кнопок, а також додаткові елементи керування (колеса прокрутки (англ. scroll wheel), джойстики, трекболи, клавіші тощо), дії яких зазвичай пов'язані з положенням курсора в цей час.

**Джойстик** (англ. Joystick – дослівно «паличка задоволень») – пристрій введення інформації, котрий представляє собою маніпулятор, за допомогою якого можна задавати екранні координати графічного об'єкта; також може виконувати функції клавіатури [6].

Джойстик представляє собою ручку, нахилом якої можна задавати напрям у двовимірній площині. На ручці, а також у платформі, на якій вона закріплена, зазвичай розташовуються кнопки та перемикачі різного призначення. За рахунок координатних осей X і Y, можлива також зміна

					<i>КНТЕУ 121 06з-19.МР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		8

координат Z, а саме – за рахунок рукоятки навколо осі, наявності другої ручки, додаткового коліщатка тощо.

**Тачпад** (англ. touchpad – сенсорна площадка), сенсорна панель – вказівний пристрій введення, що застосовується найчастіше в ноутбуках [6].

Робота тачпадів заснована на вимірі ємності пальця або ємності між сенсорами. Ємнісні сенсори розташовані вздовж вертикальної та горизонтальної осей тачпада, що дозволяє визначити положення пальця з потрібною точністю.

Оскільки робота пристрою заснована на вимірі ємності, тачпад не буде працювати, якщо водити по ньому яким-небудь непровідним предметом, наприклад, основою олівця. У випадку використання провідних предметів тачпад буде працювати тільки при достатній площі дотику. Вологі пальці ускладнюють роботу тачпада [6].

Тачпади є пристроями із доволі низькою роздільною здатністю. Її достатньо для використання у повсякденній роботі за комп'ютером (офісні додатки, веб-браузери, логічні ігри), та це ускладнює роботу в графічних програмах і робить практично неможливою гру в 3D-шутерах.

Однак у тачпадів є ряд переваг, у порівнянні з іншими маніпуляторами [6]:

- не потребують рівної поверхні;
- не потребують великого простору;
- розташування тачпада фіксоване відносно клавіатури;
- для переміщення курсора через весь екран достатньо лише невеликого переміщення пальця;
- робота з ними не потребує особливого звикання, як наприклад, у випадку з трекболом.

					<i>КНТЕУ 121 06з-19.МР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		8

За допомогою одного тачпада (не торкаючись кнопок) можна здійснювати частково маніпуляції лівої кнопки миші:

- короткий дотик – клацання;
- подвійний короткий дотик – подвійне клацання;
- незавершений подвійний дотик з наступним переміщенням – переміщення об'єкта або виділення;
- окремі ділянки тачпада (полоса справа і зверху/знизу) можуть бути використані для вертикальної і горизонтальної прокрутки.

Отже, в результаті дослідження еволюції комп'ютерних маніпуляторів було виявлено, що кожному із них притаманні власні переваги та недоліки, та усі користуються попитом і у наш час, якщо не у повсякденному житті, то у вузьких сферах професійної діяльності (як, наприклад, трекбол). Що стосується маніпуляторів щоденного вжитку, то ними є мишка і тачпад (у портативних пристроях за відсутності мишки).

Мобільний додаток, розроблений у результаті даного дослідження, надасть змогу користувачеві смартфона перетворити свій мобільний пристрій у сучасний маніпулятор – тачпад, та здійснювати за його допомогою усі базові дії, якими наділений традиційний тачпад.

Виходячи з вищезазначених недоліків комп'ютерної мишки нами був обраний спосіб з'єднання ПК із смартфоном – локальна мережа. Тобто, комп'ютер і мобільний пристрій користувача мають під'єднатися до однієї локальної мережі, щоб додаток мав змогу передавати відповідні команди на ПК.

					<i>КНТЕУ 121 06з-19.МР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		8



## 1.2. Обґрунтування вибору ОС Android

Як і будь-який ПК, кожен смартфон має операційну систему, завдяки якій виконує широкий спектр своїх функцій, що вже давно не обмежуються голосовими викликами та текстовими повідомленнями.

Станом на кінець 3-го кварталу 2019 року, питома вага найпоширеніших мобільних ОС склала: Android – 85,23%, iOS – 10,63%, KaiOS – 4,13% та Windows – 0,01% [11]. Як бачимо, ОС Android займає провідне місце серед мобільних операційних систем. Розглянемо кожен з них та з'ясуємо, які фактори вплинули на вибір ОС користувачами.

Останньою версією мобільної ОС Windows від компанії Microsoft є **Windows 10 Mobile**. На відміну від Android та iOS дана операційна система не набула поширеного використання та вже 10 грудня 2019 року вийдуть останні оновлення системи, після чого її підтримка буде повністю припинена. Microsoft рекомендує усім користувачам переходити на гаджети з Android та iOS і по можливості не зволікати, оскільки сервіси будуть поступово вимикатися. Через 12 місяців після припинення підтримки, тобто у грудні 2020 року Windows 10 Mobile припинить своє існування, а всі пристрої з нею стануть непотрібними [12].

**KaiOS** — мобільна операційна система, заснована на Linux, що «поєднує міць смартфонів із доступністю фічерфонів», призначена для кнопочкових мобільних телефонів[3/2]. Операційна система з'явилася у 2017 році та була розроблена KaiOS Technologies [13].

До основних можливостей KaiOS належать: підтримка LTE, GPS, Wi-Fi, додатків на HTML5, а також велика енергоефективність на пристроях без сенсорного екрана і менше споживання пам'яті[14].

					<i>КНТЕУ 121 06з-19.МР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		8

Деякі сервіси передвстановлені як HTML5-додатки, включаючи Twitter, Facebook і YouTube. KaiOS може працювати з пристроями, у яких розмір оперативної пам'яті складає 256 МБ та більше. Крім того, є магазин додатків, що дозволяє користувачам завантажувати нові [15].

Вищенаведене зовсім не є недоліками KaiOS, а свідчить про орієнтацію даної ОС на окрему категорію користувачів смартфонів, що або не хочуть користуватися сенсорним управлінням, або є невибагливими до можливостей свого мобільного пристрою і користуються ним переважно у традиційних цілях.

**iOS** — це власницька мобільна Unix-подібна операційна система від Apple. Розроблена спочатку у 2007 році для iPhone, згодом також вдосконалена для використання на iPad (до літа 2019, коли на конференції Apple WWDC було представлено нову ОС для iPad — iPadOS), iPod Touch та Apple TV (до 9 вересня 2015, коли на спеціальному заході Apple було представлено tvOS). Apple не дозволяє роботу ОС на мобільних телефонах інших фірм [16].

Android – відкрита операційна система для смартфонів, планшетів, електронних книг, цифрових програвачів, розумних годинників, фітнес-браслетів, ігрових приставок, ноутбуків, окулярів Google Glass, телевізорів тощо.

До переваг Android можна віднести [12]:

- Відкритий код – будь-який кваліфікований програміст може написати додаток і розмістити його в офіційному магазині Android Play Market. Крім того, власні магазини мають і виробники смартфонів.
- Зручність передачі даних – якщо користувач бажає завантажити у пам'ять смартфона улюблену пісню або відео, йому буде достатньо з'єднати

					<i>КНТЕУ 121 06з-19.МР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		8

гаджет з комп'ютером за допомогою USB-кабелю та перенести файл з однієї теки в іншу. Необхідності «воювати» з додатковим софтом немає.

- Ціна – можна знайти смартфон на Android за ціною лише в 1 тис. гривень. Саме той факт, що Android-гаджети доступні як пенсіонерам, так і студентам, є причиною тотального домінування компанії на ринку ОС.
- Кастомізація – можливостей «заточити» інтерфейс Android під свої потреби чимало. Деякі варіанти настільки видозмінюють зовнішній вигляд ОС, що впізнати у ній першоджерело досить важко.

З іншого боку, такі риси ОС Android як відкритість коду та менш жорсткі вимоги до публікації додатків у магазині Play Market (наприклад, порівняно з Apple AppStore) стали передумовами для підвищення ризиків ураження операційної системи шкідливим програмним забезпеченням

Крім того, Android-у притаманні й інші недоліки, що створюють певні незручності у користувачів [19]:

1. *Google стежить за вами* – якщо ви не міняєте стандартні настройки аккаунта, Google отримує відомості про те, кому ви телефонуєте і пишете СМС-повідомлення, що ви шукаєте, на які сайти заходите і які програми встановлюєте. Компанія легко може дізнатися, куди ви ходите і куди можете піти в майбутньому. Вся ця інформація дуже корисна для рекламодавців, а Google отримує дохід в першу чергу саме з реклами.

2. *Смартфони забиті встановленими програмами* – попередньо відеосервіси, ігри, офісні пакети та інші інструменти можуть бути корисні, але часто їх неможливо видалити. Вони займають дорогоцінну пам'ять, і часом через це пристрій працює повільніше, ніж міг би.

3. *У системних оновлення панує хаос* – іноді оновлення для Android з'являються регулярно, але частіше за все з ними діється повна плутанина. А коли виходить велике оновлення, до більшості смартфонів воно часом так і

					<i>КНТЕУ 121 06з-19.МР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		8

не добирається, так як нові версії зазвичай доступні тільки для більш-менш нових пристроїв.

Ще одна серйозна проблема – брак оновлень систем безпеки. Вони виходять не так часто, як хотілося б, тому навіть у відносно нових смартфонах залишаються уразливості, які вже давно були виявлені і виправлені.

4. *Інтерфейс Android весь час змінюється* – наприклад, кілька років тому, випустивши Android Lollipop, Google увела новий матеріальний дизайн. Він актуальний і по цей день, хоч і в зміненому вигляді. Але пізніше в Android Pie компанія цілком позбулася трьох віртуальних кнопок внизу екрану, і зламала звички багатьох користувачів.

5. *Від сервісів Google занадто багато залежить* – тепер смартфон на Android без купи додатків від Google, для яких обов'язково потрібен відповідний аккаунт, – це щось з області фантастики. І навіть якщо ви завантажуєте програму від стороннього розробника, то, цілком ймовірно, для його використання вам потрібен обліковий запис Google.

6. *Все менше компонентів Android має відкритий вихідний код* – коли ОС Android тільки починала розвиватися, її лаунчер, браузер, карти та інші програми мали відкритий вихідний код. Це дозволяло стороннім розробникам модифікувати їх, як заманеться, і робити кориснішими. Тепер Google не дозволяє видозмінювати свої сервіси.

Звичайно, останній недолік є таким лише з суб'єктивної сторони, адже тенденція зменшення відкритості коду компонентів Android позитивно впливає на безпеку та захищеність ОС, оскільки хакери вже не матимуть того обсягу інформації про систему Android, який був раніше у вільному доступі.

					<i>КНТЕУ 121 06з-19.МР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		8

### 1.3. Середовище розробки Android Studio

Android Studio — інтегроване середовище розробки (IDE) для платформи Android, представлене 16 травня 2013 року на конференції Google I/O. Середовище побудоване на базі вихідного коду продукту IntelliJ IDEA Community Edition, що розвивається компанією JetBrains. Android Studio розвивається в рамках відкритої моделі розробки та поширюється під ліцензією Apache 2.0[24].

Android Studio прийшло на зміну плагіну ADT для платформи Eclipse. Для застосунків, спочатку розроблених з використанням Eclipse і ADT Plugin, підготовлений інструмент для автоматичного імпорту існуючого проекту в Android Studio.

Середовище розробки адаптоване для виконання типових завдань, що вирішуються в процесі розробки застосунків для платформи Android. У тому числі у середовище включені засоби для спрощення тестування програм на сумісність з різними версіями платформи та інструменти для проектування застосунків, що працюють на пристроях з екранами різної роздільності (планшети, смартфони, ноутбуки, годинники, окуляри тощо). Крім можливостей, присутніх в IntelliJ IDEA, в Android Studio реалізовано кілька додаткових функцій, таких як нова уніфікована підсистема складання, тестування і розгортання застосунків, заснована на складальному інструментарії Gradle і підтримуюча використання засобів безперервної інтеграції[25].

Для прискорення розробки застосунків представлена колекція типових елементів інтерфейсу і візуальний редактор для їхнього компоновання, що надає зручний попередній перегляд різних станів інтерфейсу застосунку (наприклад, можна подивитися як інтерфейс буде виглядати для різних

					<i>КНТЕУ 121 06з-19.МР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		8

версій Android і для різних розмірів екрану). Для створення нестандартних інтерфейсів присутній майстер створення власних елементів оформлення, що підтримує використання шаблонів. У середовище вбудовані функції завантаження типових прикладів коду з GitHub.

До складу також включені пристосовані під особливості платформи Android розширені інструменти рефакторингу, перевірки сумісності з минулими випусками, виявлення проблем з продуктивністю, моніторингу споживання пам'яті та оцінки зручності використання. У редактор доданий режим швидкого внесення правок. Система підсвічування, статичного аналізу та виявлення помилок розширена підтримкою Android API. Інтегрована підтримка оптимізатора коду ProGuard. Вбудовані засоби генерації цифрових підписів. Надано інтерфейс для управління перекладами на інші мови.

Деякі особливості будуть пізніше розгорнуті для користувачів так як програмне забезпечення розвивається; наразі, передбачені такі функції[26,27,28]:

- Живі макети (layout): редагувальник WYSIWYG — живе кодування — подання (rendering) програми в реальному часі.
- Консоль розробника: підказки по оптимізації, допомога по перекладу, стеження за напрямком, агітації та акції — метрики Google аналітики.
- Резерви бета релізів та покрокові релізи.
- Базування на Gradle.
- Android-орієнтований рефакторинг та швидкі виправлення.
- Lint утиліти для охоплення продуктивності, юзабіліті, сумісності версій та інших проблем.

					<i>КНТЕУ 121 06з-19.МР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		8

- Використання можливостей ProGuard та підписів до програм.
- Шаблони для створення поширених Android дизайнів та компонентів.
- Багатий редактор макетів (layouts) що дозволяє користувачам перетягнути і покласти (drag-and-drop) компоненти користувацького інтерфейсу, як варіант, переглянути одночасно макети (layouts) на різних конфігураціях екранів.

Актуальна версія Android Studio 3.5 підтримується такими ОС: Windows, Mac, Linux, Chrome OS. Системні вимоги до кожної з них наведені у табл. 1.1 (розроблено на основі [29]).

Таблиця 1.1

Системні вимоги Android Studio 3.5

Windows	Mac	Linux	Chrome OS
Microsoft® Windows® 7/8/10 (32- або 64-розрядна) Android-емулятор підтримується лише 64-бітною Windows RAM: не менше 4 Гб Вільного місця на жорсткому диску: не менше 2 Гб	Mac® OS X® 10.10 (Yosemite) або вище, до 10.14 (macOS Mojave) RAM: не менше 4 Гб Вільного місця на жорсткому диску: не менше 2 Гб	GNOME або KDE настільна версія Протестовано на Ubuntu® 14.04 LTS, Trusty Tahr (64-бітний дистрибутив сумісний із запуском 32-бітних додатків) GNU C Library (glibc) 2.19 та вище RAM: не менше 4 Гб Вільного місця на жорсткому диску: не менше 2 Гб	Рекомендується Intel i5 або вище <u>Рекомендовані пристрої:</u> Acer: Chromebook 13/Spin 13, Chromebox CX13 Lenovo: Yoga C630 Chromebook HP: Chromebook x360 14, Chromebox G2 Dell: Inspiron Chromebook 14 ASUS: Chromebox 3 ViewSonic: NMP660 Chromebox CTL: Chromebox CBx1
Рекомендований обсяг RAM: 8 Гб			
Рекомендований обсяг вільного місця на жорсткому диску: 4 Гб (500 Мб на IDE + 1.5 Гб на Android SDK та емулятор). Роздільна здатність екрану: не менше 1280 x 800			

					<i>КНТЕУ 121 06з-19.МР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		8

Зважаючи на необхідність багаторазового виконання одних і тих самих дій у процесі роботи над проектом, у середовищі Android Studio передбачені комбінації клавіш, що полегшують та прискорюють процес розробки. Найбільш часто вживані «гарячі комбінації клавіш» наведені у таблицях 1.2 та 1.3.

Таблиця 1.2

Комбінації клавіш для розгортання деяких вікон інструментів

Вікно інструмента	Windows та Linux	Mac
Проект (Project)	Alt+1	Command+1
Контроль версій (Version Control)	Alt+9	Command+9
Виконати (Run)	Shift+F10	Control+R
Відлагодити (Debug)	Shift+F9	Control+D
Логи (Logcat)	Alt+6	Command+6
Повернутися до редактора (Return to Editor)	Esc	Esc
Згорнути усі вікна інструментів (Hide All Tool Windows)	Control+Shift+F12	Command+Shift+F12

Якщо ви хочете приховати всі панелі інструментів, вікна інструментів та вкладки редактора, натисніть Перегляд (View) > Увійти у режим без відволікань (Enter Distraction Free Mode). Щоб вийти з нього, необхідно натиснути Перегляд (View) > Вийти з режиму без відволікань (Exit Distraction Free Mode).

Крім виклику вікон інструментів існують також комбінації для різних типів авто-доповнення коду (табл. 1.3).

					<i>КНТЕУ 121 06з-19.МР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		8



## Комбінації клавіш для авто-доповнення коду

Тип	Опис	Windows та Linux	Mac
Звичайне доповнення	Відображає основні пропозиції щодо змінних, типів, методів, виразів тощо. Якщо скористатися Звичайним доповненням двічі поспіль, Ви побачите більше результатів, включаючи пропозиції щодо приватних членів та неімпортованих статичних членів.	Control + Space	Control + Space
Розумне доповнення	Відображає відповідні параметри залежно від контексту. Розумне доповнення знає очікувані типи і потоки даних.	Control + Shift + Space	Control + Shift + Space
Доповнення виразів	Завершує даний вираз за Вас, додаючи відсутні дужки, фігурні дужки, форматування тощо.	Control + Shift + Enter	Shift + Command + Enter

Також можна викликати швидкі виправлення (Quick fixes) та відобразити Цільові дії (Intention actions) шляхом натискання *Alt+Enter*.

Розпочати роботу із проектом Android-додатку дуже просто. Для цього необхідно лише обрати відповідну дію у стартовому вікні (рис. 1.1).

Після відкриття або створення проекту IDE дозволяє переглядати та працювати із компонентами у декількох розрізах, обираючи їх у лівому вертикальному меню: Проект (Project), Менеджер ресурсів (Resource Manager), Знімки макетів (Layout Captures), Структура (Structure), Обране (Favorites), Варіанти збірки (Build Variants).

Стандартним режимом перегляду є Проект (Project). Дерево проекту нашого мобільного додатку у даному режимі виглядає наступним чином (рис. 1.2).

						КНТЕУ 121 06з-19.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			8

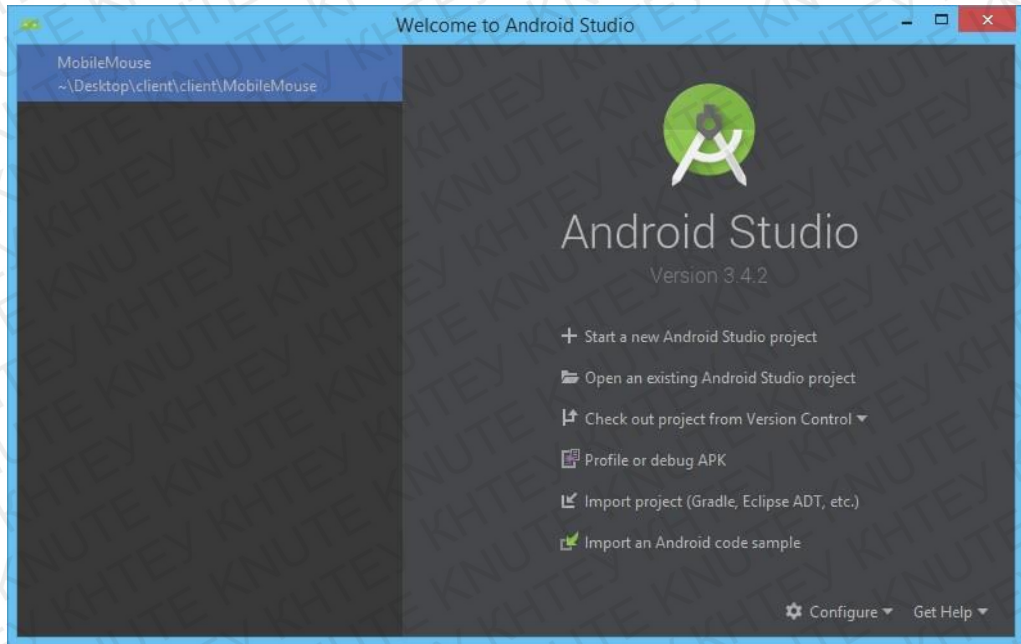


Рис. 1.1. Стартове вікно Android Studio

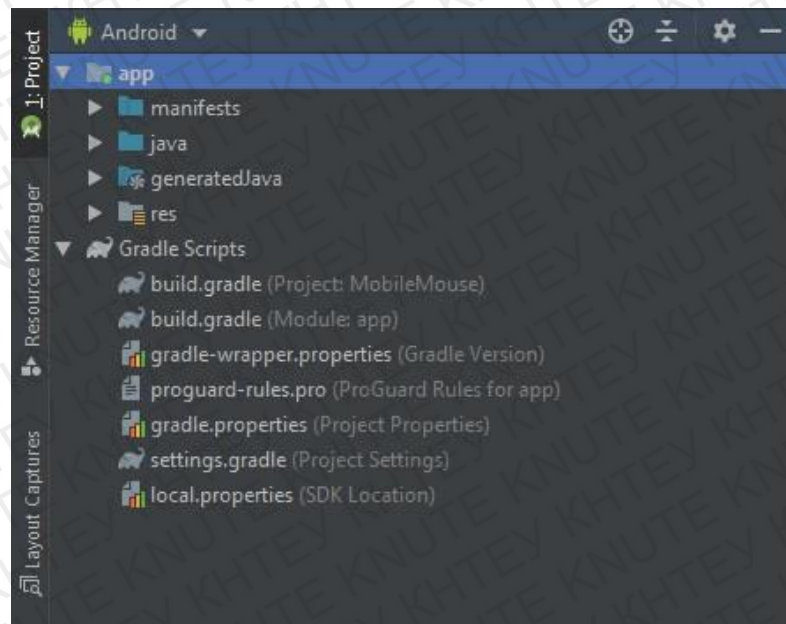


Рис. 2.2. – Дерево проекту MobileMouse

У нижній частині вікна IDE розташовані вкладки для перегляду логів виконання програми при тестуванні на підключеному пристрої (Logcat),

						Аркуш
						8
Зм.	Аркуш	№ докум	Підпис	Дата		

*КНТЕУ 121 06з-19.МР*

згенерованих автоматично та вручну поміток про необхідність доопрацювання елементів проекту (TODO), роботи у вбудованому терміналі (Terminal) та моніторингу даних про виконання збірки (Build).

#### 1.4 Висновки до розділу 1

Отже, основне меню та інтерфейс Android Studio є досить легким в опануванні для розробників, що вже раніше користувалися різноманітними IDE у процесі роботи із кодом, та, з іншого боку, є інтуїтивно-зрозумілим для новачків у використанні IDE.

					<i>КНТЕУ 121 06з-19.МР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		8

## РОЗДІЛ 2

### ВИБІР ПРОГРАМНИХ ЗАСОБІВ ТА ПРОЕКТУВАННЯ МОБІЛЬНОГО ДОДАТКУ

#### 2.1. Мова програмування Java

На сьогодні Android-додатки можливо створювати, використовуючи найрізноманітніші мови програмування (Java, Kotlin, C#, Python та ін.). Проте, офіційна IDE від Google – Android Studio – підтримує Java, C/C++ та Kotlin-код, і це означає, що зазначені мови програмування є рекомендованими для створення Android-додатків.

Доведено, що мова C/C++ є більш доцільною у використанні лише для розробки окремих модулів програм, що мають справу з високою ресурсоемністю. Цілковите ж застосування даної мови програмування при створенні додатку є досить складним та важким у налаштуванні.

Kotlin є «наймолодшою» мовою програмування в Android-розробці, що поки не набула великого поширення. Оскільки вона ще розвивається, ми надали перевагу при створенні мобільного додатку для імітації тачпада мові Java.

Під «незалежністю від архітектури» мається на увазі те, що програма, написана на мові Java, працюватиме на будь-якій підтримуваній апаратній чи системній платформі без змін у початковому коді та перекомпіляції.

Цього можна досягти, компілюючи початковий Java код у байт-код, який є спрощеними машинними командами. Потім програму можна виконати на будь-якій платформі, що має встановлену віртуальну машину Java, яка інтерпретує байткод у код, пристосований до специфіки конкретної

Зм.	Аркуш	№ докум.	Підпис	Дата	<i>КНТЕУ 121 06з-19.МР</i>			
Зав. каф.		Криворучко О.В.		07.09.20	Розробка Android-додатку для імітації тачпаду ПК	Стадія	Аркуш	Аркушів
Керівник		Криворучко О.В.		07.09.20		P2	10	50
Гарант		Криворучко О.В.		07.09.20		Факультет інформаційних технологій		
Розробив		Шевченко Ю.О.		07.09.20		2м курс, 6 група		
					<i>Вибір програмних засобів та проектування мобільного додатку</i>			

операційної системи і процесора. Зараз віртуальні машини Java існують для більшості процесорів і операційних систем.

Стандартні бібліотеки забезпечують загальний спосіб доступу до таких платформозалежних особливостей, як обробка графіки, багатопотоковість та роботу з мережами. У деяких версіях задля збільшення продуктивності JVM байт-код можна компілювати у машинний код до або під час виконання програми.

Основна перевага використання байт-коду — це портативність. Тим не менш, додаткові витрати на інтерпретацію означають, що інтерпретовані програми будуть майже завжди працювати повільніше, ніж скомпільовані у машинний код, і саме тому Java одержала репутацію «повільної» мови. Проте, цей розрив суттєво скоротився після введення декількох методів оптимізації у сучасних реалізаціях JVM.

Одним із таких методів є just-in-time компіляція (JIT, що перетворює байт-код Java у машинний під час першого запуску програми, а потім кешує його. У результаті така програма запускається і виконується швидше, ніж простий інтерпретований код, але ціною додаткових витрат на компіляцію під час виконання. Складніші віртуальні машини також використовують динамічну рекомпіляцію, яка полягає в тому, що віртуальна машина аналізує поведінку запущеної програми й вибірково рекомпілює та оптимізує певні її частини. З використанням динамічної рекомпіляції можна досягти більшого рівня оптимізації, ніж за статичної компіляції, оскільки динамічний компілятор може робити оптимізації на базі знань про довкілля періоду виконання та про завантажені класи. До того ж він може виявляти так звані гарячі точки (англ. hot spots) — частини програми, найчастіше внутрішні цикли, які займають найбільше часу при виконанні. JIT-компіляція та динамічна рекомпіляція збільшує швидкість Java-програм, не втрачаючи при цьому портативності.

					<i>КНТЕУ 121 06з-19.МР</i>	Аркуш
						22
Зм.	Аркуш	№ докум	Підпис	Дата		

## 2.2. Розмітка XML

Розширювана мова розмітки (англ. Extensible Markup Language, скорочено XML) — запропонований консорціумом World Wide Web Consortium (W3C) стандарт побудови мов розмітки ієрархічно структурованих даних для обміну між різними застосунками.

Стандарт XML визначає набір базових лексичних та синтаксичних правил для побудови мови описання інформації шляхом застосування простих тегів. Цей формат достатньо гнучкий для того, аби бути придатним для застосування в різних галузях. Іншими словами, запропонований стандарт визначає метамову, на основі якої шляхом запровадження обмежень на структуру та зміст документів визначаються специфічні, предметно-орієнтовані мови розмітки даних[33]. Прикладами мов, заснованих на XML, є: XSLT, XAML, XUL, RSS, MathML, GraphML, XHTML, SVG, а також XML Schema.

За допомогою довідника XML-елементів, що міститься в Android, можна швидко і просто створювати макети користувацького інтерфейсу та вкладені у нього елементи, так само як при створенні веб-сторінок у HTML.

У кожному файлі макету має бути всього один кореневий елемент, у якості якого виступатиме об'єкт Представлення (View) або Група представлень (ViewGroup). Після визначення кореневого елемента можна створювати додаткові об'єкти макета або віджетів у якості дочірніх елементів для поступового формування ієрархії Представлень, що визначатиме макет[34].

Кожен об'єкт View та ViewGroup підтримує свої власні атрибути XML. Деякі атрибути характерні лише для об'єкта View (наприклад, об'єкт TextView підтримує атрибут textSize), однак такі атрибути також наслідуються будь-якими об'єктами View, що наслідують даний клас. Деякі атрибути є загальними для усіх об'єктів View, оскільки вони наслідуються

					КНТЕУ 121 06з-19.МР	Аркуш
						23
Зм.	Аркуш	№ докум	Підпис	Дата		

від кореневого класу View (такі як атрибут id). Будь-які інші атрибути розглядаються як «параметри макета». Такі атрибути описують визначені орієнтації макета для об'єкта View, що задані батьківським об'єктом ViewGroup такого об'єкта.

У будь-якого об'єкта View може бути пов'язаний з ним цілочислений ідентифікатор, що служить для позначення унікальності об'єкта View в ієрархії. Під час компіляції додатка такий ідентифікатор використовується як ціле число, однак ідентифікатор зазвичай присвоюється у файлі XML макета у вигляді рядка в атрибуті id. Даний атрибут XML є загальним для усіх об'єктів View (визначених класом View).

Визначення ідентифікаторів для об'єктів Представлення має важливе значення у створенні об'єкта Відносного макета (RelativeLayout). У Відносному макеті універсальні ідентифікатори застосовуються для розташування Представлень відносно один одного.

Ідентифікатор не обов'язково має бути унікальним в рамках всієї ієрархії, а лише у тій її частині, де виконується пошук (у разі, коли він виконується не по всій ієрархії).

Атрибути макета XML, названі по типу *layout\_something* визначають параметри макета для об'єкта Представлення, доречні для класу ViewGroup, у якому він знаходиться.

Кожен клас ViewGroup реалізує вкладений клас, що наслідує ViewGroup.LayoutParams. У даному підкласі присутні типи властивостей, що визначають розмір і положення кожного дочірнього Представлення, доречні для його групи. На рисунку 2.1. зображено, що батьківська Група представлень визначає параметри макета для кожного дочірнього Представлення (включаючи дочірню Групу представлень).

					КНТЕУ 121 06з-19.МР	Аркуш
						24
Зм.	Аркуш	№ докум	Підпис	Дата		

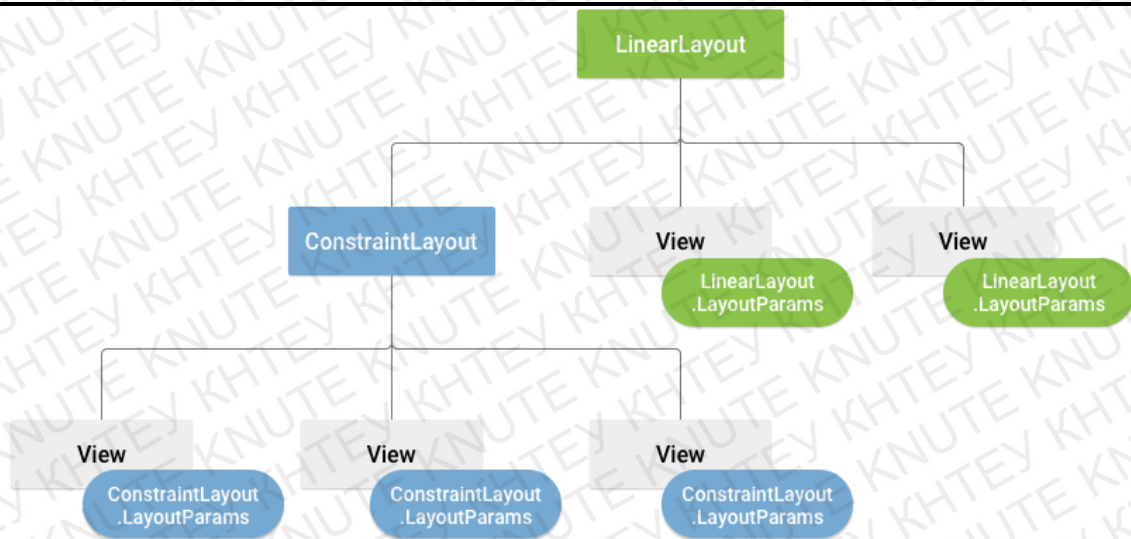


Рис. 2.1. Приклад ієрархії Представлення з параметрами макета [34]

Після оголошення макета у файлі XML необхідно зберегти файл із розширенням .xml в каталозі res/layout/ проекту Android для подальшої компіляції.

Проект додатка MobileMouse включає два макети користувацького інтерфейсу: activity\_main.xml – початковий екран та activity\_board.xml – екран для роботи з імітованим тачпадом.

Макет **activity\_main.xml** описаний наступним чином:

```

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent"
    android:layout_height="match_parent" tools:context=".MainActivity">
    <LinearLayout
        android:orientation="vertical"
        android:layout_centerInParent="true"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">
    <EditText
        android:id="@+id/edit_ip"
        android:hint="server ip"
        android:background="@drawable/phone_border"
        android:layout_width="200dp"
        android:layout_height="wrap_content" />
    <EditText
        android:inputType="numberDecimal"
        android:id="@+id/edit_port"
        android:layout_marginTop="5dp"
        android:hint="port"
        android:background="@drawable/phone_border"
        android:layout_width="200dp"
        android:layout_height="wrap_content" />
    
```

						Аркуш
						25
Зм.	Аркуш	№ докум	Підпис	Дата		



```

<Button
    android:id="@+id/ip_port_sure"
    android:layout_gravity="center_horizontal"
    android:text="Done"
    android:textColor="#ffffff"
    android:layout_marginTop="5dp"
    android:background="@drawable/phone_border1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
</LinearLayout>
</RelativeLayout>

```

Макет **activity\_board.xml** має наступний вміст:

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent">
    <View
        android:id="@+id/touch_board"
        android:background="@drawable/phone_border2"
        android:layout_margin="20dp"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />
    <LinearLayout
        android:layout_alignParentBottom="true"
        android:layout_marginBottom="50dp"
        android:orientation="horizontal"
        android:layout_width="match_parent"
        android:layout_height="80dp">
        <View
            android:layout_marginRight="5dp"
            android:background="@drawable/phone_border3"
            android:layout_weight="1"
            android:id="@+id/btn_left_click"
            android:layout_marginBottom="10dp"
            android:layout_width="match_parent"
            android:layout_height="match_parent" />
        <View
            android:layout_marginLeft="5dp"
            android:background="@drawable/phone_border3"
            android:layout_weight="1"
            android:id="@+id/btn_right_click"
            android:layout_width="match_parent"
            android:layout_marginBottom="10dp"
            android:layout_height="match_parent" />
    </LinearLayout>
    <EditText
        android:id="@+id/edit_board_input"
        android:layout_alignParentBottom="true"
        android:layout_margin="5dp"
        android:textColor="#ffffff"
        android:background="@drawable/phone_border4"
        android:layout_width="match_parent"
        android:layout_height="50dp" />
</RelativeLayout>

```

					Аркуш
					26
Зм.	Аркуш	№ докум	Підпис	Дата	

*КНТЕУ 121 063-19.МР*

### 2.3. Gradle

Автоматизація збирання – етап процесу розробки програмного забезпечення, що полягає в автоматизації широкого спектру задач програмістів у їх повсякденній діяльності.

**Gradle** – система автоматичного збирання, яка далі розвиває принципи, закладені в Apache Ant та Apache Maven і використовує предметно-орієнтовану мову (DSL) на основі мови Groovy замість традиційної XML-подібної форми представлення конфігурації проекту. Для визначення порядку виконання завдань Gradle використовує орієнтований ациклічний граф ("DAG").

У процесі збірки використовуються різноманітні інструменти та процедури, що конвертують проект Android-додатку у Android Application Package (APK).

Типова збірка модуля Android-додатка (рис.2.2) складається з наведених нижче етапів:

1. Компілятори перетворюють вихідний код у DEX файли;
2. APK «пакувальник» комбінує DEX файли з відкомпільованими ресурсами у єдиний APK;
3. APK «пакувальник» підписує APK ключем відладки або релізу, залежно від поточних потреб;
4. Перед створенням фінальної версії APK, «пакувальник» використовує спеціальний інструмент zipalign для оптимізації додатка.

По завершенню збірки отримується APK для відладки або релізний APK додатка, який можна розгорнути, протестувати або надати зовнішнім користувачам.

Gradle також дозволяє вносити зміни до налаштувань у наступних аспектах збірки: Build types, Product flavors, Build variants, Manifest entries, Dependencies, Signing, Code and resource shrinking, Multiple APK support.

					КНТЕУ 121 06з-19.МР	Аркуш
						27
Зм.	Аркуш	№ докум	Підпис	Дата		

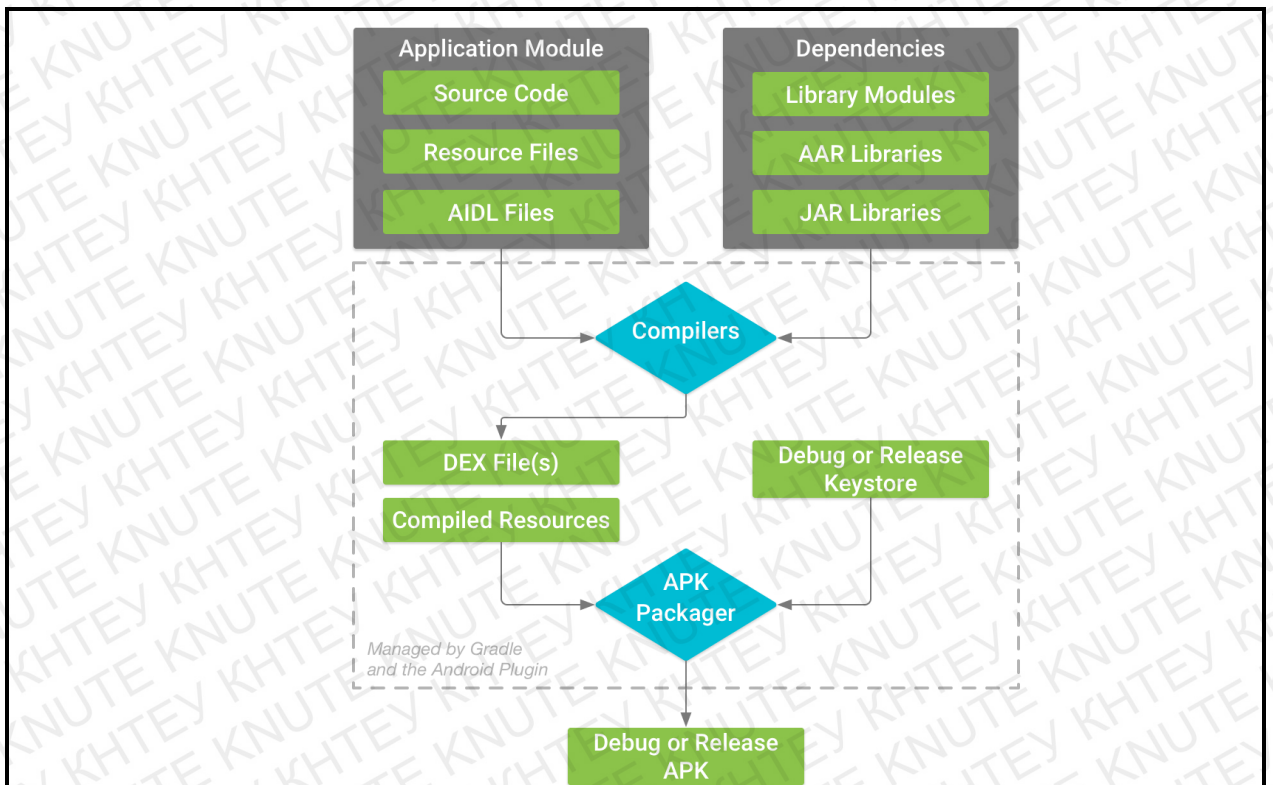


Рис. 2.2. Процес збірки модуля Android-додатка[38]

У випадку нового проекту, Android Studio автоматично створює файли (рис.2.3), та заповнює їх параметрами за замовчуванням.

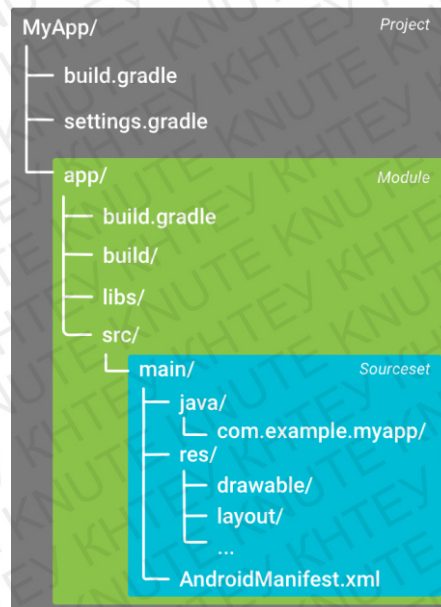


Рис.2.3. Стандартна структура Android-проекту[38]

Певні файли налаштувань Gradle збірки є частиною стандартної структури проекту Android-додатку. До того як вносити зміни до цих

налаштувань, важливо розуміти обсяг і мету кожного із файлів, а також основні DSL-елементи, що вони визначають. Дерево Gradle-файлів мобільного додатка для імітації тачпада зображено на рис.2.4.

Файл `settings.gradle`, розміщений у корені проекту, повідомляє Gradle про модулі, що мають бути включені до збірки.

Файл вищого рівня `build.gradle`, також розміщений у корені проекту, визначає налаштування збірки, що будуть застосовані до усіх модулів проекту. За замовчуванням даний файл використовує скриптовий блок, що визначає сховища та залежності, загальні для усіх модулів.

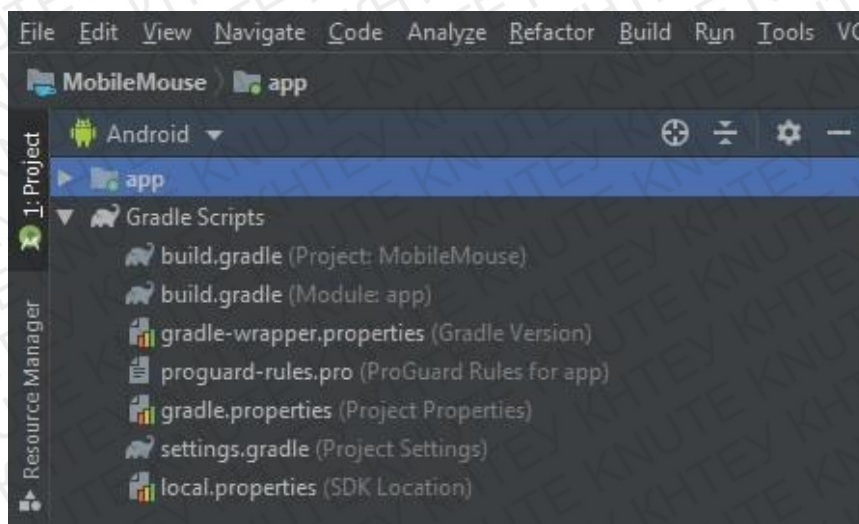


Рис. 2.4. Структура Gradle у проекті MobileMouse

Файл `build.gradle` рівня модуля, розміщений у кожній теці `project/module/`, дозволяє налаштувати збірку особливим чином для конкретного модуля.

Gradle також включає два файли із властивостями, що дозволяють вказати специфічні налаштування для збірки Gradle:

- `gradle.properties` – налаштування загально-проектних параметрів Gradle, таких як максимальний розмір «купи» Gradle тощо;
- `local.properties` – налаштування параметрів локального середовища для системи збірки.

Для синхронізації файлів проекту, можна натиснути Синхронізувати зараз (Sync Now) на панелі повідомлень, що з'являється при внесенні змін, або Синхронізувати проект (Sync Project) на панелі меню (рис. 2.5). Якщо Android Studio повідомить про помилки у поточній конфігурації, з'явиться вікно з повідомленням і описом проблеми.

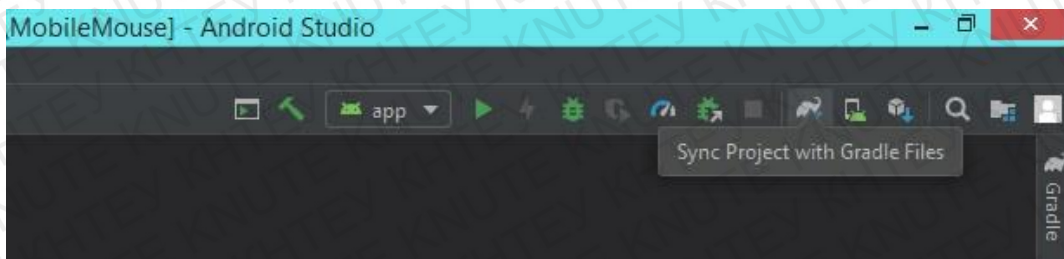


Рис. 2.5. Швидкий запуск синхронізації Gradle

Приклад інформації про результати збірки наведено нижче на рис. 2.6.

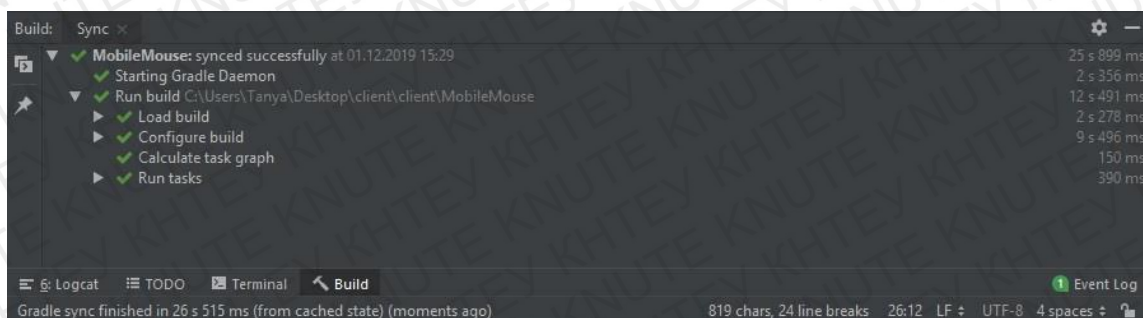


Рис. 2.6. Інформація про виконану збірку Gradle

## 2.4. Архітектура додатку

У більшості випадків, додатки настільних ПК мають єдину «точку входу» та виконуються як один монолітний процес. Android-додатки, з іншого боку, мають більш складну структуру. Типовий Android-додаток містить декілька компонентів, таких як: дії (activities), фрагменти (fragments), служби (services), контент-провайдери (content providers), та ширококомвні приймачі (broadcast receivers).

Перелік компонентів додатка зазначається у його маніфест-файлі. ОС Android використовує даний файл у процесі інтеграції додатка у

					Аркуш
					КНТЕУ 121 06з-19.МР
Зм.	Аркуш	№ докум	Підпис	Дата	30

всеохоплюючий user experience. Зважаючи на те, що правильно написаний додаток для Android містить декілька компонентів і що користувачі часто взаємодіють із декількома додатками у короткий проміжок часу, додатки повинні адаптуватися до різних видів робочих процесів та завдань, керованих користувачем.

Компоненти додатка можна віднести до одного з чотирьох типів: дія, служба, контент-провайдер, широкомовний приймач. Компоненти кожного типу призначені для конкретної мети, мають власний життєвий цикл, що визначає спосіб створення і припинення існування компонента[39].

Коли система запускає компонент, вона запускає процес для відповідного додатка (якщо він ще не був запущений) та створює екземпляри класів, що необхідні для даного компонента. Наприклад, якщо додаток зніціює запуск Дії фотографування у додатку для камери, ця Дія буде виконуватися у процесі, що належить до стороннього додатка, а не у процесі поточного додатка. Тому, на відміну від додатків для більшості інших систем, у додатках для Android відсутня єдина точка входу.

Оскільки система виконує кожний додаток в окремому процесі з такими правами доступу до файлів, що обмежують доступ до інших додатків, додаток не може напряму викликати компонент із іншого додатка. Це може зробити сама система Android. Тому, щоб викликати компонент в іншому додатку, необхідно повідомити систему про свій намір (Intent) запустити конкретний компонент. Після цього система активує вказаний компонент[39].

Компоненти трьох із чотирьох типів — дії, служби та широкомовні приймачі — активуються асинхронним повідомленням, що має назву Intent (намір). Об'єкти Intent пов'язують один з одним окремі компоненти додатків під час виконання.

					КНТЕУ 121 06з-19.МР	Аркуш
						31
Зм.	Аркуш	№ докум	Підпис	Дата		

Об'єкт Наміру створюється за допомогою об'єкта Intent, що описує запит на активацію або конкретного компонента, або компонента конкретного типу — відповідно, Намір може бути явним або неявним.

Для Дій і Служб Намір визначає дію, яку потрібно виконати, а також може вказувати URI (Uniform Resource Identifier – уніфікований ідентифікатор ресурсу) даних, з якими таку дію потрібно виконати (окрім інших відомостей, які необхідно знати компоненту). Наприклад, Намір може передавати запит на виконання Дії «показати зображення» або «відкрити веб-сторінку»[39].

Для широкомовних приймачі Намір визначає повідомлення, що передається (наприклад, повідомлення про низький рівень заряду акумулятора містить лише рядок «акумулятор розряджений»).

Компоненти четвертого типу – контент-провайдери – повідомленнями Наміру не активуються. Вони активуються за запитом від ContentResolver. Процедура визначення контенту (content resolver) обробляє усі прямі транзакції з контент-провайдером, щоб цього не довелося робити компоненту, який виконує транзакції з провайдером. Замість цього він викликає методи для об'єкта ContentResolver. Це формує шар, що абстрагує (в цілях безпеки) контент-провайдера від компонента, що запитує інформацію.

Для запуску компонента додатка системі Android необхідно знати, що компонент існує. Для цього вона зчитує файл AndroidManifest.xml додатка (файл маніфесту). У зазначеному файлі, що має знаходитися у корені додатка, мають бути оголошені усі компоненти додатка[40].

Всі компоненти необхідно оголошувати наступним чином:

- елементи <activity> для дій;
- елементи <service> для служб;
- елементи <receiver> для широкомовних приймачів повідомлень;

						Аркуш
						32
Зм.	Аркуш	№ докум	Підпис	Дата	КНТЕУ 121 06з-19.МР	

- елементи <provider> для контент-провайдерів.

Система не бачить дії, служби та контент-провайдери, наявні у вихідному коді, але не оголошені у файлі маніфесту, тому вони не можуть бути запущені. А от широкомовні приймачі повідомлень можна або оголосити в маніфесті, або створити динамічно в коді (як об'єкти BroadcastReceiver) та зареєструвати в системі шляхом виклику registerReceiver() [40].

### Вміст файлу **AndroidManifest.xml** проекту MobileMouse:

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.dev.mouse" >
    <uses-permission android:name="android.permission.INTERNET"></uses-permission>
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name="com.dev.mouse.MainActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name="com.dev.mouse.BoardActivity">
        </activity>
    </application>
</manifest>
```

## 2.5. Висновки до розділу 2

Отже, підсумовуючи вище викладене, можна визначити, що вбудований в Android Studio інструментарій автоматизації Gradle дозволяє полегшити процес збірки проекту та забезпечує детальною інформацією про хід і результати виконаної збірки. Це дозволяє провести аналіз та, за потреби, виявити та усунути існуючі недоліки. Мобільний додаток імітації тачпада MobileMouse використовує два елементи компонента Activity та один Intent-об'єкт.

						КНТЕУ 121 06з-19.МР	Аркуш
							33
Зм.	Аркуш	№ докум	Підпис	Дата			



## РОЗДІЛ 3

### РЕАЛІЗАЦІЯ МОБІЛЬНОГО ДОДАТКУ

#### 3.1. Перелік програмних модулів

Як вже зазначалося у попередніх розділах, робота мобільного додатка для імітації тачпада ПК побудована на взаємодії двох пристроїв, що виступають один для одного клієнтом (смартфон) та сервером (комп'ютер). Рисунки 3.1 та 3.2 схематично описують механізм роботи додатка та обґрунтовують розробку окремих модулів, що представляють собою серверну частину та клієнтську частину програмного забезпечення.

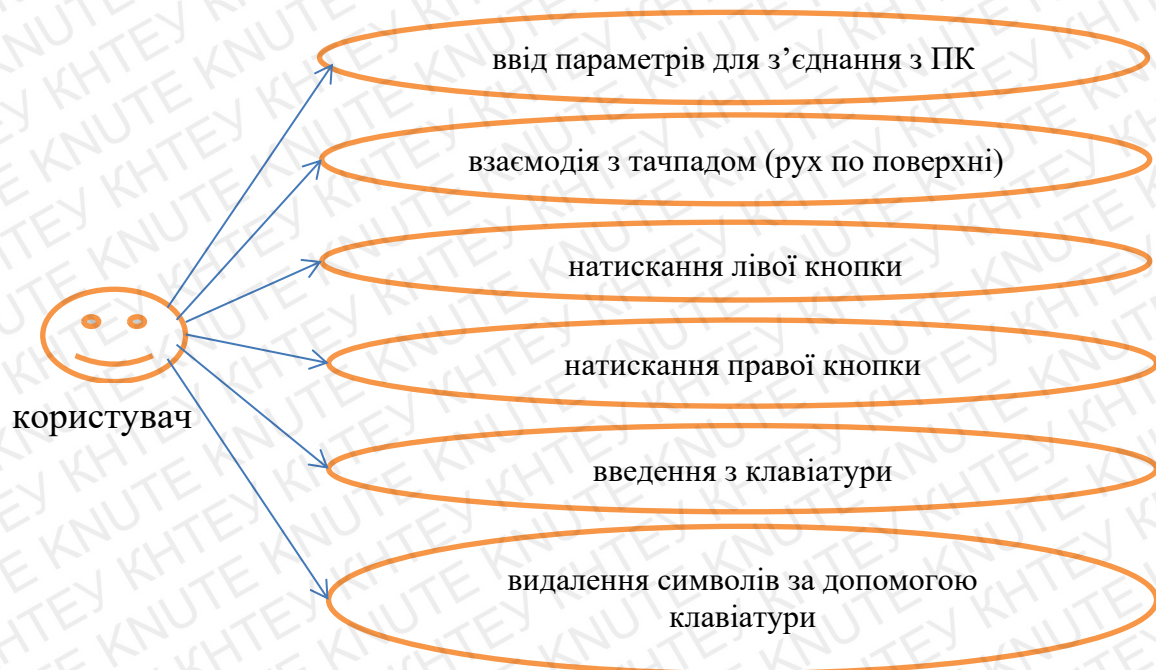


Рис.3.1. Use-cases клієнтської сторони MobileMouse

Зм.	Аркуш	№ докум.	Підпис	Дата	КНТЕУ 121 06з-19.МР			
Зав. каф.		Криворучко О.В.		19.10.20	Розробка Android-додатку для імітації тачпаду ПК	Стадія	Аркуш	Аркушів
Керівник		Криворучко О.В.		19.10.20		РЗ	34	50
Гарант		Криворучко О.В.		19.10.20	Реалізація мобільного додатку	Факультет інформаційних технологій		
Розробив		Шевченко Ю.О.		19.10.20		2м курс, 6 група		

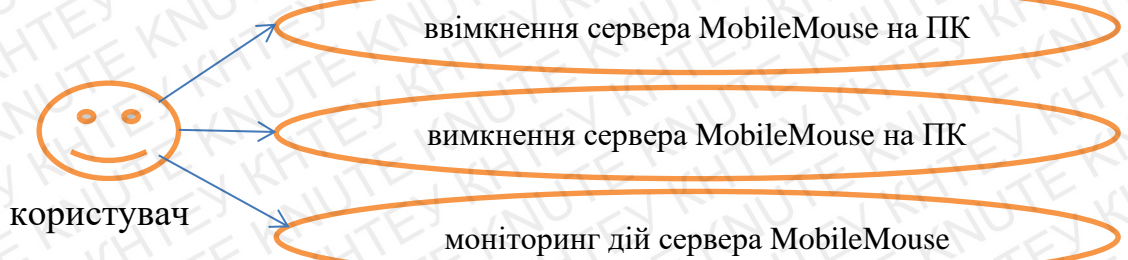


Рис.3.2. Use-cases серверної сторони MobileMouse

Однак, use-cases не надають повної картини роботи програми, адже найважливіші процеси відбуваються між клієнтом та сервером. Розглянемо їх на рис. 3.3.

Виходячи з наведеної вище блок-схеми, бачимо, що мобільний додаток є клієнтською складовою програмного забезпечення для імітації тачпада, зчитує дії користувача та передає їх в обробленій відповідними процедурами формі для подальшої інтерпретації та використання ПК (сервером).

Розглянемо детально модульну структуру обох сторін програми далі.

Перелік набору модулів мобільного додатка MobileMouse зображено на рис. 3.4 та 3.5.

Кожний програмний модуль має своє призначення, що розкрито нижче.

LoadApp – обробляє запуск мобільного додатка на виконання (Додаток А).

MainActivity – модуль роботи з початковим екраном додатка – екраном авторизації (Додаток Б).

BoardActivity – модуль роботи з екраном імітації пристроїв вводу ПК (Додаток В).

TouchPadActivity – модуль роботи з імітованим тачпадом (Додаток Г).

						КНТЕУ 121 06з-19.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			35

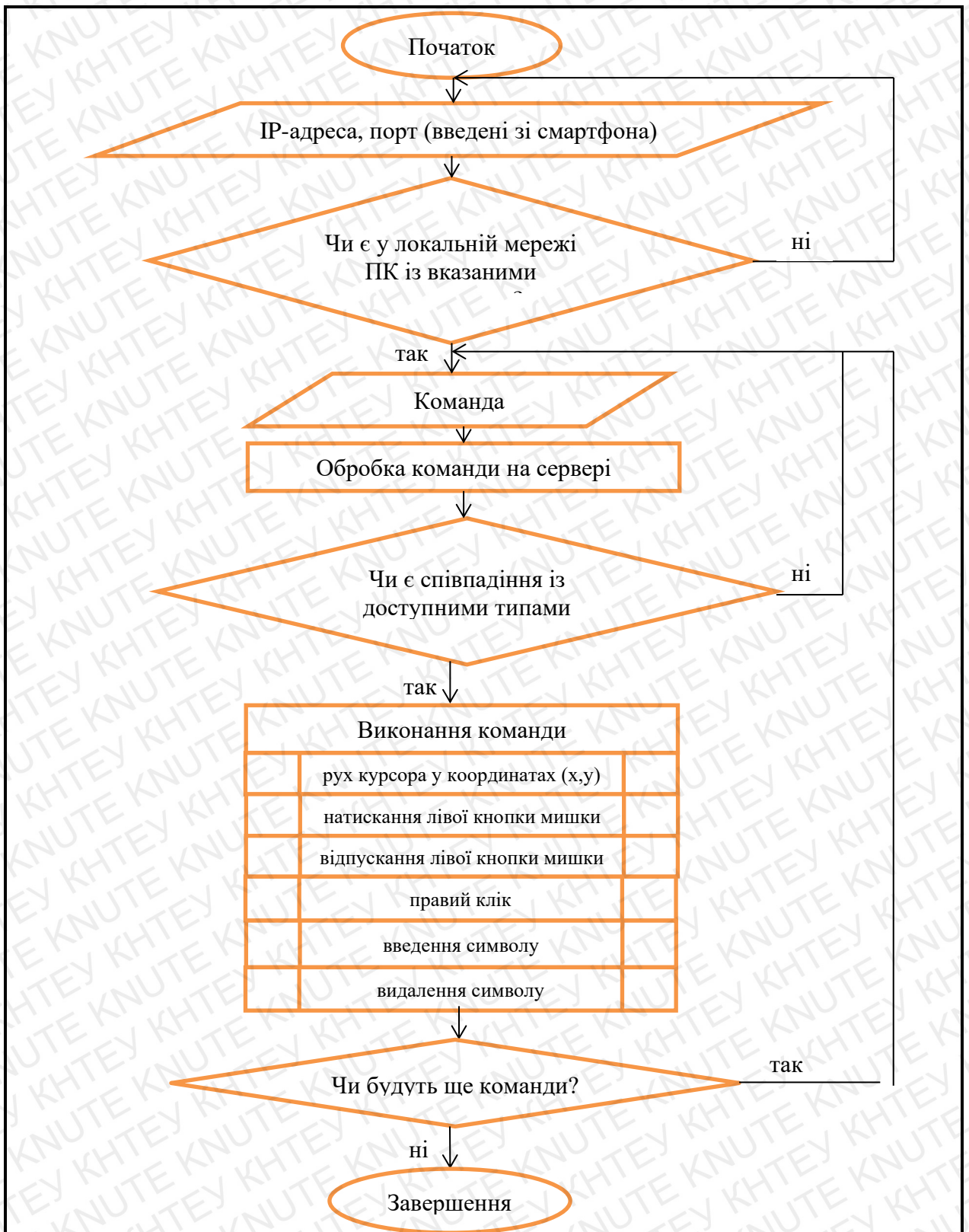


Рис.3.3. Блок-схема взаємодії клієнтської та серверної частин MobileMouse

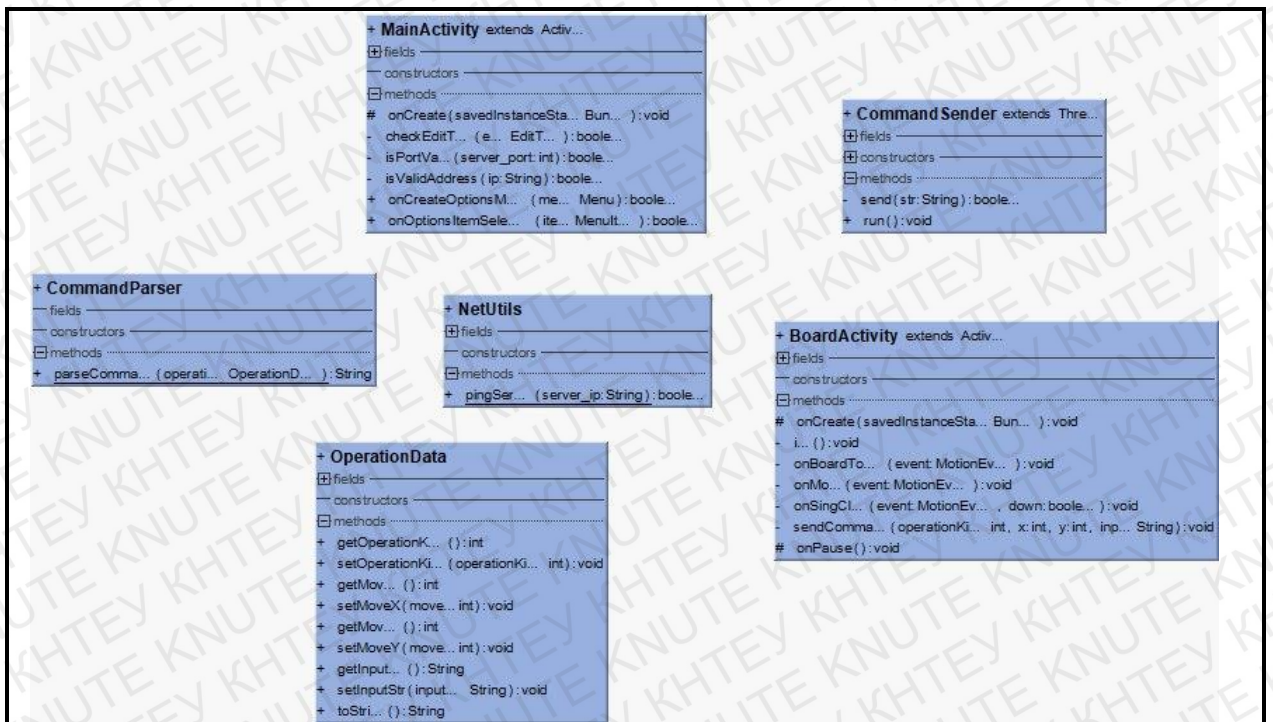


Рис. 3.4. Модулі клієнта. Частина 1

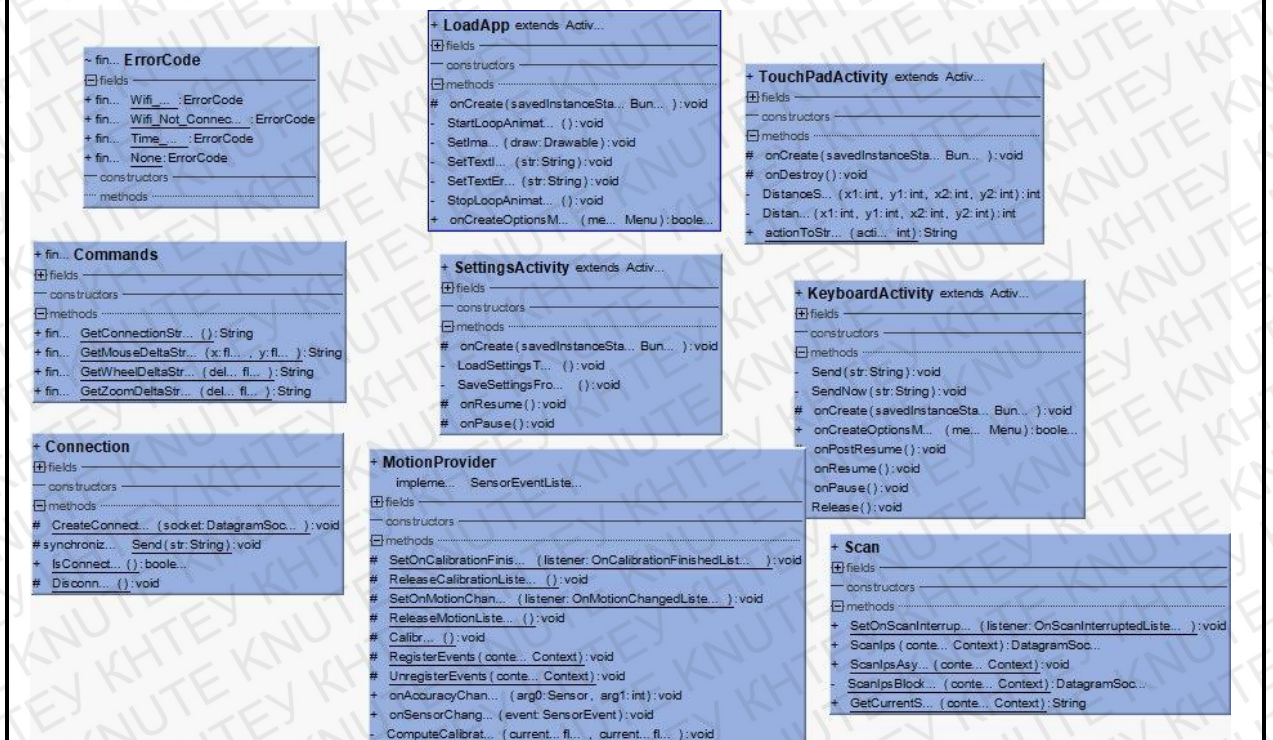


Рис. 3.5. Модулі клієнта. Частина 2

KeyboardActivity – модуль роботи з полем вводу для імітації вводу з клавіатури (Додаток Д).

Connection – процедура встановлення зв'язку із сокетом ПК для подальшої передачі команд (Додаток Е).

NetUtils – обробник параметрів підключення до ПК (IP-адреса, порт) (Додаток Ж).

Scan – «прослуховувач» команд, що передає користувач шляхом взаємодії з робочими зонами імітованих пристроїв вводу ПК (Додаток И).

MotionProvider – обробник взаємодії користувача з екранним сенсором (Додаток К).

OperationData - обробник вхідної команди від користувача та перетворення її у числове значення типу команди для подальшої її інтерпретації сервером (Додаток Л).

CommandParser – упорядкування та перетворення детальних параметрів команди у єдиний масив даних (Додаток М).

CommandSender – передача масиву параметрів на сервер (Додаток Н).

ErrorCode – обробник можливих помилок, що виникають у ході роботи програми, що дозволяє програмі продовжувати безперебійну роботу мобільного додатка (Додаток П).

Перелік набору модулів серверної сторони програмного забезпечення, що виконується на ПК наведено на рис. 3.6.

Опис та призначення кожного з наведених модулів розкрито нижче.

MyServer – модуль запуску програми на ПК, що «вмикає» готовність приймати команди від клієнта (Додаток Р).

Command Receiver – «прослуховувач» сокета на предмет вхідних команд від клієнта (Додаток С).

OperationData – обробник вхідних команд від клієнта, що розпізнає тип команди для подальшої її інтерпретації (Додаток Т).

						Аркуш
						38
Зм.	Аркуш	№ докум	Підпис	Дата	КНТЕУ 121 06з-19.МР	

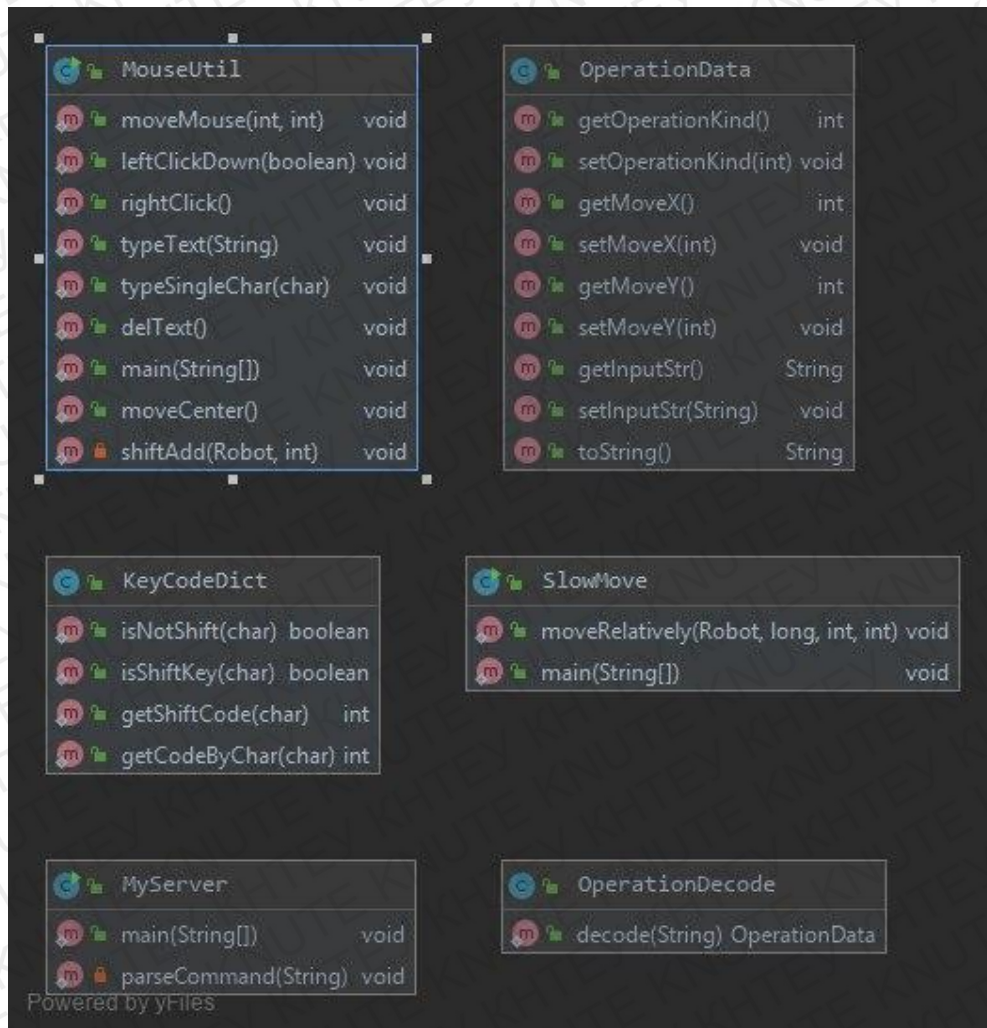


Рис. 3.6. Модулі сервера

OperationDecode – обробляє вхідну команду як масив даних та зчитує властивості команди (її детальні характеристики) (Додаток У).

MouseUtil – перетворює координати руху на імітованому тачпаді у координати руху курсора мишки на екрані комп’ютера (Додаток Ф).

SlowMove – спеціальна процедура навмисного масштабування довжини руху курсора миші на мобільному тачпаді, зумовлена маленькою робочою площею на екрані смартфона у порівнянні з цільовою робочою площею екрана ПК (Додаток Х).

KeyCodeDict – обробляє символи, введені у зону імітованої клавіатури для передачі в активне поле вводу (за наявності) на ПК (Додаток Ц).

### 3.2. Моделювання руху мишки

Рух курсора мишки на екрані комп'ютера формується від впливом багатьох факторів, до яких належать: швидкість, прискорення та прикладена сила, із якою користувач рухає мишку або палець на сенсорній поверхні (тачпаді). Сукупність факторів визначає траєкторію руху, що формалізовано описується рівнянням.

За правилами кінематики

$$\vec{r}(t) \Rightarrow \Delta \vec{r} \Rightarrow \frac{\Delta \vec{r}}{\Delta t} = \vec{v}_c \Rightarrow \lim_{\Delta t \rightarrow 0} \vec{v}_c = \vec{v}(t),$$

$$\vec{r}(t) = \{x(t), y(t), z(t)\}; \vec{v}(t) = \{\dot{x}(t), \dot{y}(t), \dot{z}(t)\}, \quad (3.1)$$

де  $\vec{r}(t)$  – радіус-вектор рухомої точки М (мишки), початок якого співпадає з початком координат, а кінець – з точкою М;

$\Delta \vec{r}$  – зміщення точки за час від довільного моменту  $t$  до моменту  $t + \Delta t$ ;

$\vec{v}_c$  – середня швидкість точки М в інтервалі  $(t, t + \Delta t)$ ;

$\vec{v}(t)$  – швидкість точки М у даний момент  $t$ .

Введена величина – швидкість  $\vec{v}(t)$  – визначається через граничний перехід.

У відповідності до визначення, швидкість  $\vec{v} = d\vec{r}/dt$  – це кінематична міра руху мишки, що дорівнює похідній за часом від радіус-вектору руху в системі відліку. Швидкість характеризує темп зміни радіус-вектору  $\vec{r}$ . Модуль і напрям вектору  $\vec{v}$  співпадають з модулем і напрямом зміщення мишки за одиницю часу в рівномірному русі дотичною до траєкторії за припущення, що швидкість мишки, починаючи з даного моменту, незмінна.

						КНТЕУ 121 06з-19.МР	Аркуш
							40
Зм.	Аркуш	№ докум	Підпис	Дата			

За аналогічною схемою вводиться поняття прискорення  $\vec{w}(t)$ :

$$\vec{v}(t) \Rightarrow \Delta \vec{v} \Rightarrow \frac{\Delta \vec{v}}{\Delta t} = \vec{w}_c \Rightarrow \lim_{\Delta t \rightarrow 0} \vec{w}_c = \vec{w}(t), \quad (3.2)$$

Вектор  $\Delta \vec{v}$  тут можна роз'яснити як приріст швидкості за час  $\Delta t$  в уявному рівноприскореному русі з постійним за величиною та напрямом прискоренням  $\vec{w}_c$ . Прискорення  $\vec{w}$  – міра зміни швидкості мишки, рівна похідній за часом від швидкості мишки у системі відліку. Модуль і напрям вектору  $\vec{w}$  співпадає з вектором швидкості мишки.

Відповідно до рівняння динаміки точки (формула 3.3) розглядають дві задачі:

- Перша задача: за заданим рівнянням руху визначають діючу на точку силу (зворотня задача).
- Друга задача: за заданою силою знаходять рівняння руху (пряма задача).

$$m d^2 \vec{r} / dt^2 = \vec{F} \quad (3.3)$$

Розв'язання оберненої задачі потребує виконання диференціювання, у той час як розв'язання прямої задачі (у нашому випадку з рухом мишки) – інтегрування.

У зв'язку з математичними складнощами інтегрування рівнянь руху особливого значення набуває знаходження перших інтегралів вищезгаданих рівнянь. Перші інтеграли містять ту чи іншу інформацію про рух.

Першим інтегралом диференціальних рівнянь руху називають рівність

$$f(x, y, z, \dot{x}, \dot{y}, \dot{z}, t) = c, \quad (3.4)$$

де  $c$  – довільна стала, що функціонально пов'язує координати рухомої мишки, їх довільні за часом (проекції швидкості) та, можливо, час.

Шість незалежних перших інтегралів руху надають повну інформацію про рух точки у просторі. Дійсно, у результаті розв'язання шести рівнянь

$$f(x, y, z, \dot{x}, \dot{y}, \dot{z}, t) = c_i, \quad i = 1, 2, 3, 4, 5, 6 \quad (3.5)$$

						КНТЕУ 121 06з-19.МР	Аркуш
							41
Зм.	Аркуш	№ докум	Підпис	Дата			



отримаємо шість функцій  $x, y, z, \dot{x}, \dot{y}, \dot{z}, t$ , залежних від часу і шість сталих інтегрування  $c_1, c_2, c_3, c_4, c_5, c_6$ . Тим самим повністю визначаються кінематичні рівняння руху мишки в проекції її швидкості. Сталі знаходяться згідно з заданою початковою умовою руху.

Другим інтегралом називають рівність

$$\varphi(x, y, z, c_1, c_2, c_3, t) = c. \quad (3.6)$$

Достатньо знати три других інтеграла, щоб мати повну інформацію про рух мишки. Її координати  $x, y, z$  визначаються як функції часу і шести сталих інтегрування, що можуть бути знайдені, як зазвичай, із початкової умови руху.

Для того, щоб перейти від векторної форми рівнянь до координатної, необхідно спроектувати рівняння на координатні осі.

Якщо спроектувати рівняння руху (3.3) на декартові осі, отримаємо

$$\frac{md^2x}{dt^2} = F_x, \frac{md^2y}{dt^2} = F_y, \frac{md^2z}{dt^2} = F_z \quad (3.7)$$

або

$$m \frac{dv_x}{dt} = F_x, m \frac{dv_y}{dt} = F_y, m \frac{dv_z}{dt} = F_z. \quad (3.8)$$

У випадку руху в межах однієї площини (мишка рухається у площині  $Oxy$ ), застосовують лише перші два рівняння.

Виходячи з вищевказаного, реалізація руху курсора мишки на екрані комп'ютера з використанням програмного комплексу MobileMouse зображена у вигляді блок-схеми в Додатку III.

Змінні, використані у блок-схемі:

duration – тривалість руху курсора на імітованому тачпаді;

relativeX – довжина переміщення по осі X на імітованому тачпаді;

relativeY – довжина переміщення по осі Y на імітованому тачпаді;

startTime – час у момент початку руху курсора;

endTime – час у момент закінчення руху курсора;

cur\_x – початкові координати курсора на екрані комп'ютера по осі X;

						Аркуш
						42
Зм.	Аркуш	№ докум	Підпис	Дата	КНТЕУ 121 063-19.МР	

cur\_y – початкові координати курсора на екрані комп'ютера по осі Y;  
ratio – співвідношення часу, необхідного для такого відносного переміщення курсора на екрані комп'ютера і часу, затраченого на переміщення на екрані смартфона;

move\_x – довжина переміщення по осі X на екрані комп'ютера;

move\_y – довжина переміщення по осі Y на екрані комп'ютера;

SCREEN\_W – ширина екрану комп'ютера;

SCREEN\_H – висота екрану комп'ютера.

### 3.3. Інтерфейс та функціонал додатку

Клієнтська частина програмного забезпечення, реалізована у вигляді мобільного додатка MobileMouse, має інтерфейс, заснований на двох макетах.

Головний екран (рис. 3.7) містить два поля вводу та одну кнопку. Характеристики макету містяться у файлі activity\_main.xml.

Поле «Server IP» призначене для введення IP-адреси локального ПК, управління курсором миші якого прагне здійснювати користувач.

Поле «Port» призначене для введення номера порту, через який здійснюватиметься підключення та передавання відповідних команд комп'ютеру.

Натискання кнопки «Done» застосовує введені у поля параметри та викликає процедуру підключення до ПК.

Після натискання кнопки «Done» додаток відкриває екран імітації пристроїв вводу ПК (рис. 3.8). Характеристики макету містяться у файлі activity\_board.xml.

Даний екран має наступні складові елементи:

- імітований тачпад (зелений елемент);

						КНТЕУ 121 06з-19.МР	Аркуш
							43
Зм.	Аркуш	№ докум	Підпис	Дата			

- імітовані ліва та права кнопки комп'ютерної мишки (червоні елементи);
- поле введення для імітації вводу з клавіатури.

Відповідно, пересування пальця у зеленій зоні призводитиме до руху курсора мишки на екрані ПК, натискання червоних елементів – до виконання комп'ютером дій при натисканні лівої та правої кнопок мишки, а введений у поле текст автоматично буде введений у активне доступне для редагування поле (за наявності).



Рис. 3.7. Головний екран додатка MobileMouse

						Аркуш
						44
Зм.	Аркуш	№ докум	Підпис	Дата	КНТЕУ 121 06з-19.МР	

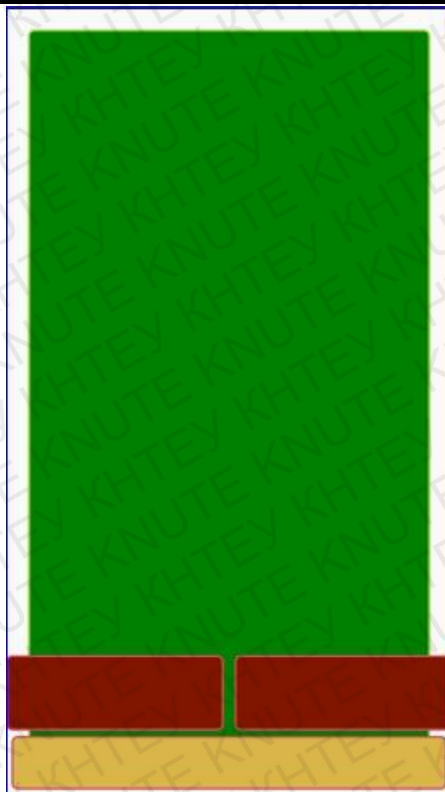


Рис. 3.8. Екран імітації пристроїв вводу ПК

Детальний опис усіх можливих маніпуляцій наведено у п.3.4 «Інструкція користувача».

Серверна частина програмного забезпечення, що виконується на ПК, передбачає лише логінг у консолі виконуваних команд, а отже, не має інтерфейсу як такого.

### 3.4. Висновок до розділу 3

Отже, систематизоване поєднання наведених модулів у клієнтську та серверну частини, і налагодження обміну даними між ними, дозволили створити єдиний програмний комплекс для управління курсором мишки та вводом тексту на ПК зі смартфона, що входять до однієї локальної мережі.

					КНТЕУ 121 06з-19.МР	Аркуш
						45
Зм.	Аркуш	№ докум	Підпис	Дата		

## ВИСНОВКИ ТА ПРОПОЗИЦІЇ

З проведених досліджень можна зробити наступні висновки:

1. Проведено аналіз слабких місць комп'ютерних маніпуляторів та вибрано клієнт-серверну архітектуру застосунку, яка є оптимальною для реалізації поставлених завдань. Серверна частина, на відміну від альтернативних рішень, встановлюється локально і не вимагає доступу до Інтернет.
2. В якості операційних систем обрано ОС Android (мобільна система) та Windows (персональний комп'ютер) як найзручніші та найрозповсюдженіші ОС на сучасному ринку. Поліпшена модель розрахунку швидкості руху курсора мишки, а також використання Windows 10 як базової комп'ютерної ОС у процесі розробки з її вдосконаленою системою захисту складає новизну роботи, оскільки аналогів, які би надійно працювали у такому середовищі не знайдено.
3. Проведений аналіз засобів розробки показав, що найзручнішим для реалізації поставлених завдань є Android Studio та IntelliJIDEA.
4. В процесі виконання магістерської роботи розроблено систему дистанційного керування комп'ютером через wi-fi - з'єднання з імітацією роботи тачпаду за допомогою смартфона.
5. Розроблена система повністю задовольняє поставлені функціональні вимоги. Тестування продемонструвало повну відповідність функціональним вимогам та надійну роботу застосунку.

<i>КНТЕУ 121 06з-07.МР</i>					
Зм.	Аркуш	№ докум.	Підпис	Дата	
Зав. каф.		Криворучко О.В.		30.10.20	
Керівник		Криворучко О.В.		30.10.20	
Гарант		Криворучко О.В.		30.10.20	
Розробив		Шевченко Ю.О.		30.10.20	
<i>Висновки та пропозиції</i>					
Розробка Android-додатку для імітації тачпаду ПК			<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
			<i>ВП</i>	46	50
			Факультет інформаційних технологій 2м курс, 6 група		

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Пристрої введення та виведення даних [Електронний ресурс]. - Режим доступу: <https://narodna-osvita.com.ua/5353-pristroyi-vvedennya-ta-vivedennya-danih.html>
2. Апаратне забезпечення персонального комп'ютера [Електронний ресурс]. - Режим доступу: [https://kafinfo.org.ua/files/Informatyka\\_10\\_11/Glava\\_2\\_5.pdf](https://kafinfo.org.ua/files/Informatyka_10_11/Glava_2_5.pdf)
3. Трекбол. Історія виникнення і особливості використання [Електронний ресурс]. - Режим доступу: <http://ecolora.ru/index.php/2010-07-09-03-51-16/veb-dizajn-i-programmirovanie/1009-trekbol-istorija-voznikovenija-i-osobennosti-ispolzovanija>
4. Комп'ютерна миша [Електронний ресурс]. - Режим доступу: [https://uk.wikipedia.org/wiki/Комп%27ютерна\\_миша](https://uk.wikipedia.org/wiki/Комп%27ютерна_миша)
5. Мышь [Електронний ресурс]. - Режим доступу: <http://www.tofmal.ru/projects/contest/ber/manipulators.html>
6. Історія розвитку пристроїв введення та виведення [Електронний ресурс]. - Режим доступу: [http://vvod-vivod.blogspot.com/p/blog-page\\_3386.html](http://vvod-vivod.blogspot.com/p/blog-page_3386.html)
7. Історія ігрового джойстика [Електронний ресурс]. - Режим доступу: <http://scsiexplorer.com.ua/index.php/istoria-otkritiy/2238-istorija-igrovogo-dzhojstika.html>
8. Разница между геймпадом и джойстиком [Електронний ресурс]. - Режим доступу: <https://thedifference.ru/chem-otlichaetsya-gejmpad-ot-dzhojstika/>
9. Джойстик [Електронний ресурс]. - Режим доступу: <https://uk.wikipedia.org/wiki/Джойстик>

<i>КНТЕУ 121 06з-19.МР</i>				
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>
Зав. каф.		Криворучко О.В.		24.01.20
Керівник		Криворучко О.В.		24.01.20
Гарант		Криворучко О.В.		24.01.20
Розробив		Шевченко Ю.О.		24.01.20
Розробка Android-додатку для імітації тачпаду ПК				
Список використаних джерел				
		<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
		<i>СВД</i>	47	50
Факультет інформаційних технологій 2м курс, 6 група				

10. Тачпад [Электронний ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/Тачпад>.
11. Как Android стал доминирующей мобильной ОС [Электронний ресурс]. – Режим доступу: <https://pdalife.info/video-pokazalo-kak-android-stal-dominiruyushey-mobilnoy-os-5721p.html>
12. Какая операционная система лучше для смартфона? [Электронний ресурс]. – Режим доступу: <https://setphone.ru/stati/kakaya-operacionnaya-sistema-luchshe-dlya-smartfona/>
13. What is KaiOS? [Электронний ресурс]. – Режим доступу: <https://primea2z.blogspot.com/2017/10/what-is-kaios.html>
14. KaiOS Emerging OS [Электронний ресурс]. – Режим доступу: <https://www.developingtelecoms.com/tech/devices-platforms/7595>
15. KaiOS is on 30 million phones and now has Google apps, Facebook, Twitter [Электронний ресурс]. – Режим доступу: <https://mobilityarena.com/kaios-30-million-phones/>
16. iOS [Электронний ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/IOS>
17. iOS проти Android: 10 переваг операційної системи Apple [Электронний ресурс]. – Режим доступу: <https://ishop.if.ua/novyny/ios-proty-android-10-perevag-operaciynoyi-systemy-apple>
18. Android [Электронний ресурс]. – Режим доступу: <https://ru.wikipedia.org/wiki/Android>
19. 6 серйозних недоліків Android, які точно потрібно виправити [Электронний ресурс]. – Режим доступу: <http://dialogs.org.ua/tech/15758/>
20. An Overview of the Android Architecture [Электронний ресурс]. – Режим доступу: [https://www.techotopia.com/index.php/An\\_Overview\\_of\\_the\\_Android\\_Architecture](https://www.techotopia.com/index.php/An_Overview_of_the_Android_Architecture)

					<i>КНТЕУ 121 063-19.МР</i>	Аркуш
						48
Зм.	Аркуш	№ докум	Підпис	Дата		

21. Чорна, А. В. and Деменок, К. С. (2018) Порівняльний аналіз версій операційної системи Android. In?ynieria i technologia. Teoretyczne i praktyczne aspekty rozwoju współczesnej nauki: zbiór artykułów naukowych. Konferencji Międzynarodowej NaukowoPraktycznej (29.12.2018-30.12.2018). pp. 37-41. ISSN 978-83-66030-72-5
22. История версий Android [Електронний ресурс]. – Режим доступу: [https://ru.wikipedia.org/wiki/История\\_версий\\_Android](https://ru.wikipedia.org/wiki/История_версий_Android)
23. Яка версія Android швидше працює: порівняли Nougat, Oreo, Pie та Android Q [Електронний ресурс]. – Режим доступу: <https://techtoday.in.ua/news/yaka-versiya-android-shvydshe-pratsyuye-porivnyaly-nougat-oreo-pie-ta-android-q-112207.html>
24. Android Studio 1.0 [Електронний ресурс]. – Режим доступу: <https://android-developers.googleblog.com/2014/12/android-studio-10.html>
25. Android Studio: An IDE built for Android [Електронний ресурс]. – Режим доступу: <https://android-developers.googleblog.com/2013/05/android-studio-ide-built-for-android.html>
26. Google intros Android Studio, an IDE for building apps [Електронний ресурс]. – Режим доступу: <http://www.engadget.com/2013/05/15/google-android-studio/>
27. Android Studio unveiled at Google I/O keynote [Електронний ресурс]. – Режим доступу: <http://www.androidcentral.com/android-studio-unveiled-google-io-keynote>
28. Google Launches Android Studio And New Features For Developer Console, Including Beta Releases And Staged Rollout [Електронний ресурс]. – Режим доступу: <http://techcrunch.com/2013/05/15/google-launches-android-studio-a-development-tool-for-apps/>
29. Android Studio [Електронний ресурс]. – Режим доступу: <https://developer.android.com/studio>

					<i>КНТЕУ 121 063-19.МР</i>	Аркуш
						49
Зм.	Аркуш	№ докум	Підпис	Дата		



30. The Java History Timeline [Електронний ресурс]. – Режим доступу: <http://www.java.com/en/javahistory/timeline.jsp>
31. 1.2 Design Goals of the Java™ Programming Language [Електронний ресурс]. – Режим доступу: <http://www.oracle.com/technetwork/java/intro-141325.html>
32. Текст ліцензії Microsoft [Електронний ресурс]. – Режим доступу: <http://msdnaa.oit.umass.edu/Neula.asp>
33. XML [Електронний ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/XML>
34. Layouts [Електронний ресурс]. – Режим доступу: <https://developer.android.com/guide/topics/ui/declaring-layout>
35. Автоматизація складання [Електронний ресурс]. – Режим доступу: [https://uk.wikipedia.org/wiki/Автоматизація\\_складання](https://uk.wikipedia.org/wiki/Автоматизація_складання)
36. Build and Release Management [Електронний ресурс]. – Режим доступу: <https://web.archive.org/web/20070930222406/http://freshmeat.net/articles/vi>  
[ew/392/](https://web.archive.org/web/20070930222406/http://freshmeat.net/articles/vi)
37. Gradle [Електронний ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/Gradle>
38. Configure your build [Електронний ресурс]. – Режим доступу: <https://developer.android.com/studio/build>
39. App components [Електронний ресурс]. – Режим доступу: <https://developer.android.com/guide/components/fundamentals.html#Components>
40. The manifest file [Електронний ресурс]. – Режим доступу: <https://developer.android.com/guide/components/fundamentals.html#Manifest>  
[t](https://developer.android.com/guide/components/fundamentals.html#Manifest)

					<i>КНТЕУ 121 063-19.МР</i>	Аркуш
						50
Зм.	Аркуш	№ докум	Підпис	Дата		

## ТЕХНІЧНЕ ЗАВДАННЯ

Однією з найголовніших заporук відповідності вимогам та високої якості розробленого програмного рішення є грамотно спроектоване до початку розробки технічне завдання. Отже, далі розкриємо невід'ємні положення технічного завдання до розробки Android-додатку для імітації тачпаду ПК.

### 1. Загальні відомості

*Назва проекту:* MobileMouse.

*Планові строки розробки:* серпень 2019 – вересень 2019.

*Потенційні користувачі:* користувачі стаціонарних ПК з встановленою ОС Windows.

*Призначення програмного рішення:* надати можливість користувачу ПК за відсутності мишки управляти її курсором на екрані за допомогою свого смартфона.

*Мета створення програмного рішення:* опанування сучасних технологій розробки Android-додатків та їх закріплення на практиці.

### 2. Структура проекту

Оскільки процес управління передбачає наявність суб'єкта управління (смартфон) і об'єкта управління (курсор миші на екрані ПК), програмне рішення складатиметься з двох частин: серверної і клієнтської.

Серверна частина виконуватиметься на ПК, оскільки він надає доступ до своєї системи.

Клієнтська частина буде реалізована у вигляді Android-додатку, що виконуватиметься на смартфоні.

Зм.	Аркуш	№ докум.	Підпис	Дата				
					<i>КНТЕУ 121 063-19.МР</i>			
Зав. каф.		Криворучко О.В.		28.02.20	Розробка Android-додатку для імітації тачпаду ПК	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
Керівник		Криворучко О.В.		28.02.20		<i>ТЗ</i>	<i>51</i>	<i>60</i>
Гарант		Криворучко О.В.		28.02.20		Факультет інформаційних технологій 2м курс, 6 група		
Розробив		Шевченко Ю.О.		28.02.20				
					<i>Технічне завдання</i>			

### **3. Функціональні вимоги до додатку**

Перелік функціональних можливостей визначений базовими діями, що виконує комп'ютерна мишка:

- натискання лівої кнопки;
- подвійне натискання лівої кнопки;
- натискання правої кнопки;
- переміщення курсора миші на екрані ПК.

Крім цього, додаток повинен містити екран вибору конкретного ПК, до якого під'єднуватиметься смартфон для виконання ролі тачпада.

### **4. Вимоги до програмного забезпечення**

Розробка серверної частини програмного рішення здійснюватиметься із використанням мови програмування Java, у середовищі розробки IntelliJ IDEA.

Розробка мобільного додатку (клієнтської частини програмного рішення) забезпечуватиметься наступним:

- мова програмування Java;
- мова розмітки XML;
- інструментарій автоматизації збірки Gradle;
- середовище розробки Android Studio.

### **5. Вимоги до технічного забезпечення**

Найважливішою вимогою до задіяних пристроїв є належність до однієї локальної мережі.

Серверна частина для ПК має виконуватися за наступних мінімальних конфігурацій:

- Microsoft Windows 10/8/7.
- RAM: 1 Гб і більше.
- JDK 1.6 і вище.

						КНТЕУ 121 063-19.МР	Аркуш
							52
Зм.	Аркуш	№ докум	Підпис	Дата			

• Передвстановлене середовище виконання Java-сценаріїв із можливістю одночасного перегляду консолі під час виконання.

Клієнтська частина (мобільний додаток) має виконуватися на смартфоні із наступними конфігураціями:

- Android 10/9/8/7/6/5;
- RAM: 2 Гб і більше.

						Аркуш
						53
Зм.	Аркуш	№ докум	Підпис	Дата	КНТЕУ 121 063-19.МР	

## ТЕСТУВАННЯ ДОДАТКА MOBILEMOUSE

З метою забезпечення якості розробленого програмного забезпечення нами було проведено тестування мобільного додатка для імітації тачпада ПК.

Оскільки об'єкт дослідження передбачає взаємодію двох різнотипних пристроїв (смартфон та комп'ютер) тестування в Android-емуляторі є недоцільним. Так, для тестування розробленої програми було використано реальні пристрої з наступними конфігураціями (табл. 3.1).

Таблиця Т.1

Конфігурація пристроїв, використаних під час тестування

Характеристика	Комп'ютер	Смартфон
Операційна система	Windows 10 Pro (x64)	Android 8.0
Об'єм ОЗУ	8 Гб	4 Гб
Спосіб підключення до локальної мережі	Wi-Fi	Wi-Fi

На основі вимог до програмного забезпечення (технічне завдання) було розроблено перелік обов'язкових позитивних тест-кейсів (тестових випадків), виконання яких ми перевірили. Зміст тест-кейсів зазначено у табл. 3.2.

Передумова: виконано запуск обох частин програмного забезпечення (клієнтської та серверної).

					<i>КНТЕУ 121 063-19.МР</i>			
Зм.	Аркуш	№ докум.	Підпис	Дата				
Зав. каф.	Криворучко О.В.			24.01.20	Розробка Android-додатку для імітації тачпада ПК	Стадія	Аркуш	Аркушів
Керівник	Криворучко О.В.			24.01.20		ПМТ	54	60
Гарант	Криворучко О.В.			24.01.20		Факультет інформаційних технологій 2м курс, 6 група		
Розробив	Шевченко Ю.О.			24.01.20				
					<i>Програма та методика тестування</i>			

## Позитивні тест-кейси на основі технічного завдання

п.п.	Виконувана дія	Очікуваний результат	Результат тесту
1	Заповнення полів головного екрану коректними IP-адресою комп'ютера у мережі та номером вільного порту. Натискання кнопки «Done».	Виконано з'єднання з комп'ютером. У мобільному додатку відкривається екран імітації пристроїв вводу ПК.	
2	Проведення пальцем у зоні імітації тачпада вгору.	Курсор миші на екрані ПК рухається вгору.	
3	Проведення пальцем у зоні імітації тачпада вниз.	Курсор миші на екрані ПК рухається вниз.	
4	Проведення пальцем у зоні імітації тачпада вліво.	Курсор миші на екрані ПК рухається вліво.	
5	Проведення пальцем у зоні імітації тачпада вправо.	Курсор миші на екрані ПК рухається вправо.	
6	Натискання лівої червоної кнопки.	Виконання комп'ютером дії натискання лівої кнопки миші.	
7	Натискання правої червоної кнопки.	Виконання комп'ютером дії натискання правої кнопки миші.	
8	Подвійне натискання лівої червоної кнопки.	Виконання комп'ютером дії подвійного натискання лівої кнопки миші.	

Усі вищенаведені тест-кейси були виконані успішно.

Виконання **тест-кейсу 1** зображено на рис. 3.9.

Тестування здійснювалося з паралельним моніторингом логів виконуваних сервером дій у консолі IDE IntelliJ IDEA.

Відтак, тест-кейси 2-8 супроводжувалися наступним логінгом.

**Тест-кейс 2. Проведення пальцем у зоні імітації тачпада вгору:**

```
got a socket... socket:Socket[addr=/192.168.0.101,port=38140,localport=6221]
{"moveX":5,"moveY":-121,"operationKind":0}
Start Server
close
```

						Аркуш
						55
Зм.	Аркуш	№ докум	Підпис	Дата	КНТЕУ 121 063-19.МР	

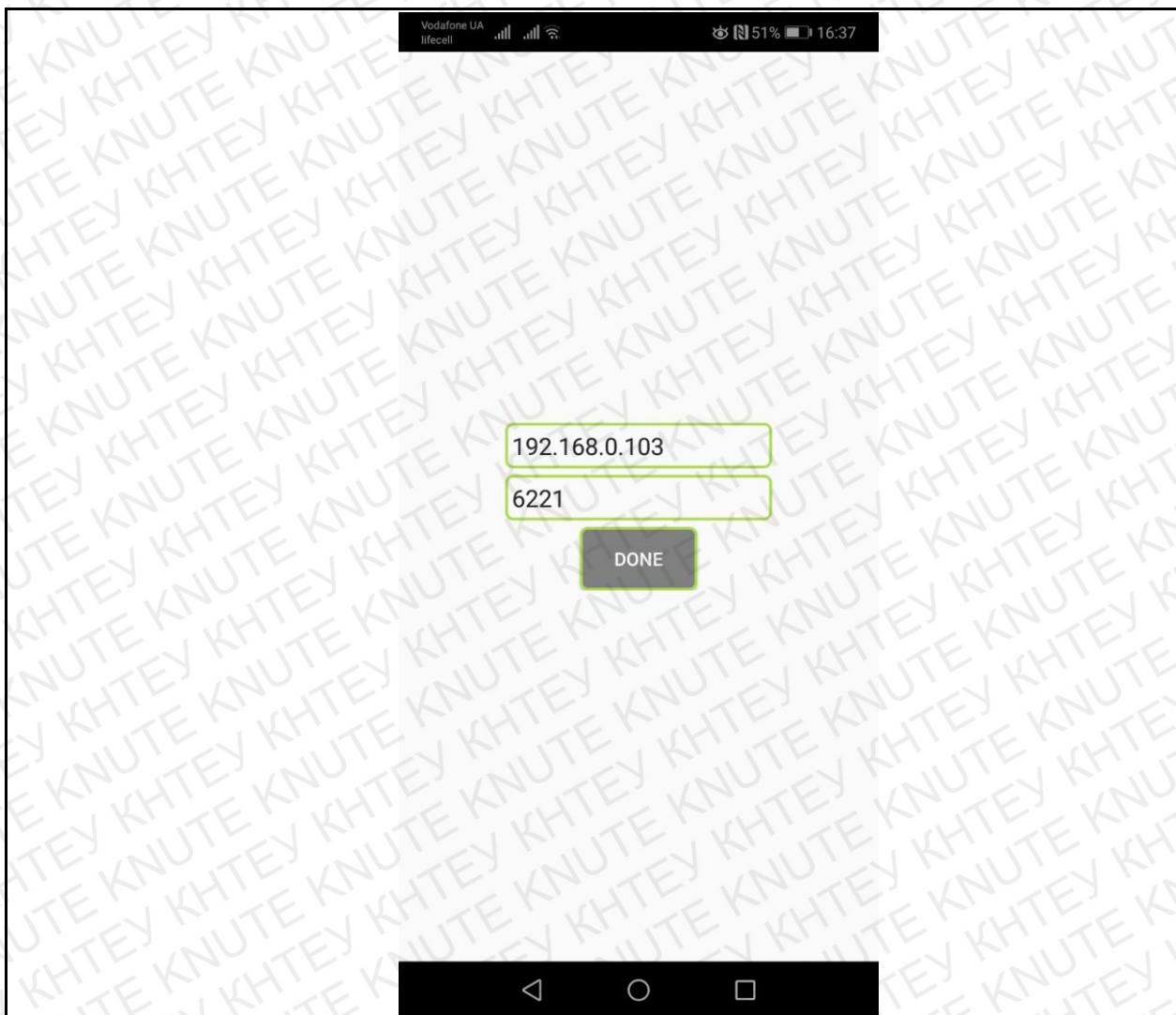


Рис. Т.9. Заповнення полів у тест-кейсі 1

**Тест-кейс 3. Проведення пальцем у зоні імітації тачпада вниз:**

```
got a socket... socket:Socket[addr=/192.168.0.101,port=38142,localport=6221]
{"moveX":3,"moveY":78,"operationKind":0}
Start Server
close
```

**Тест-кейс 4. Проведення пальцем у зоні імітації тачпада вліво:**

```
got a socket... socket:Socket[addr=/192.168.0.101,port=38144,localport=6221]
{"moveX":-78,"moveY":4,"operationKind":0}
Start Server
close
```

**Тест-кейс 5. Проведення пальцем у зоні імітації тачпада вправо:**

```
got a socket... socket:Socket[addr=/192.168.0.101,port=38146,localport=6221]
{"moveX":185,"moveY":-1,"operationKind":0}
Start Server
```

					<i>КНТЕУ 121 063-19.МР</i>	Аркуш
						56
Зм.	Аркуш	№ докум	Підпис	Дата		

close

### Тест-кейс 6. Натискання лівої червоної кнопки:

```
got a socket... socket:Socket[addr=/192.168.0.101,port=38280,localport=6221]
{"moveX":-1,"moveY":-1,"operationKind":1}
```

Start Server

close

```
got a socket... socket:Socket[addr=/192.168.0.101,port=38282,localport=6221]
{"moveX":-1,"moveY":-1,"operationKind":2}
```

Start Server

close

### Тест-кейс 7. Натискання правої червоної кнопки:

```
got a socket... socket:Socket[addr=/192.168.0.101,port=38284,localport=6221]
{"moveX":-1,"moveY":-1,"operationKind":3}
```

Start Server

close

### Тест-кейс 8. Подвійне натискання лівої червоної кнопки:

```
got a socket... socket:Socket[addr=/192.168.0.101,port=38288,localport=6221]
{"moveX":-1,"moveY":-1,"operationKind":1}
```

Start Server

close

```
got a socket... socket:Socket[addr=/192.168.0.101,port=38290,localport=6221]
{"moveX":-1,"moveY":-1,"operationKind":2}
```

Start Server

close

```
got a socket... socket:Socket[addr=/192.168.0.101,port=38292,localport=6221]
{"moveX":-1,"moveY":-1,"operationKind":1}
```

Start Server

close

```
got a socket... socket:Socket[addr=/192.168.0.101,port=38294,localport=6221]
{"moveX":-1,"moveY":-1,"operationKind":2}
```

Start Server

close

В процесі розробки мобільного додатка ми також реалізували й інші додаткові можливості, для перевірки коректності роботи яких доповнено тест-кейси переліком у табл. Т.3.

Таблиця Т.3

Тест-кейси додаткових дій у додатку MobileMouse

№ п.п.	Виконувана дія	Очікуваний результат	Результат тесту
9	<u>Передумова:</u> на екрані ПК активовано текстове поле для редагування. Введення у поле на екрані мобільного додатка символу латиниці.	Введений з мобільної клавіатури символ з'явився у місці редагування текстового поля ПК.	

						КНТЕУ 121 063-19.МР	Аркуш
							57
Зм.	Аркуш	№ докум	Підпис	Дата			



№ п.п.	Виконувана дія	Очікуваний результат	Результат тесту
10	<u>Передумова:</u> на екрані ПК активовано текстове поле для редагування. Введення у поле на екрані мобільного додатка символу кирилиці.	Введений з мобільної клавіатури символ з'явився у місці редагування текстового поля ПК.	
11	<u>Передумова:</u> на екрані ПК активовано непорожнє текстове поле для редагування. Натискання кнопки «Backspace» у полі мобільного додатка.	Видалення символу у активованому полі на екрані ПК зліва від встановленого курсору.	
12	<u>Передумова:</u> на екрані ПК активовано текстове поле для редагування. Натискання кнопки «Enter» у полі мобільного додатка.	Виконання комп'ютером дії натискання кнопки «Enter».	
13	Обрати на робочому столі ПК або у файл-менеджері файл/теку одиночним натисканням лівої червоної кнопки у додатку. Не відпускаючи ліву червону кнопку за допомогою імітованого тачпада перетягнути файл/теку у довільне місце призначення іншим пальцем. Відпустити ліву червону кнопку і тачпад.	Обраний файл/тека перемістився у вказане місце.	

Усі вищенаведені тест-кейси були виконані успішно та супроводжувалися наступним логінгом.

**Тест-кейс 9. Введення у поле на екрані мобільного додатка символу латиниці:**

```
got a socket... socket:Socket[addr=/192.168.0.101,port=38298,localport=6221]
{"inputStr":"w","moveX":-1,"moveY":-1,"operationKind":5}
```

					Аркуш
					58
Зм.	Аркуш	№ докум	Підпис	Дата	

*КНТЕУ 121 063-19.МР*

Start Server  
close

### Тест-кейс 10. Введення у поле на екрані мобільного додатка символу

#### кирилиці:

```
got a socket... socket:Socket[addr=/192.168.0.101,port=38306,localport=6221]
{"inputStr":"т","moveX":-1,"moveY":-1,"operationKind":5}
Start Server
close
```

### Тест-кейс 11. Натискання кнопки «Backspace» у полі мобільного

#### додатка:

```
got a socket... socket:Socket[addr=/192.168.0.101,port=38314,localport=6221]
{"moveX":-1,"moveY":-1,"operationKind":4}
Start Server
close
```

### Тест-кейс 12. Натискання кнопки «Enter» у полі мобільного

#### додатка:

```
got a socket... socket:Socket[addr=/192.168.0.101,port=39504,localport=6221]
{"inputStr":"\n","moveX":-1,"moveY":-1,"operationKind":5}
Start Server
close
```

### Тест-кейс 13. Переміщення файлу/теки ПК за допомогою

#### імітованого тачпада та лівої кнопки мишки:

```
got a socket... socket:Socket[addr=/192.168.0.101,port=39514,localport=6221]
{"moveX":-1,"moveY":-1,"operationKind":1}
Start Server
close
got a socket... socket:Socket[addr=/192.168.0.101,port=39518,localport=6221]
{"moveX":-1,"moveY":-1,"operationKind":0}
Start Server
close
got a socket... socket:Socket[addr=/192.168.0.101,port=39520,localport=6221]
{"moveX":4,"moveY":25,"operationKind":0}
Start Server
close
got a socket... socket:Socket[addr=/192.168.0.101,port=39578,localport=6221]
{"moveX":-1,"moveY":-1,"operationKind":2}
Start Server
close
```

Таким чином, ми провели функціональне тестування програмного забезпечення MobileMouse та довели справне виконання його призначення на прикладі найбільш часто виконуваних дій користувачів із комп'ютерним маніпулятором. Також, продемонстрували та протестували додаткові

					КНТЕУ 121 063-19.МР	Аркуш
						59
Зм.	Аркуш	№ докум	Підпис	Дата		

елементарні дії, що зможе виконувати користувач із задіянням мобільної клавіатури (набір тексту та його видалення).

						Аркуш
						60
Зм.	Аркуш	№ докум	Підпис	Дата	<i>КНТЕУ 121 063-19.МР</i>	

# ДОДАТКИ

## ДОДАТОК А

### Код класу LoadApp

```
public class LoadApp extends Activity {
    byte _wifiStep = 0;
    ImageView _imageView;
    TextView _textViewInfo;
    TextView _textViewError;
    Timer _timer = null;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_board);
        getWindow().addFlags(WindowManager.LayoutParams.FLAG_KEEP_SCREEN_ON);
        Settings.LoadSettings(getApplicationContext());
        PackageManager packageManager = getPackageManager();
        boolean gyroExists = packageManager.hasSystemFeature(PackageManager.FEATURE_SENSOR_GYROSCOPE);
        if (!gyroExists) {
            SetTextError("This phone is not equipped with gyroscope, air mouse function won't be available");
            Settings.setMODE(Settings.Mode.Touchpad);
        }
        StartLoopAnimation();
        Scan.SetOnScanInterrupted(OnScanInterrupted);
        Scan.ScanIpsAsync(this.getContext());
    }
    private void StartLoopAnimation() {
        _timer = new Timer();
        _timer.scheduleAtFixedRate(new TimerTask() {
            @Override
            public void run() {
                Drawable clickedDrawable = null;
                if (_wifiStep == 0)
                    clickedDrawable = getResources().getDrawable(R.drawable.phone_border);
                else if (_wifiStep == 1)
                    clickedDrawable = getResources().getDrawable(R.drawable.phone_border);
                else if (_wifiStep == 2)
                    clickedDrawable = getResources().getDrawable(R.drawable.phone_border);
                else if (_wifiStep == 3)
                    clickedDrawable = getResources().getDrawable(R.drawable.phone_border);
                SetImage(clickedDrawable);
                if (_wifiStep == 3)
                    _wifiStep = 0;
                else
                    _wifiStep++;
            }
        }, 500, 500);
    }
}
```

						Аркуш
						61
Зм.	Аркуш	№ докум	Підпис	Дата	КНТЕУ 121 063-19.МР	

```
private void SetImage(final Drawable draw) {
```

## Продовження дод. А

```
    runOnUiThread(new Runnable() {  
        public void run() {  
            _imageView.setImageDrawable(draw);  
        }  
    });  
}
```

```
private void SetTextInfo(final String str) {  
    runOnUiThread(new Runnable() {  
        public void run() {  
            _textViewInfo.setText(str);  
        }  
    });  
}
```

```
private void SetTextError(final String str) {  
    runOnUiThread(new Runnable() {  
        public void run() {  
            _textViewError.setText(str);  
        }  
    });  
}
```

```
private void StopLoopAnimation() {  
    if (_timer != null) {  
        _timer.cancel();  
        _timer.purge();  
    }  
    SetImage(getResources().getDrawable(R.drawable.phone_border));  
}
```

```
@Override  
public boolean onCreateOptionsMenu(Menu menu) {  
    return true;  
}
```

```
private OnScanInterruptedException OnScanInterrupted = new OnScanInterruptedException() {  
    @Override  
    public void IpFound(DatagramSocket socket) {  
        StopLoopAnimation();  
        SetTextInfo("Connected");  
        Connection.CreateConnection(socket);  
        Intent intent;  
        if (Settings.getMODE() == Settings.Mode.Gyroscope) {  
            intent = new Intent(getApplicationContext(), MainActivity.class);  
        } else  
            intent = new Intent(getApplicationContext(), TouchPadActivity.class);  
        finish();  
        startActivity(intent);  
    }  
}
```

```
@Override  
public void ErrorHappened(ErrorCode error) {
```

					Аркуш
					62
Зм.	Аркуш	№ докум	Підпис	Дата	

*KHTEY 121 063-19.MP*

						Аркуш
					<i>КНТЕУ 121 063-19.МР</i>	63
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>		

**Код класу MainActivity**

```

public class MainActivity extends Activity {
    public static final String SERVER_IP_FLAG = "com.zhanglei.server.ip";
    public static final String SERVER_PORT_FLAG = "com.zhanglei.server.port";
    private static final String PATTERN =
        "^((([01]?\\d\\d?|2[0-4]\\d|25[0-5]))\\.){3}([01]?\\d\\d?|2[0-4]\\d|25[0-5])$";
    private EditText editIp, editPort;
    private Button btnDone;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        editIp = (EditText) findViewById(R.id.edit_ip);
        editPort = (EditText) findViewById(R.id.edit_port);
        btnDone = (Button) findViewById(R.id.ip_port_sure);
        btnDone.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String ipStr = null;
                int port = -1;
                boolean isAddressInvalid = false;
                boolean isPortInvalid = false;
                if (checkEditText(editIp)) {
                    ipStr = editIp.getText().toString();
                    if (!isValidAddress(ipStr)) {
                        ipStr = null;
                        isAddressInvalid = true;
                    }
                }
                if (checkEditText(editPort)) {
                    port = Integer.parseInt(editPort.getText().toString());
                    if (!isPortValid(port)) {
                        port = -1;
                        isPortInvalid = true;
                    }
                }
                String errorStr = "";
                if (isAddressInvalid && isPortInvalid) {
                    errorStr = "Invalid address and port!";
                } else if (isAddressInvalid || isPortInvalid) {
                    errorStr = "Invalid address or port";
                }
                if (!errorStr.equals("")) {
                    Toast.makeText(MainActivity.this, errorStr, Toast.LENGTH_LONG).show();
                }
                Intent intent = new Intent (MainActivity.this, BoardActivity.class);
            }
        });
    }
}

```

						Аркуш
						64
Зм.	Аркуш	№ докум	Підпис	Дата		

KHTEY 121 063-19.MP

```

if (port != -1) {
    intent.putExtra(SERVER_PORT_FLAG, port);
}

```

## Продовження дод. Б

```

if (ipStr!=null) {
    intent.putExtra(SERVER_IP_FLAG, ipStr);
}
startActivity(intent);
}
});
}
private boolean checkEditText(EditText edit) {
    boolean result = true;
    String editStr = edit.getText().toString();
    if (editStr == null || editStr.equals("")) {
        return false;
    }
    return result;
}
private boolean isPortValid(int server_port) {
    boolean result = true;
    if (server_port < 0 || server_port > 65535) {
        return false;
    }
    return result;
}
private boolean isValidAddress(String ip) {
    boolean result = true;
    Pattern pattern = Pattern.compile(PATTERN);
    Matcher matcher = pattern.matcher(ip);
    if (!matcher.matches()) {
        return false;
    }
    return result;
}
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.menu_main, menu);
    return true;
}
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    int id = item.getItemId();
    //noinspection SimplifiableIfStatement
    if (id == R.id.action_settings) {
        return true;
    }
}

```

					<i>КНТЕУ 121 063-19.МР</i>	Аркуш
						65
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>		



return super.onOptionsItemSelected(item);

}

}

						Аркуш
						66
Зм.	Аркуш	№ докум	Підпис	Дата	<i>КНТЕУ 121 063-19.МР</i>	

**Код класу BoardActivity**

```

public class BoardActivity extends Activity {

    private CommandSender mSender;
    private View touchBoard;
    private View mLeftClick;
    private View mRightClick;
    private EditText boardInput;
    private String mLastInput;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        init();
    }
    private void init() {
        String ipStr = getIntent().getStringExtra(MainActivity.SERVER_IP_FLAG);
        int port = getIntent().getIntExtra(MainActivity.SERVER_PORT_FLAG, -1);

        setContentView(R.layout.activity_board);
        touchBoard = (View) findViewById(R.id.touch_board);
        mLeftClick = (View) findViewById(R.id.btn_left_click);
        mRightClick = (View) findViewById(R.id.btn_right_click);
        boardInput = (EditText) findViewById(R.id.edit_board_input);
        mLeftClick.setOnTouchListener(new View.OnTouchListener() {
            @Override
            public boolean onTouch(View v, MotionEvent event) {
                switch (event.getActionMasked()) {
                    case MotionEvent.ACTION_DOWN:
                        sendCommand(OperationData.OPERATION_CLICK_DOWN, 0, 0, null);
                        break;
                    case MotionEvent.ACTION_UP:
                        sendCommand(OperationData.OPERATION_CLICK_UP, 0, 0, null);
                        break;
                }
                return true;
            }
        });

        mRightClick.setOnTouchListener(new View.OnTouchListener() {
            @Override
            public boolean onTouch(View v, MotionEvent event) {
                switch (event.getActionMasked()) {
                    case MotionEvent.ACTION_DOWN:
                        sendCommand(OperationData.OPERATION_RIGHT_CLICK, 0, 0, null);
                        break;
                }
            }
        });
    }
}

```

					<i>КНТЕУ 121 063-19.МР</i>	Аркуш
						67
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>		

```

return true;
}
});

```

## Продовження дод. В

```

boardInput.setOnKeyListener(new View.OnKeyListener() {
    @Override
    public boolean onKey(View v, int keyCode, KeyEvent event) {
        //if (KeyEvent.KEYCODE_BACK == keyCode) {
            //Toast.makeText(BoardActivity.this, "click back" + keyCode, Toast.LENGTH_LONG).show();
        //}
        if (keyCode == KeyEvent.KEYCODE_DEL && event.getAction() == KeyEvent.ACTION_DOWN) {
            sendCommand(OperationData.OPERATION_DEL_TEXT, 0, 0, null);
        }
        return false;
    }
});
boardInput.addTextChangedListener(new TextWatcher() {
    @Override
    public void beforeTextChanged(CharSequence s, int start, int count, int after) {
        Log.i("fuck", "before: " + s);
        mLastInput = s.toString();
    }
    @Override
    public void onTextChanged(CharSequence s, int start, int before, int count) {
        if (s.length()==0) return;
        Log.i("fuck", "on: " + s + ", last char: " + s.toString().substring(s.length() - 1));
        String lastChar = s.subSequence(s.length() - 1, s.length()).toString();
        if (!TextUtils.isEmpty(lastChar) && s.length() > mLastInput.length()) {
            sendCommand(OperationData.OPERATION_TYPE_TEXT, 0, 0, lastChar);
        }
    }
    @Override
    public void afterTextChanged(Editable s) {
        Log.i("fuck", "after: " + s.toString());
    }
});
mSender = new CommandSender(ipStr, port);
mSender.start();
touchBoard.setOnTouchListener(new View.OnTouchListener() {
    @Override
    public boolean onTouch(View v, MotionEvent event) {
        onBoardTouch(event);
        return true;
    }
});
}
private long mDownTime;
private float mDownX;
private float mDownY;

```

					<i>КНТЕУ 121 063-19.МР</i>	Аркуш
						68
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>		

```

private long mUpTime;
private float mUpX;
private float mUpY;
private float mLastMoveX = Float.MAX_VALUE;
private float mLastMoveY = Float.MAX_VALUE;
private float mCurMoveX;

```

## Продовження дод. В

```

private float mCurMoveY;
private long mLastMoveTime;

private void onBoardTouch(MotionEvent event) {
    switch (event.getActionMasked()) {
        case MotionEvent.ACTION_DOWN:
            onSingClick(event, true);
            break;
        case MotionEvent.ACTION_UP:
            onSingClick(event, false);
            break;
        case MotionEvent.ACTION_MOVE:
            onMove(event);
            break;
    }
}

private void onMove(MotionEvent event) {
    int distanceX = 0;
    int distanceY = 0;
    mCurMoveX = event.getX();
    mCurMoveY = event.getY();

    if (mLastMoveX != Float.MAX_VALUE && mLastMoveY != Float.MAX_VALUE) {
        distanceX = (int) (mCurMoveX - mDownX);
        distanceY = (int) (mCurMoveY - mDownY);
    }

    int distance = (int) Math.sqrt(distanceX * distanceX + distanceY * distanceY);

    // send a move command per 0.5 s
    if (distance > 100 || System.currentTimeMillis() - mLastMoveTime > 100) {
        sendCommand(OperationData.OPERATION_MOVE, distanceX, distanceY, null);
        mLastMoveX = mCurMoveX;
        mLastMoveY = mCurMoveY;
        mLastMoveTime = System.currentTimeMillis();
    }
    // sendCommand(OperationData.OPERATION_MOVE, distanceX, distanceY);
}

private void onSingClick(MotionEvent event, boolean down) {
    if (down) {
        mDownTime = System.currentTimeMillis();
    }
}

```

					Аркуш
					<i>KHTEY 121 063-19.MP</i>
Зм.	Аркуш	№ докум	Підпис	Дата	69

```

        mDownX = event.getX();
        mDownY = event.getY();
    } else {
        mUpTime = System.currentTimeMillis();
        mUpX = event.getX();
        mUpY = event.getY();
    }
}

```

## Продовження дод. В

```

private void sendCommand(int operationKind, int x, int y, String input) {
    Message msg = Message.obtain(mSender.mHandler);
    OperationData operation = new OperationData();
    operation.setOperationKind(operationKind);
    if (x!=0) operation.setMoveX(x);
    if (y!=0) operation.setMoveY(y);
    if (input!=null) operation.setInputStr(input);
    Log.i("fuck", "" + operation);
    msg.obj = CommandParser.parseCommand(operation);
    msg.sendToTarget();
}

@Override
protected void onPause() {
    super.onPause();
    mSender.interrupt();
}
}

```

					<i>КНТЕУ 121 063-19.МР</i>	Аркуш
						70
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>		

**Код класу TouchPadActivity**

```

public class TouchPadActivity extends Activity {
    ImageView _touchPad;
    TextView _tvInfo;
    TextView _tvIndex;
    TextView _tvCoord1;
    TextView _tvCoord2;
    private static final int ClickTime = 300;
    private static final int ClickDragTime = 400;
    int _downx0;
    int _downy0;
    int _downx1;
    int _downy1;
    int _lastx0;
    int _lasty0;
    int _lastx1;
    int _lasty1;
    int _x0;
    int _y0;
    int _x1;
    int _y1;

    boolean _motion0;
    boolean _motion1;
    boolean _isClickAndDragging;
    boolean _isPinching;
    boolean _isWheeling;

    Date _dateLastPrimaryClick;
    Date _dateDown0;
    Date _dateDown1;
    int _pixelTolerance;
    int _pixelToleranceSqr;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_board);
        getWindow().addFlags(WindowManager.LayoutParams.FLAG_KEEP_SCREEN_ON);
        Point size = new Point();
        getWindowManager().getDefaultDisplay().getSize(size);
        int width = size.x;
        int height = size.y;
        _pixelTolerance = (int) ((float) (Math.min(width, height)) *
Settings.getPIXEL_TOLERANCE_PERCENTAGE());
        _pixelToleranceSqr = (int) Math.pow(_pixelTolerance, 2);
        _dateLastPrimaryClick = new Date(0);

```

						Аркуш
						71
Зм.	Аркуш	№ докум	Підпис	Дата	<i>КНТЕУ 121 063-19.МР</i>	

						Аркуш
					<i>КНТЕУ 121 063-19.МР</i>	72
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>		

**Код класу KeyboardActivity**

```

public class KeyboardActivity extends Activity {
    OrientationEventListener myOrientationEventListener;
    EditText _editText;
    String _lastTest = "";
    Timer _sendTimer = new Timer();
    final int _timeToWaitBeforeSendMessage = 1500;
    private void Send(final String str) {
        if (_sendTimer != null) {
            _sendTimer.cancel();
            _sendTimer.purge();
        }
        _sendTimer = new Timer();
        _sendTimer.schedule(new TimerTask() {

            @Override
            public void run() {
                SendNow(str);
            }
        }, _timeToWaitBeforeSendMessage);
    }
    private void SendNow(final String str) {
        Connection.Send("keyboard" + str);
        runOnUiThread(new Runnable() {
            public void run() {
                _editText.setText("");
            }
        });
    }
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        boolean reverse = getIntent().getExtras().getBoolean("reverse", false);
        if (reverse)
        {
            setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_REVERSE_LANDSCAPE);
        }
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_board);
        _editText = (EditText) findViewById(R.id.action_bar_activity_content);
        myOrientationEventListener = new OrientationEventListener(this, SensorManager.SENSOR_DELAY_UI) {
            @Override
            public void onOrientationChanged(int arg0) {
                if (arg0 != -1) {
                    if (arg0 > 330 || arg0 < 30) {
                        Release();
                        finish();
                    }
                }
            }
        };
    }
}

```

					<i>КНТЕУ 121 063-19.МР</i>	Аркуш
						73
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>		



						Аркуш
						74
Зм.	Аркуш	№ докум	Підпис	Дата	<i>КНТЕУ 121 063-19.МР</i>	

**Код класу Connection**

```

public class Connection {

    private static DatagramSocket _socket = null;

    protected static void CreateConnection(DatagramSocket socket) {

        _socket = socket;
    }

    protected synchronized static void Send(String str) {
        if (_socket != null && _socket.isConnected() && !_socket.isClosed()) {
            try {
                byte[] bytes = str.getBytes("UTF-8");
                DatagramPacket packet = new DatagramPacket(bytes, bytes.length);
                _socket.send(packet);
            } catch (Exception e) {
                e.printStackTrace();
                Log.i("air", e.getMessage());
            }
        }
    }

    public static boolean IsConnected() {
        if (_socket == null)
            return false;
        return _socket.isConnected();
    }

    protected static void Disconnect() {
        if (_socket == null)
            return;
        if (_socket.isConnected()) {
            Send(Commands.Bye);
            // _socket.disconnect();
        }
        _socket.close();
    }
}

```

					<i>КНТЕУ 121 063-19.МР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		75

**Код класу NetUtils**

```

public class NetUtils {

    public static final String SERVER_IP = "192.168.1.2";
    public static final int SERVER_PORT = 6221;
    public static boolean pingServer(String server_ip){
        Socket socket = null;
        try {
            Log.i("fuck", "" + server_ip);
            socket = new Socket(server_ip==null?SERVER_IP:server_ip, SERVER_PORT);
            return true;
        } catch (IOException e) {
            e.printStackTrace();
            return false;
        } finally {
        }
    }
}

```

						Аркуш
						76
Зм.	Аркуш	№ докум	Підпис	Дата	<i>КНТЕУ 121 063-19.МР</i>	

**Код класу Scan**

```

public class Scan {
    public static int TIMEOUT = 30000;
    public static ErrorCode ERROR;
    private static OnScanInterruptedListener _listener;
    public static void SetOnScanInterrupted(OnScanInterruptedListener listener) {
        _listener = listener;
    }
    public DatagramSocket ScanIps(Context context) {
        return ScanIpsBlocking(context);
    }
    public static void ScanIpsAsync(final Context context) {
        Thread t = new Thread(new Runnable() {
            public void run() {
                DatagramSocket ret = ScanIpsBlocking(context);
                if (ret == null) {
                    _listener.ErrorHappened(ERROR);
                } else {
                    _listener.IpFound(ret);
                }
            }
        });
        t.start();
    }
    private static DatagramSocket ScanIpsBlocking(Context context) {
        ERROR = ErrorCode.None;
        DatagramSocket ret = null;
        long startScan = new Date().getTime();
        ConnectivityManager connManager = (ConnectivityManager)
context.getSystemService(Context.CONNECTIVITY_SERVICE);
        WifiManager wim = (WifiManager) context.getSystemService(android.content.Context.WIFI_SERVICE);
        int ipAd = wim.getConnectionInfo().getIpAddress();
        final String ipString = String.format(Locale.US, "%d.%d.%d.%d", (ipAd & 0xff), (ipAd >> 8 & 0xff), (ipAd >>
16 & 0xff), (ipAd >> 24 & 0xff));
        InetAddress ipAddress = null;
        byte[] ip = null;
        try {
            ipAddress = InetAddress.getByAddress(ipString);
        } catch (UnknownHostException e1) {
            e1.printStackTrace();
        }
        if (ipAddress != null)
            ip = ipAddress.getAddress();
        else
            return null;
        for (int i = 1; i <= 254; i++) {
            long now = new Date().getTime();

```

					<i>КНТЕУ 121 063-19.МР</i>	Аркуш
						77
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>		

```

if (now - startScan > TIMEOUT) {
    ERROR = ErrorCode.Time_Out;
    ret = null;
}

```

## ДОДАТОК К

### Код класу MotionProvider

```

public class MotionProvider implements SensorEventListener {
    private static boolean _loaded = false;
    private static long _lastSensorChange;
    private static int _sensorType = Sensor.TYPE_GYROSCOPE;
    private static Boolean _calibrating = false;
    private static float[] _calibratingX;
    private static float[] _calibratingY;
    private static int _calibrationSamples = 50;
    private static int _calibrationStep = -1;
    private static MotionProvider _self = new MotionProvider();
    private static OnMotionChangedListener _motionListener;
    private static OnCalibrationFinishedListener _calibrationFinishedListener;
    protected static void SetOnCalibrationFinished(OnCalibrationFinishedListener listener) {
        _calibrationFinishedListener = listener;
    }
    protected static void ReleaseCalibrationListener() {
        _calibrationFinishedListener = null;
    }
    protected static void SetOnMotionChanged(OnMotionChangedListener listener) {
        _motionListener = listener;
    }
    protected static void ReleaseMotionListener() {
        _motionListener = null;
    }
    protected static void Calibrate() {
        _calibrating = true;
    }
    protected static void RegisterEvents(Context context) {
        if (_loaded)
            return;
        SensorManager sm = (SensorManager)
context.getSystemService(android.content.Context.SENSOR_SERVICE);
        List<Sensor> sensors = sm.getSensorList(_sensorType);
        if (sensors.size() > 0) {
            sm.registerListener(_self, sensors.get(0), SensorManager.SENSOR_DELAY_FASTEST);
            _loaded = true;
        }
    }
    protected static void UnregisterEvents(Context context) {
        SensorManager sm = (SensorManager)
context.getSystemService(android.content.Context.SENSOR_SERVICE);
        sm.unregisterListener(_self);
    }
}

```

					<i>KHTEY 121 063-19.MP</i>	Аркуш
						78
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>		

```

        _loaded = false;
    }
    @Override
    public void onAccuracyChanged(Sensor arg0, int arg1) {
    }
    @Override
    public void onSensorChanged(SensorEvent event) {

```

					<i>КНТЕУ 121 063-19.МР</i>	Арқуш
Зм.	Арқуш	№ докум	Підпис	Дата		79

						Аркуш
					<i>КНТЕУ 121 063-19.МР</i>	80
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>		