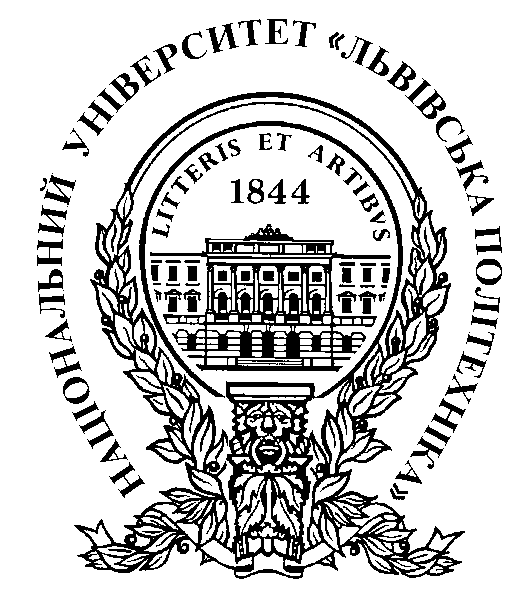
Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Інститут комп’ютерних наук та інформаційних технологій

Кафедра автоматизованих систем управління



Лабораторна робота №1

з дисципліни

«Програмування мобільних пристроїв»

Виконала:

ст. групи КН-311

Стефанишин Віта

Прийняв:

Григорович В. Г.

Львів 2019

Лабораторна робота №1

**Тема:** налаштування середовища програмування Android Studio або XCode. Робота з Git.

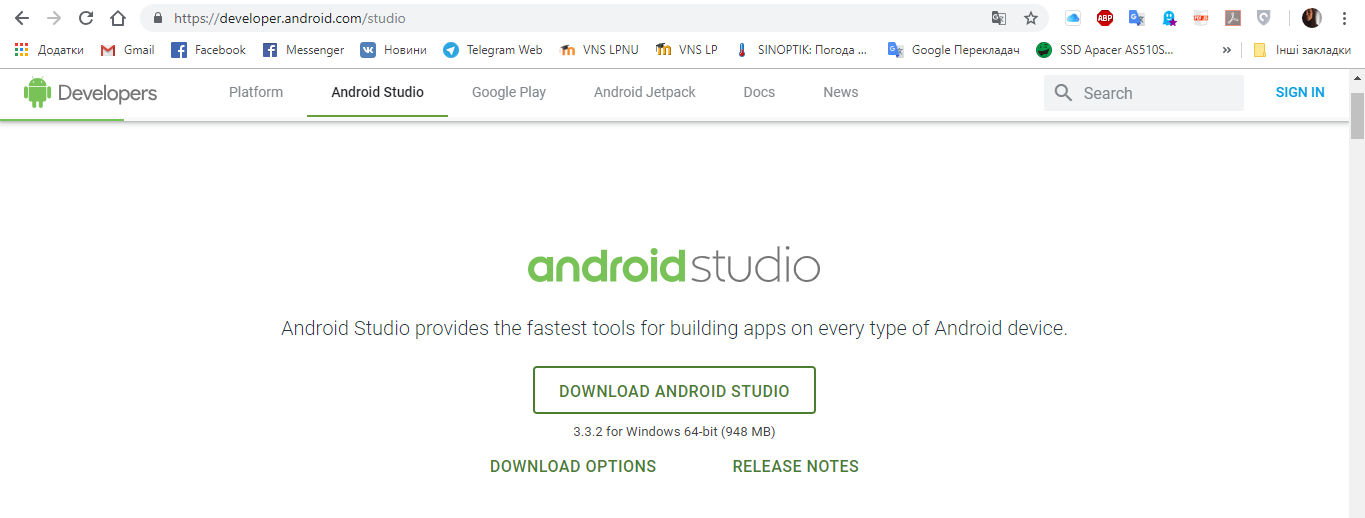
**Мета:** інсталювати Android Studio або XCode, навчитись працювати із засобами відлагодження програм.

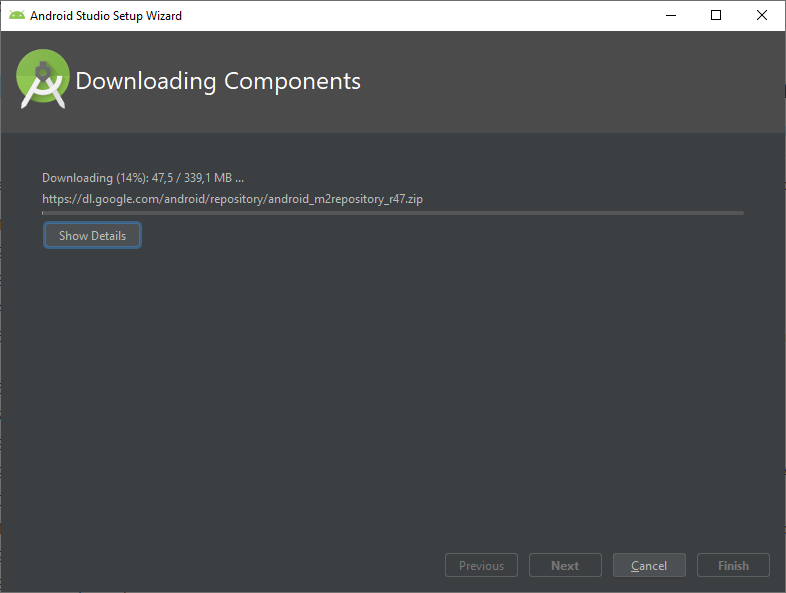
Завдання:

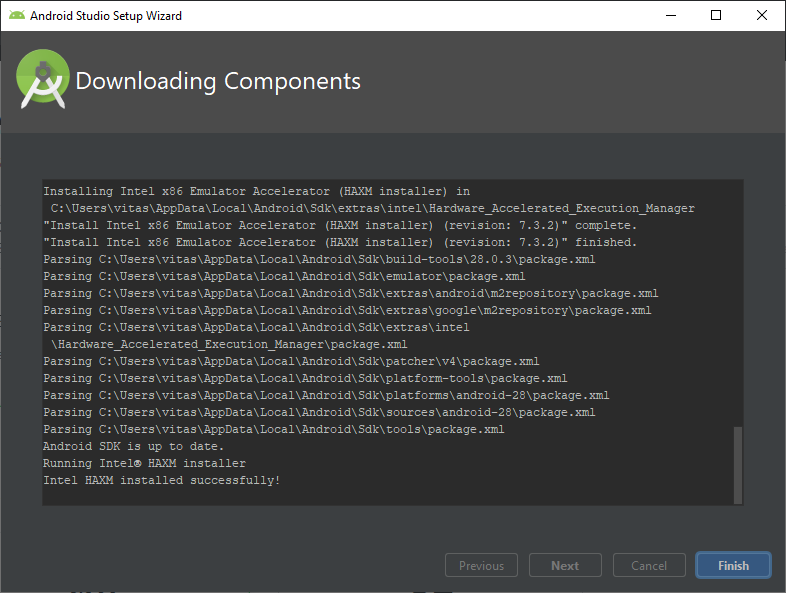
1. Встановити на власний комп'ютер Android Studio (Windows, Linux, Mac) або XСode (Mac)
2. Вивчити команди системи контролю версій git (див. Basic Git commands )
3. Зареєструватися на GitHub.
4. Встановити git кліент на вибір: SourceTree, TortoiseGit, SmartGit або інший.
5. Створити новий проект для Android чи iOS або склонувати проект-приклад згідно до індивідуального завдання (див. нижче)
6. Розібратися зі структурою проекту
7. Розібратися зі структурою файлу AndroidManifest.xml
8. Запустити проект на емуляторі
9. Під’єднати мобільний телефон з ОС “Android” за допомогою USB і увімкнути відлагодження через USB (ADB)
10. Запустити проект на телефоні
11. Налаштувати запуск тестів. Створити хибний тест. Запустити і пересвідчитися, що хибний тест не пройшов
12. Знайти основну діяльність (MainActivity), перейти на неї
13. Вивчити роботу команди ADB
14. Вивчити клас Log для виводу відлагоджувальних повідомлень
15. Перейти на клас MainActivity, знайти метод onCreate, додати лог Log.d(TAG, "Activity onCreate"), поставити Breakpoint на цьому методі.
16. Запустити проект в режимі відлагодження, розібратися з панеллю відлагодження
    * Знайти, де знаходиться програма adb (під Linux/Mac команда find ~/Library/Android/ -name adb), та встановити режим виведення логів VERBOSE для MainActivity (<шлях до adb>/adb shell setprop log.tag.MainActivity VERBOSE)
17. Перейти в Android Monitor → LogCat, Відфільтрувати журнал по рядку “Activity ”
18. Перезапустити прогаму
19. Перейти в LogCat, дослідити, у які стани переходять класи діяльностей програми (Activity) при перемиканні на інші програми
20. Написати звіт, розмістити скріншоти на DropBox (або інший ресурс)

Хід роботи

1. Встановлююю android studio 3.3.2 та необхідні програмні компоненти для моєї версії Windows (10, 64-bit).



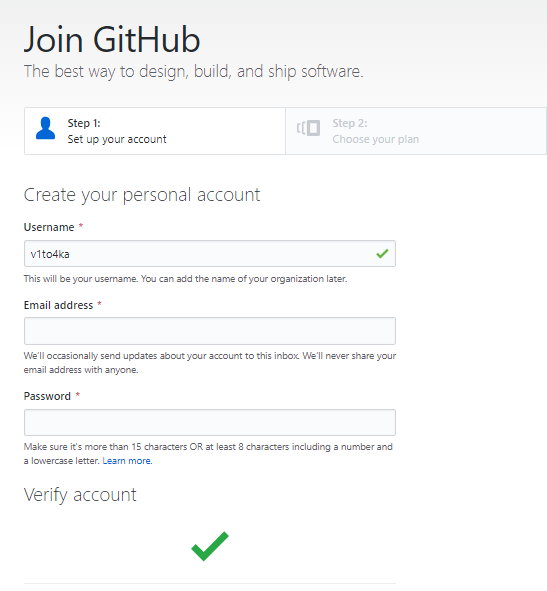




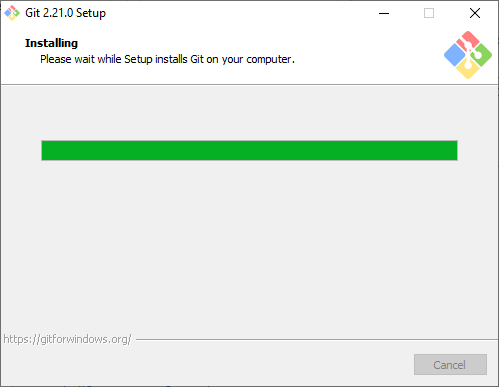
2. Команди системи контролю версій git:

|  |  |  |
| --- | --- | --- |
| Git task | Notes | Git commands |
| **Налаштування Git** | Configure the author name and email address to be used with your commits.  Note that Git [strips some characters](http://stackoverflow.com/questions/26159274/is-it-possible-to-have-a-trailing-period-in-user-name-in-git/26219423#26219423) (for example trailing periods) from user.name. | git config --global user.name "Sam Smith"  git config --global user.email sam@example.com |
| Create a new local repository | **Створення нового репозиторію** | git init |
| [Check out a repository](https://www.atlassian.com/git/tutorials/setting-up-a-repository/git-clone) | **Клонування репозиторію** | git clone /path/to/repository |
| [Add files](https://www.atlassian.com/git/tutorials/saving-changes#git-add) | Add one or more files to staging (index): | git add <filename>  git add \* |
| [Commit](https://www.atlassian.com/git/tutorials/saving-changes#git-commit) | Коміт відображає стан нашого репозиторія у момент, коли він був зроблений. Це як знімок, до якого ми можемо повернутися і побачити, як усе було, коли ми його тільки зробили. | git commit -m "Commit message" |
| Коміт відображає стан нашого репозиторія у момент, коли він був зроблений. Це як знімок, до якого ми можемо повернутися і побачити, як усе було, коли ми його тільки зробили. | git commit -a |
| [Push](https://www.atlassian.com/git/tutorials/syncing#git-push) | **Завантаження на сервер** | git push origin master |
| [Status](https://www.atlassian.com/git/tutorials/inspecting-a-repository#git-status) (перевірка статусу) | List the files you've changed and those you still need to add or commit: | git status |
| [Connect to a remote repository](https://www.atlassian.com/git/tutorials/syncing#git-remote) | **Підключення до дистанційного репозиторія** | git remote add origin <server> |
| **Підключення до дистанційного репозиторія** | git remote -v |
| [Branches](https://www.atlassian.com/git/tutorials/using-branches) | Create a new branch and switch to it: | git checkout -b <branchname> |
| **Перехід між гілками** | git checkout <branchname> |
| **Створення нової гілки** | git branch |
| Delete the feature branch: | git branch -d <branchname> |
| Push the branch to your remote repository, so others can use it: | git push origin <branchname> |
| Push all branches to your remote repository: | git push --all origin |
| Delete a branch on your remote repository: | git push origin :<branchname> |
| [Update from the remote repository](https://www.atlassian.com/git/tutorials/syncing) | **Отримання змін з серверу** | git pull |
| **З'єднання гілок** | git merge <branchname> |
| Tags | **Перевірка різниці між комітами** | git log |  |
| Push all tags to remote repository: | git push --tags origin |
| [Undo local changes](https://www.atlassian.com/git/tutorials/undoing-changes) | **Повернення файлу до попереднього стану** | git checkout -- <filename> |
| Instead, to drop all your local changes and commits, fetch the latest history from the server and point your local master branch at it, do this: | git fetch origin  git reset --hard origin/master |

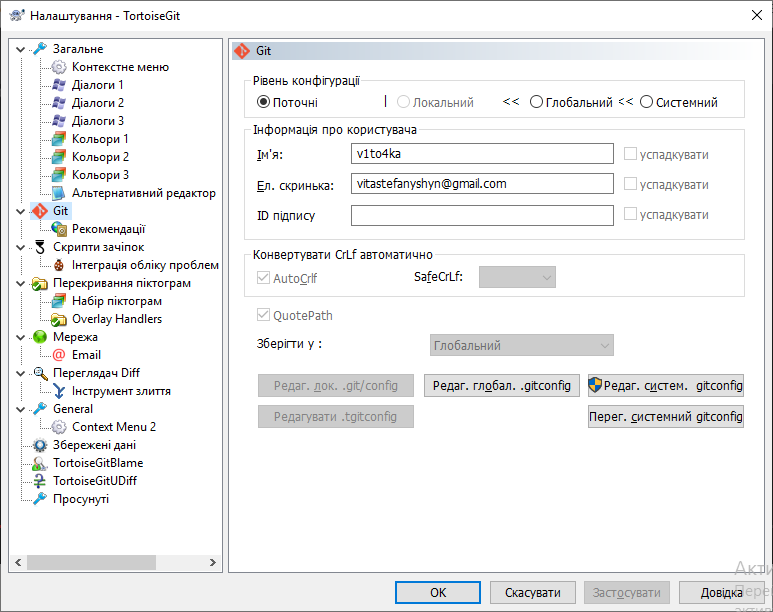
3) Реєструюсь на GitHub:



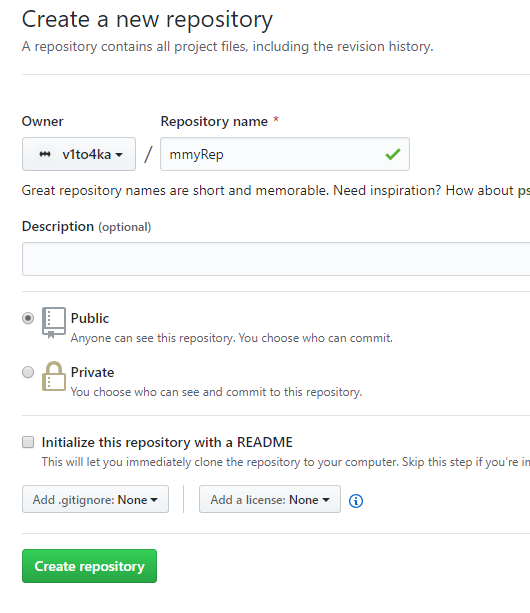
4) Встановлюю git кліент на вибір: TortoiseGit.

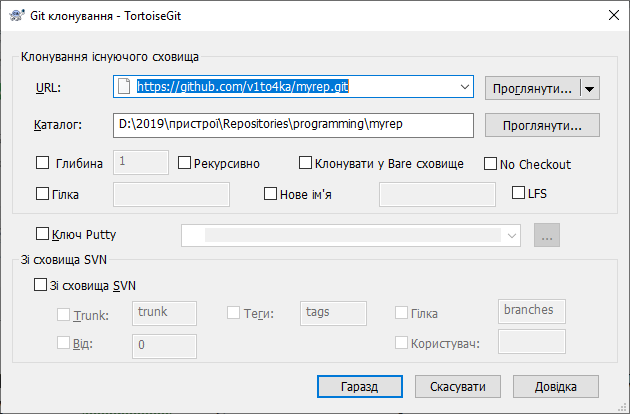
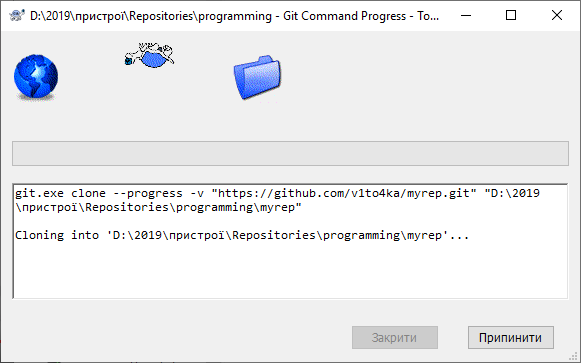
 

Прив’язую свій аккаунт до git-клієнта.

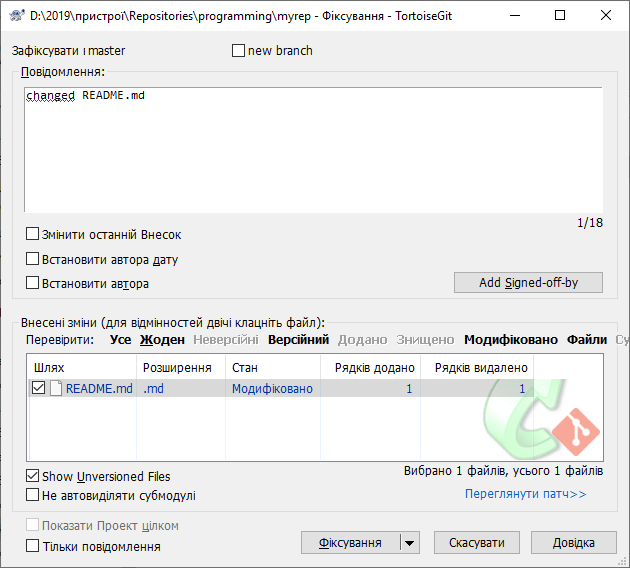
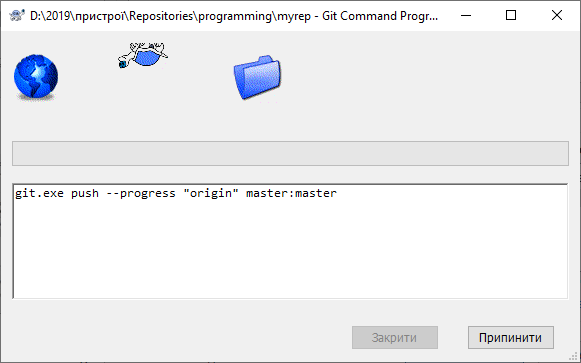


Створюю власний репозиторій:

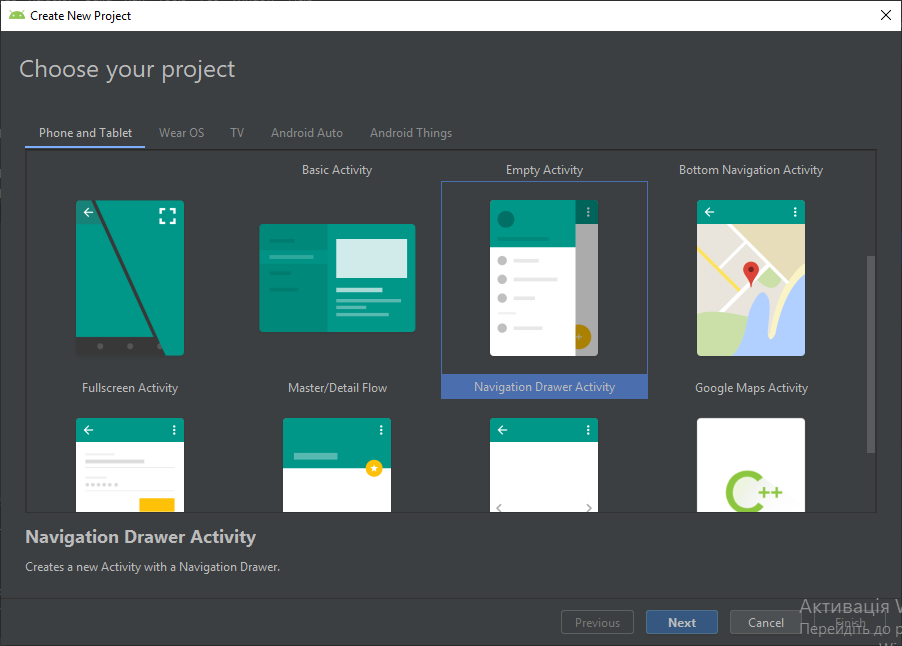


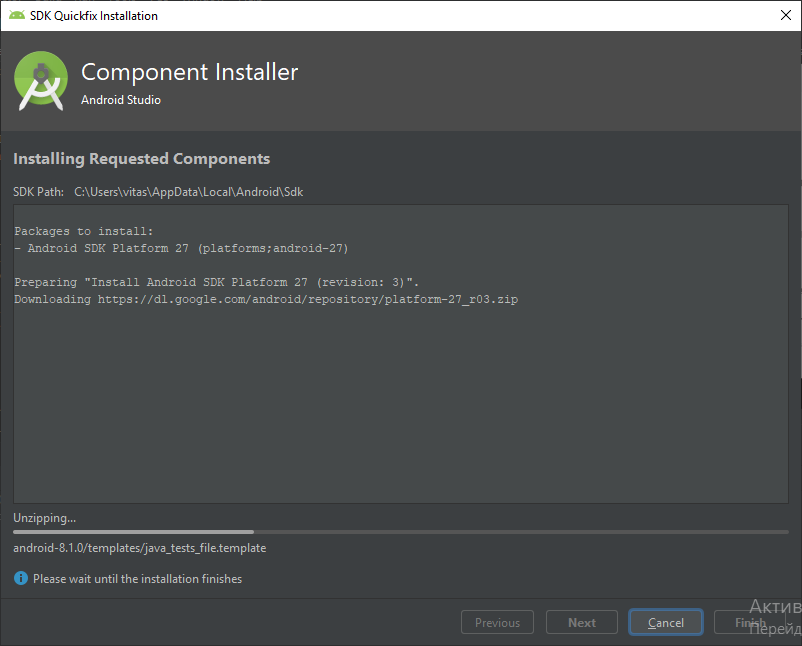
 

Роблю зміни у репозиторії та фіксую їх:

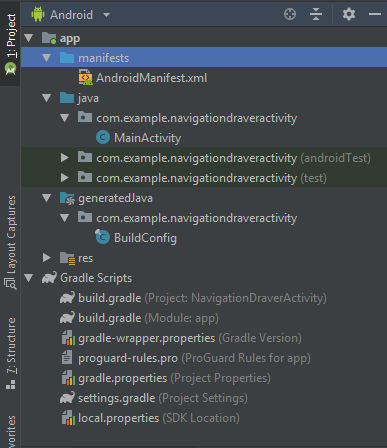
 

5) Створюю новий проект для Android або склонувати проект-приклад згідно до індивідуального завдання : Navigation Drawer Activity.



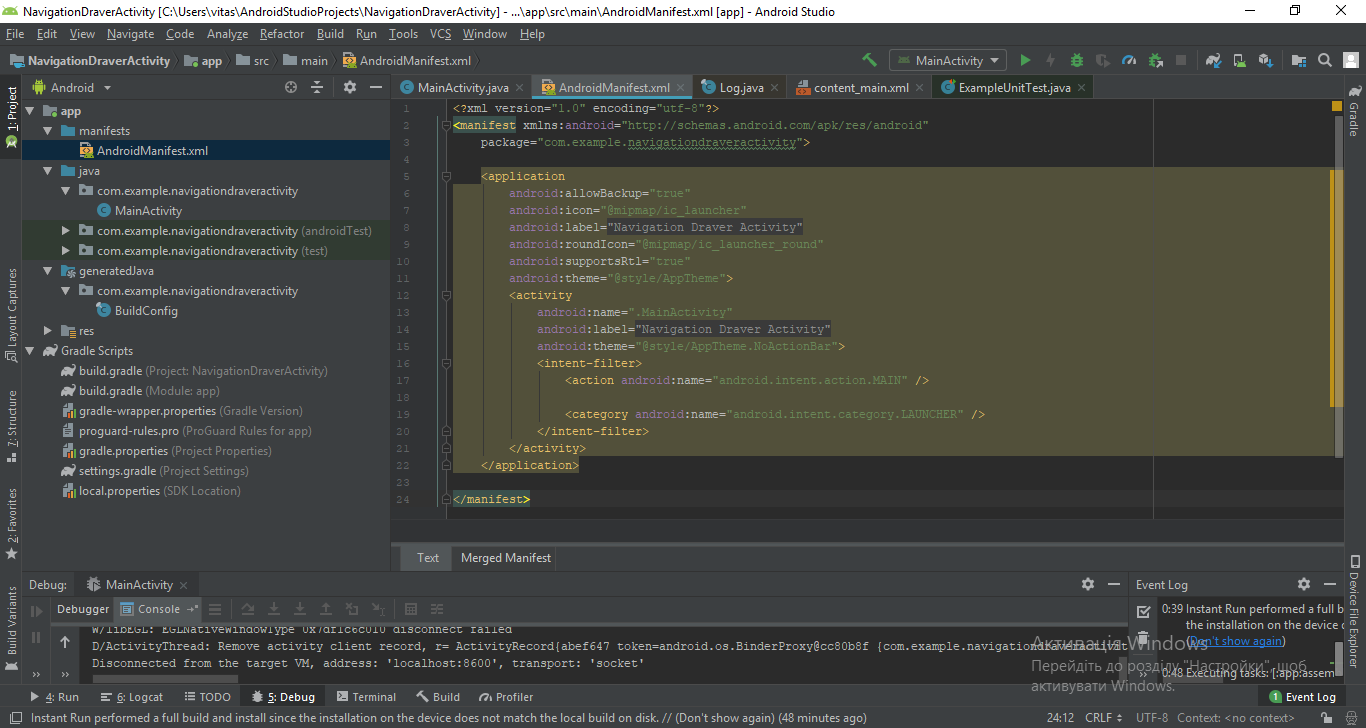


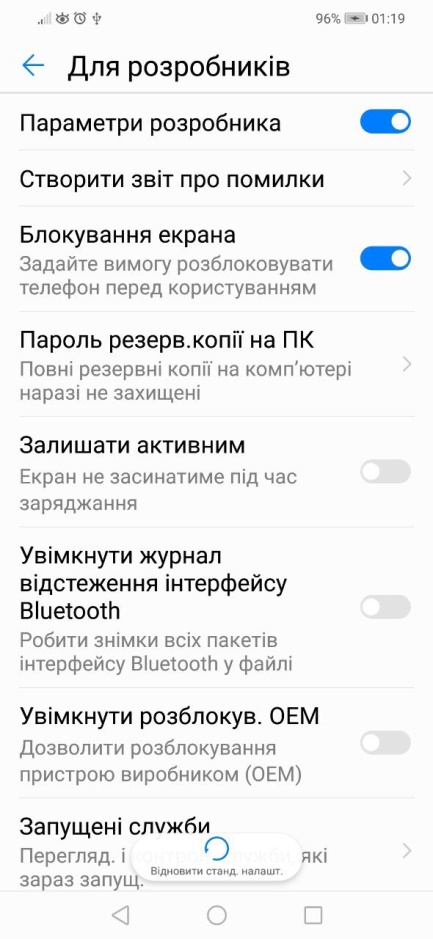
6) Структура Android-проекту:

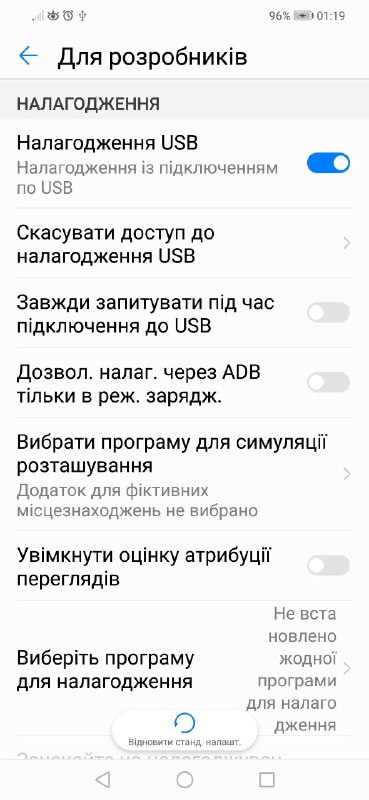


7) Структура файлу AndroidManifest.xml:

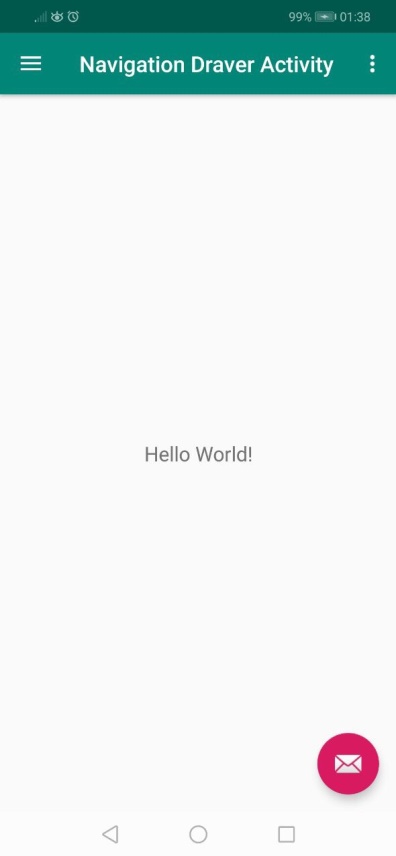
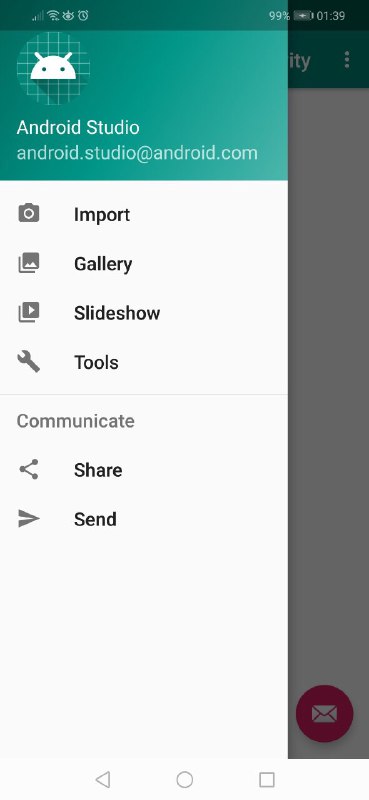
У manifests згруповані маніфести наших модулів. AndroidManifest - одна з ключових частин програми. Маніфест описує, з яких «частин» складається додаток. Він являє собою XML-файл, в якому описані розширення додатків, «екрани», сервіси, метадані і т.д. Приклад AndroidManifest.xml з пустого проекту:



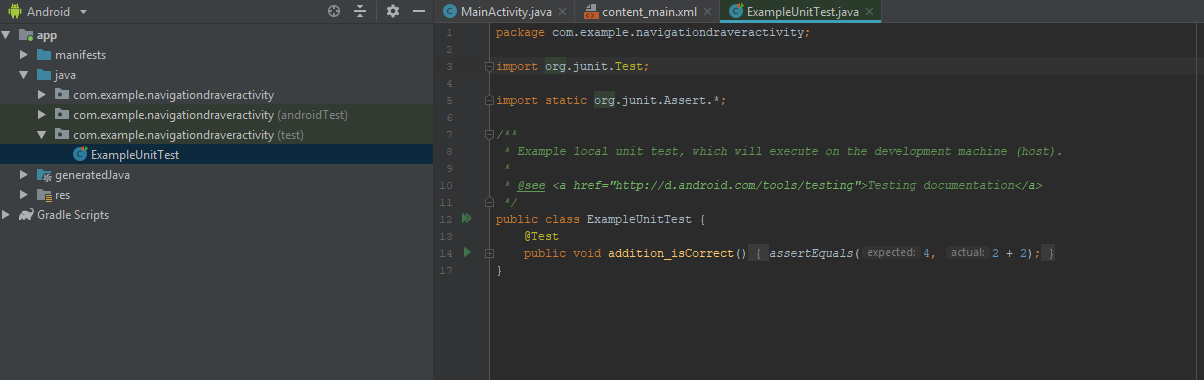
9) Під’єдную мобільний телефон з ОС “Android 8.1.0” за допомогою USB і вмикаю відлагодження через USB (ADB):

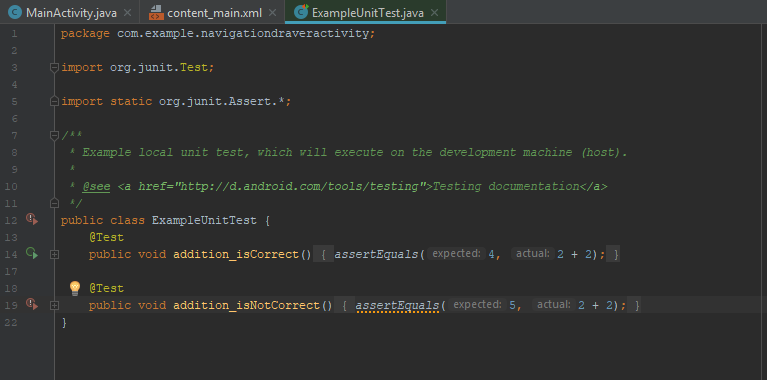


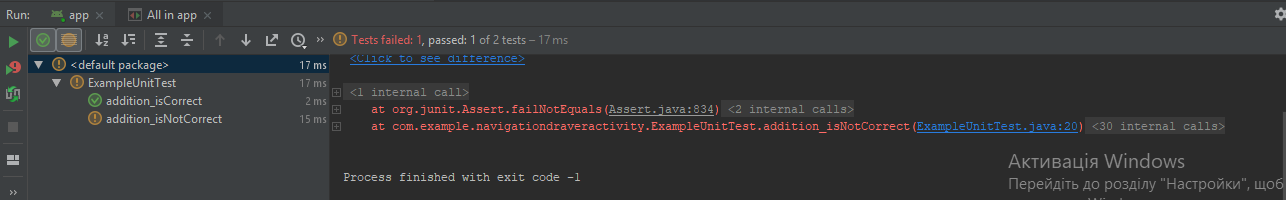
10) Запускаю проект на пристрої Huawei INE-LX1 (Android 8.1.0, API 27):

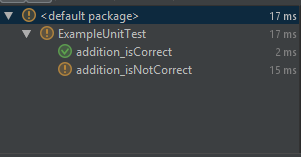
 

11) Налаштовую запуск тестів. Створюю хибний тест. Запускаю і пересвідчуюсь, що хибний тест не пройшов.



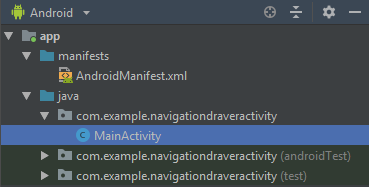






Тест 1 пройшов успішно, 2 – ні.

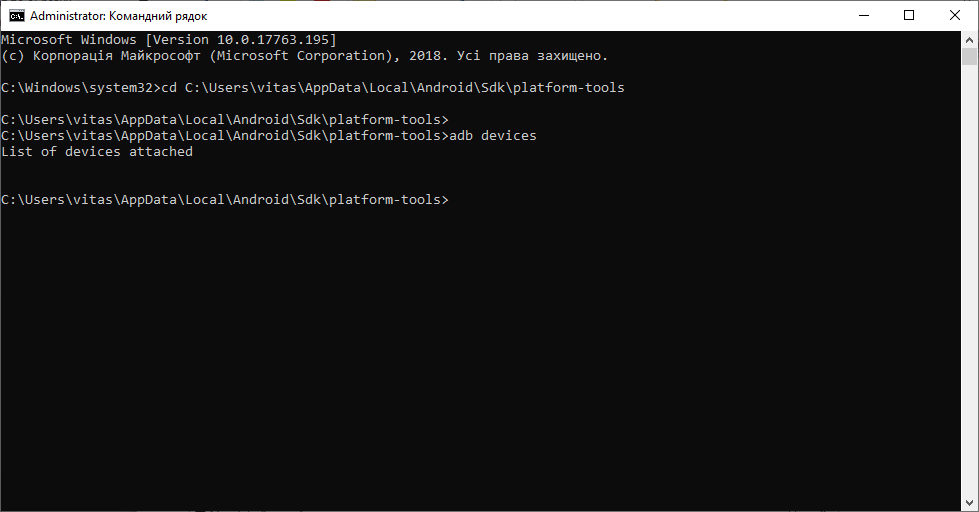
12) MainActivity – основна діяльність проекту.



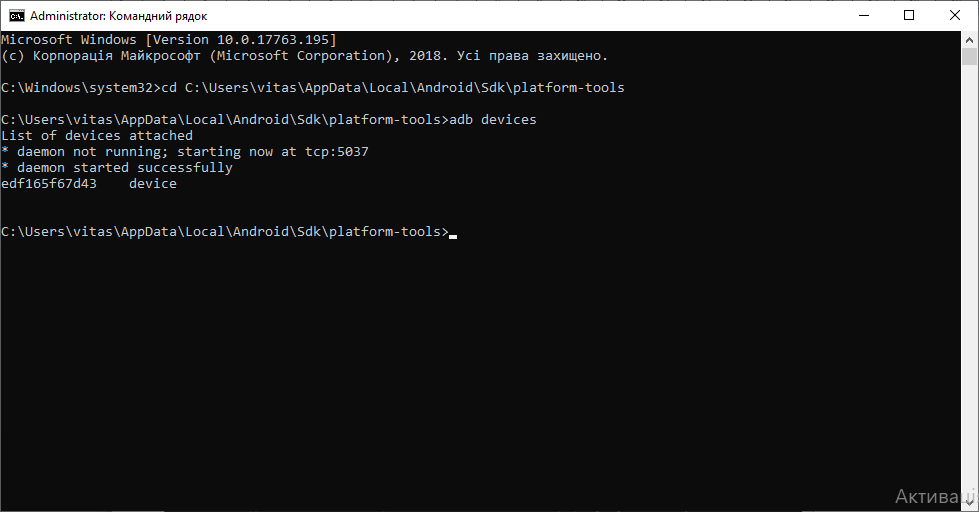
13) Міст налагодження Android (adb) - це універсальний інструмент командного рядка, який дозволяє спілкуватися з пристроєм.

Команда adb devices у каталозі **android\sdk\platform-tools\** показує список всіх

підключених пристроїв:



При відключеному пристрої



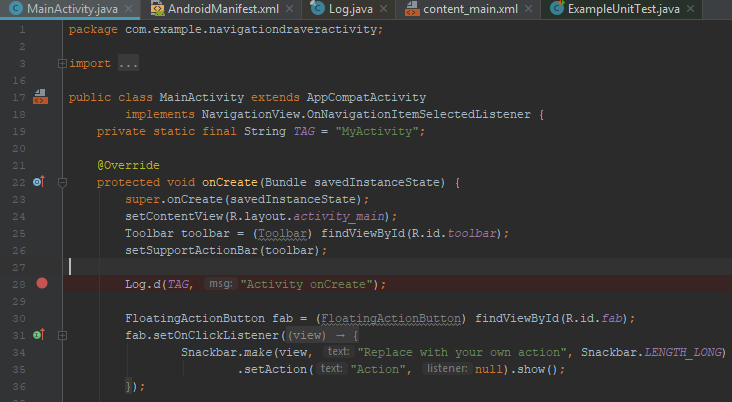
При підключеному пристрої.

14) Клас Log - для виводу відлагоджувальних повідомлень. [Log](https://developer.android.com/reference/android/util/Log.html)Клас дозволяє створювати лог - повідомлення , які з'являються в LogCat.

* [Log.e(String, String)](https://developer.android.com/reference/android/util/Log.html#e(java.lang.String,%20java.lang.String)) (помилка, error)
* [Log.w(String, String)](https://developer.android.com/reference/android/util/Log.html#w(java.lang.String,%20java.lang.String)) (увага, warn)
* [Log.i(String, String)](https://developer.android.com/reference/android/util/Log.html#i(java.lang.String,%20java.lang.String)) (інформація, info)
* [Log.d(String, String)](https://developer.android.com/reference/android/util/Log.html#d(java.lang.String,%20java.lang.String)) (налагоджувати, debug)
* [Log.v(String, String)](https://developer.android.com/reference/android/util/Log.html#v(java.lang.String,%20java.lang.String)) (багатослівний, verbose)

[Log](https://developer.android.com/reference/android/util/Log.html)Клас дозволяє створювати лог - повідомлення , які з'являються в LogCat

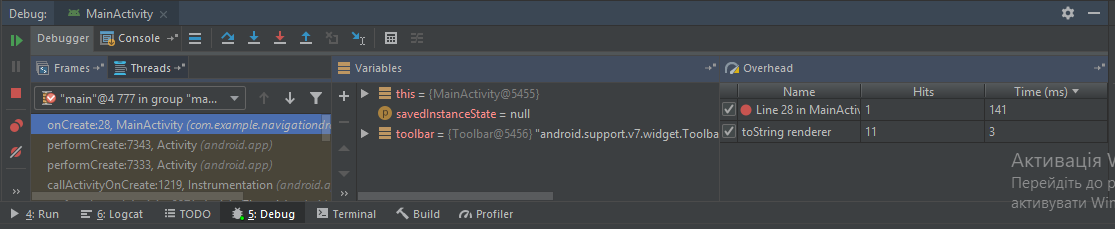
15) У класі MainActivity, у методі onCreate, додаю лог Log.d(TAG, "Activity onCreate"):



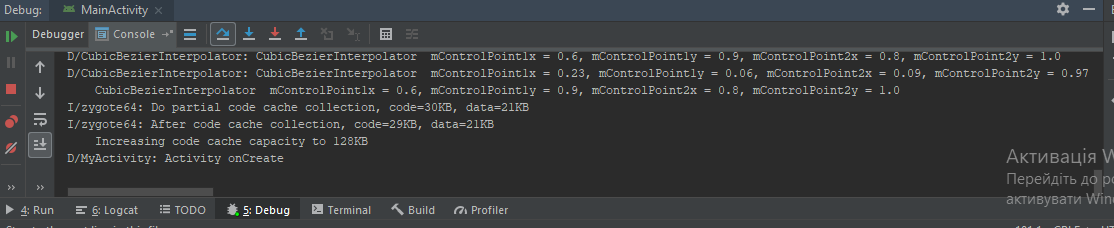
Cтавлю Breakpoint на цьому методі:



16) Запускаю проект в режимі від лагодження. Панель відлагодження:

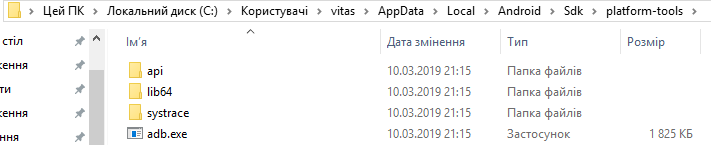


Зайшла у кінець методу:



17) Програма adb знаходиться за шляхом:

C:\Users\vitas\AppData\Local\Android\Sdk\platform-tools

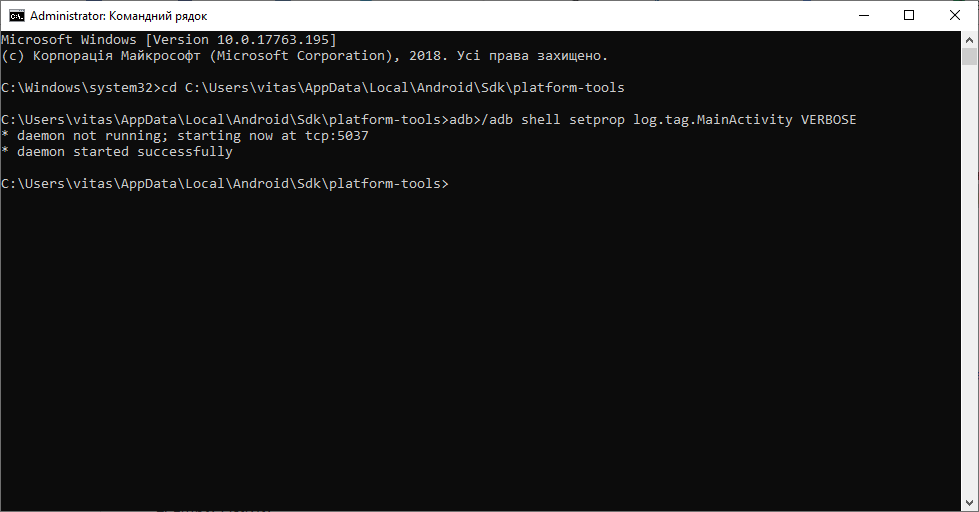


Встановлюю режим виведення логів VERBOSE для MainActivity

(<шлях до adb>/adb shell setprop log.tag.MainActivity VERBOSE):

Коли я запускаю adb-клієнт, він спочатку перевіряє, чи є вже запущений сервер adb. Якщо не існує, він запускає серверний процес. Коли сервер запускається, він зв'язується з локальним TCP-портом 5037 і слухає команди, відправлені з adb-клієнтів - всі клієнти adb використовують порт 5037 для зв'язку з сервером adb.

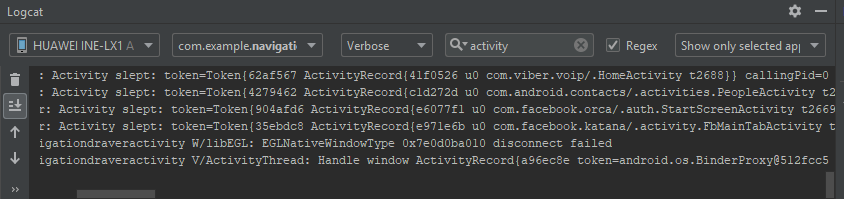
Потім сервер встановлює підключення до всіх запущених пристроїв.



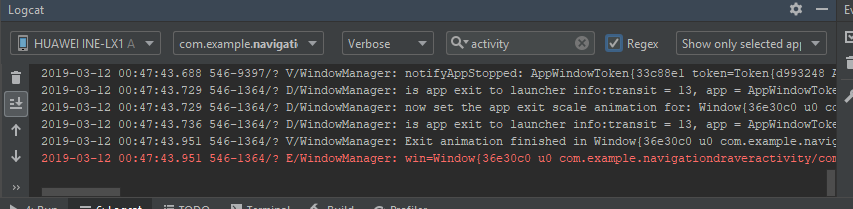
18. Переходжу в Android Monitor → LogCat,

Відфільтровую журнал по рядку “Activity ”

**Verbose:** Показати всі повідомлення журналу (за замовчуванням).



19-20. Після перезапуску Android Studio, переходжу в LogCat, досліджую, в які стани переходять класи діяльностей програми (Activity) при перемиканні на інші програми. При виході з додатку, журнал отримує наступне повідомлення:



Всі скріни розміщені за посиланням:

*https://drive.google.com/drive/folders/1rD1OIxF7KQgwzGuU2RyZVDUiJ\_8\_5XbI?usp=sharing*

**Висновки:** в результаті виконання цієї лабораторної роботи я інсталювала Android Studio, навчилась використовувати репозиторій, робити у ньому зміни, створила перший проект, запустивши його на Android, навчитись працювати із засобами відлагодження програм.