

Konfiguracja



eeStec
AGH Kraków



Na początek:

- Pierwszym krokiem będzie instalacja Java jdk7 (pliki znajdują się w folderze jdk7), oczywiście dobieramy odpowiednią wersję do systemu operacyjnego.
- Drugim krokiem będzie instalacja Java jre7 – podobnie jak punkt wyżej
- Instalacja Android Studio (można również skopiować folder z instalacją) – folder jest nazwany „ECLIPSE”
- Skopiowanie ant-a do kompilowania aplikacji (kopiujemy folder do „C:\Program Files\Java”)
- Ostatni etap to konfiguracja zmiennych środowiskowych systemu

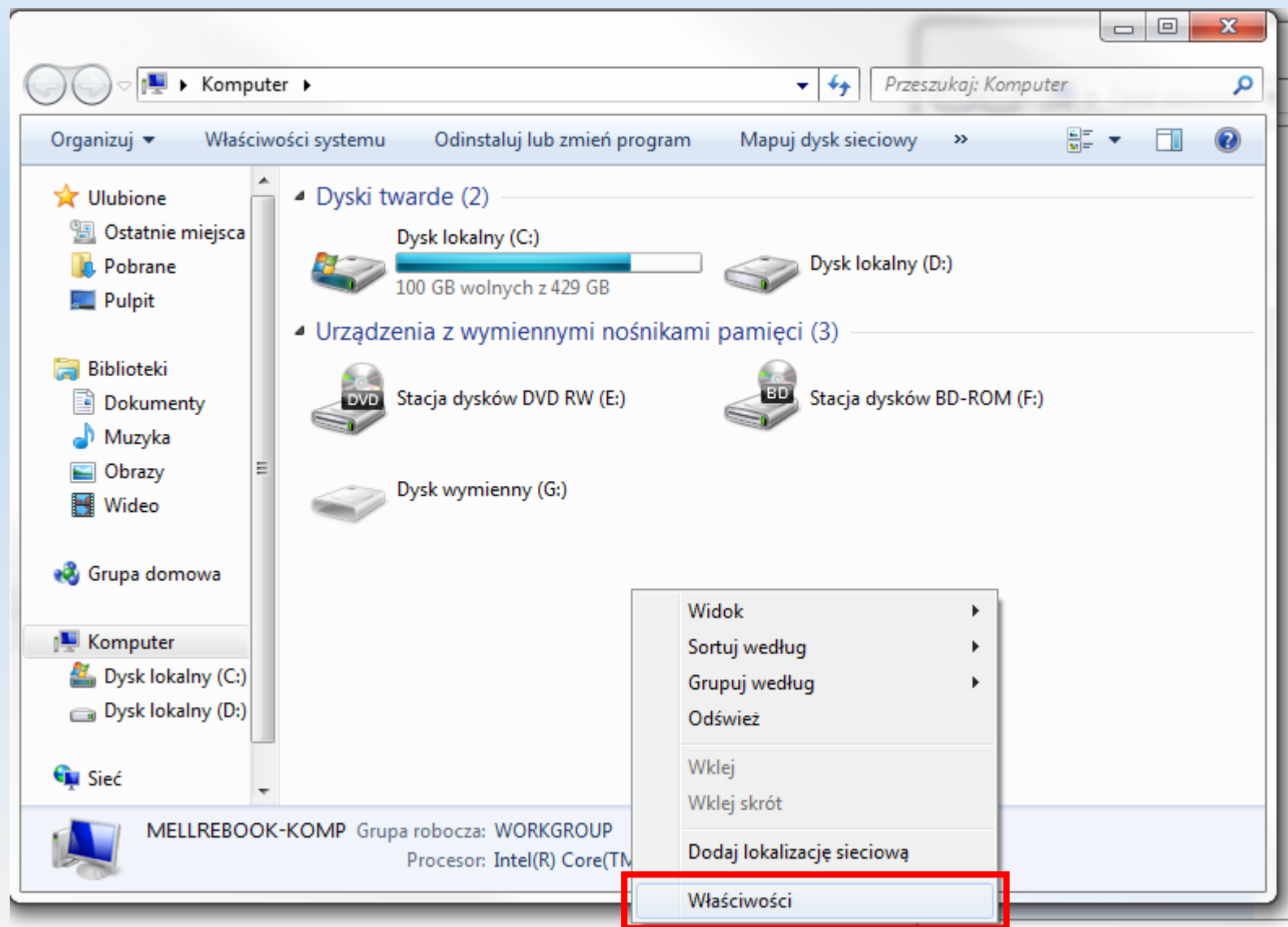
Tak więc do dzieła!

Instalacja oprogramowania...

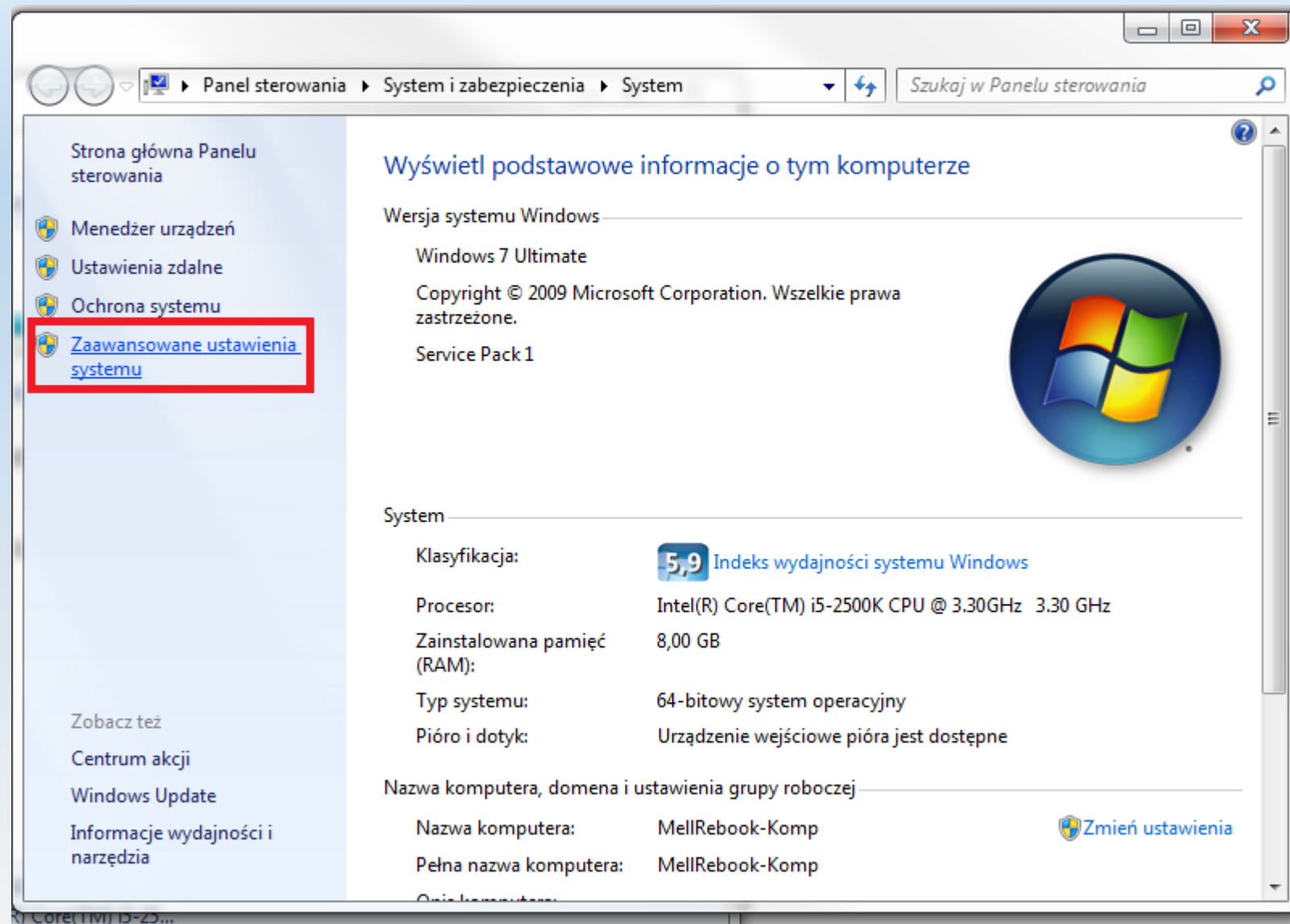


Konfiguracja zmiennych środowiskowych

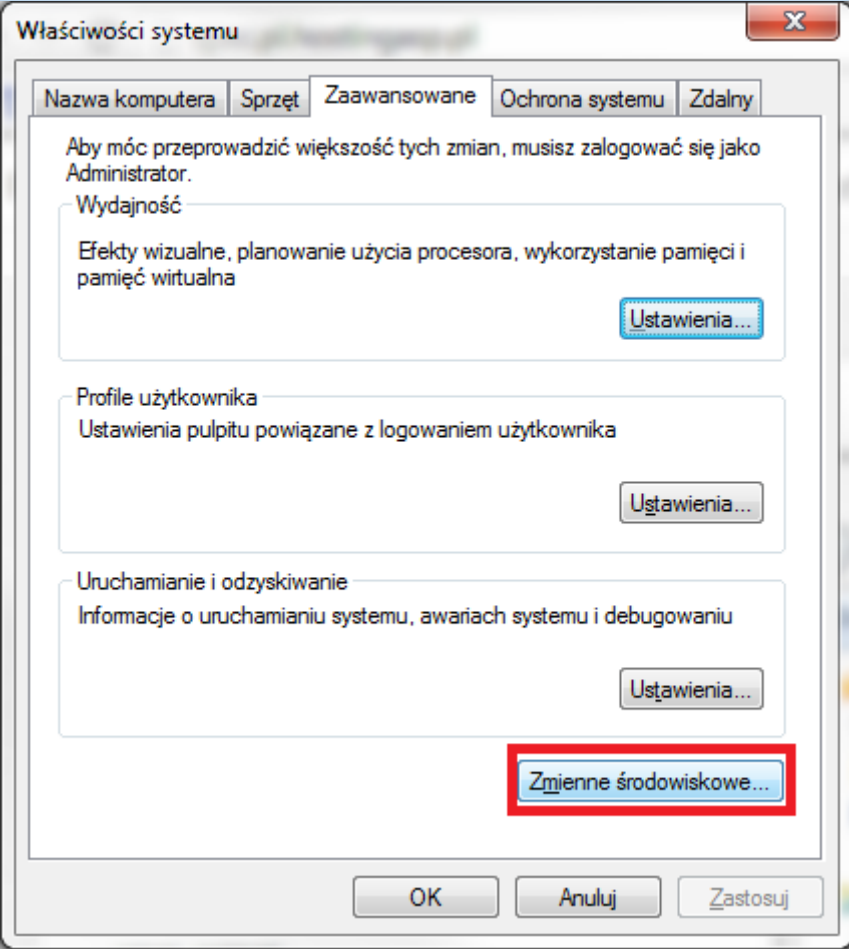
W „Mój komputer” należy dostać się do „Właściwości”



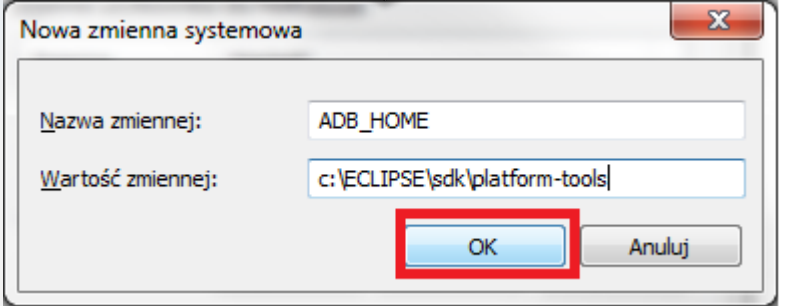
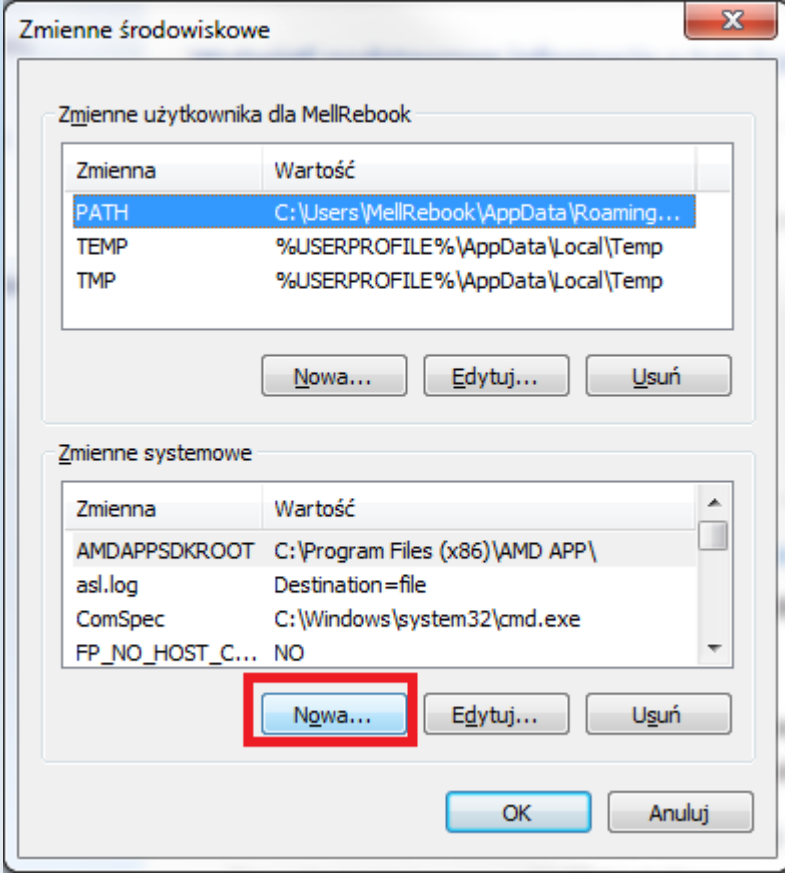
W właściwościach systemu udajemy się do „Zaawansowanych ustawień systemu”



Następnie udajemy się do „Zmiennych środowiskowych”



I dodajemy nową zmienną



Lista zmiennych środowiskowych:

- „ADB_HOME” – „c:\ECLIPSE\sdk\platform-tools”
- „ANDROID_HOME” – „c:\ECLIPSE\sdk\tools”
- „ANT_HOME” – „C:\Program Files\Java\ant”
- „CSCRIPT_PATH” – „C:\Windows\System32”
- „JAVA_HOME” – „C:\Program Files\Java\jdk1.7.0_51”
- „JDK_HOME” – „C:\Program Files\Java\jdk1.7.0_51”
- „PATH” – „%ADB_HOME%;%ANT_HOME%\bin;%JDK_HOME%\bin;%ANDROID_HOME%;%CSCRIPT_PATH%;”

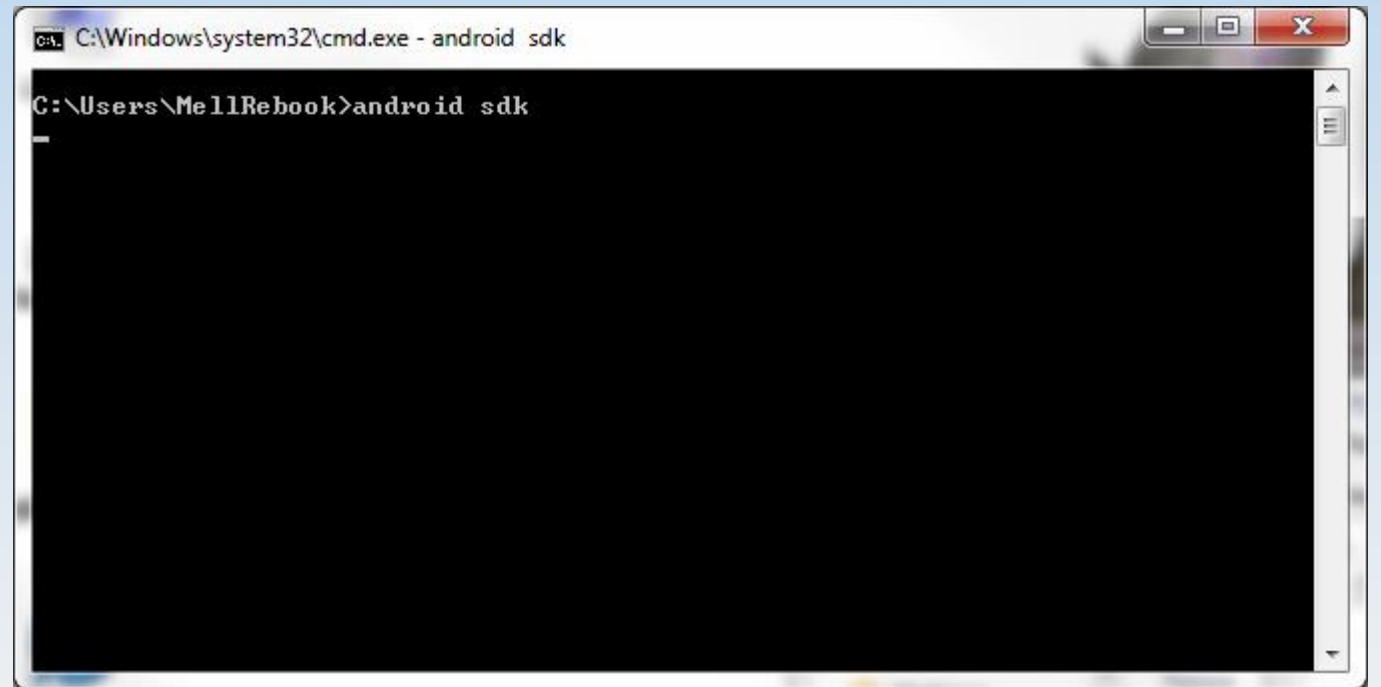
UWAGA: domyślnie po zainstalowaniu Android studio, zamiast „c:\ECLIPSE\...” znajduje się w miejscu zainstalowania programu (domyślnie: „c:\Program Files (x86)\Android\android-studio\”).

Lista zmiennych znajduje się w pliku tekstowym zmienne.txt

Aktualizacja emulatora



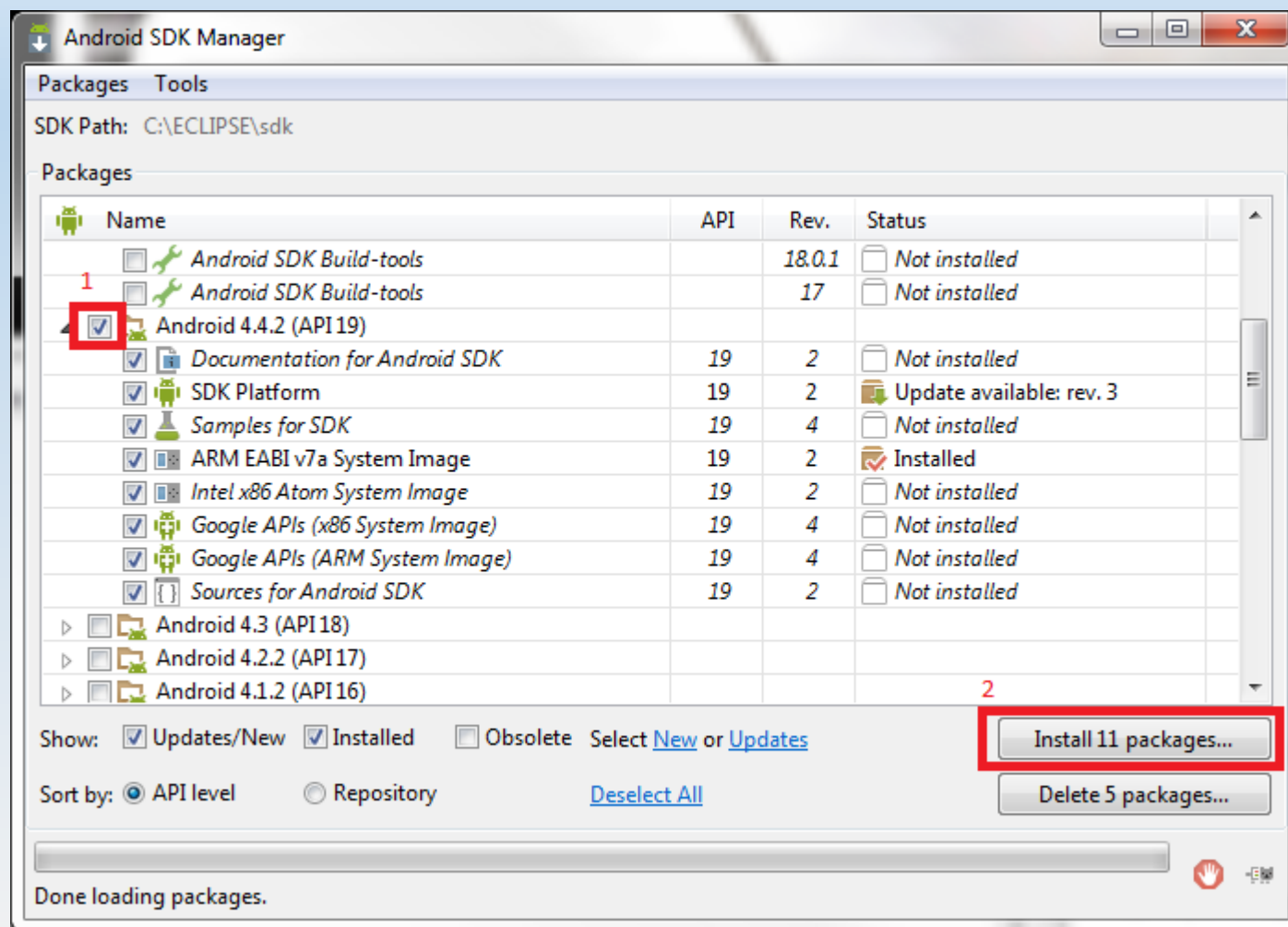
Odpalamy wiersz poleceń i wpisujemy
„android sdk”

A screenshot of a Windows command prompt window. The title bar at the top reads "C:\Windows\system32\cmd.exe - android sdk". The command prompt shows the current directory as "C:\Users\MellRebook>" followed by the command "android sdk" entered. The rest of the window is black, indicating no output has been displayed yet.

```
C:\Windows\system32\cmd.exe - android sdk  
C:\Users\MellRebook>android sdk
```

Wybieramy interesujące nas paczki i instalujemy je.

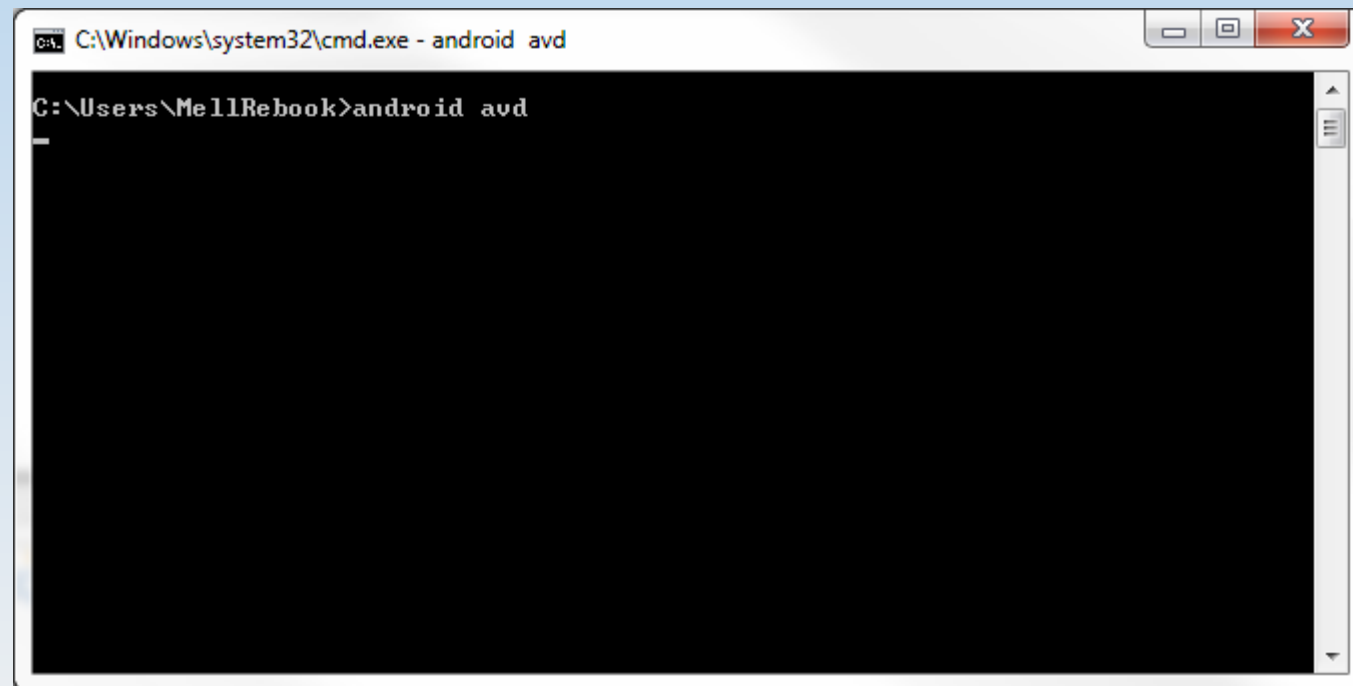
Dla szybszego działania emulatora polecam zainstalowanie paczki „Intel x86 Emulator Accelerator (HAXM)”



Konfiguracja emulatora

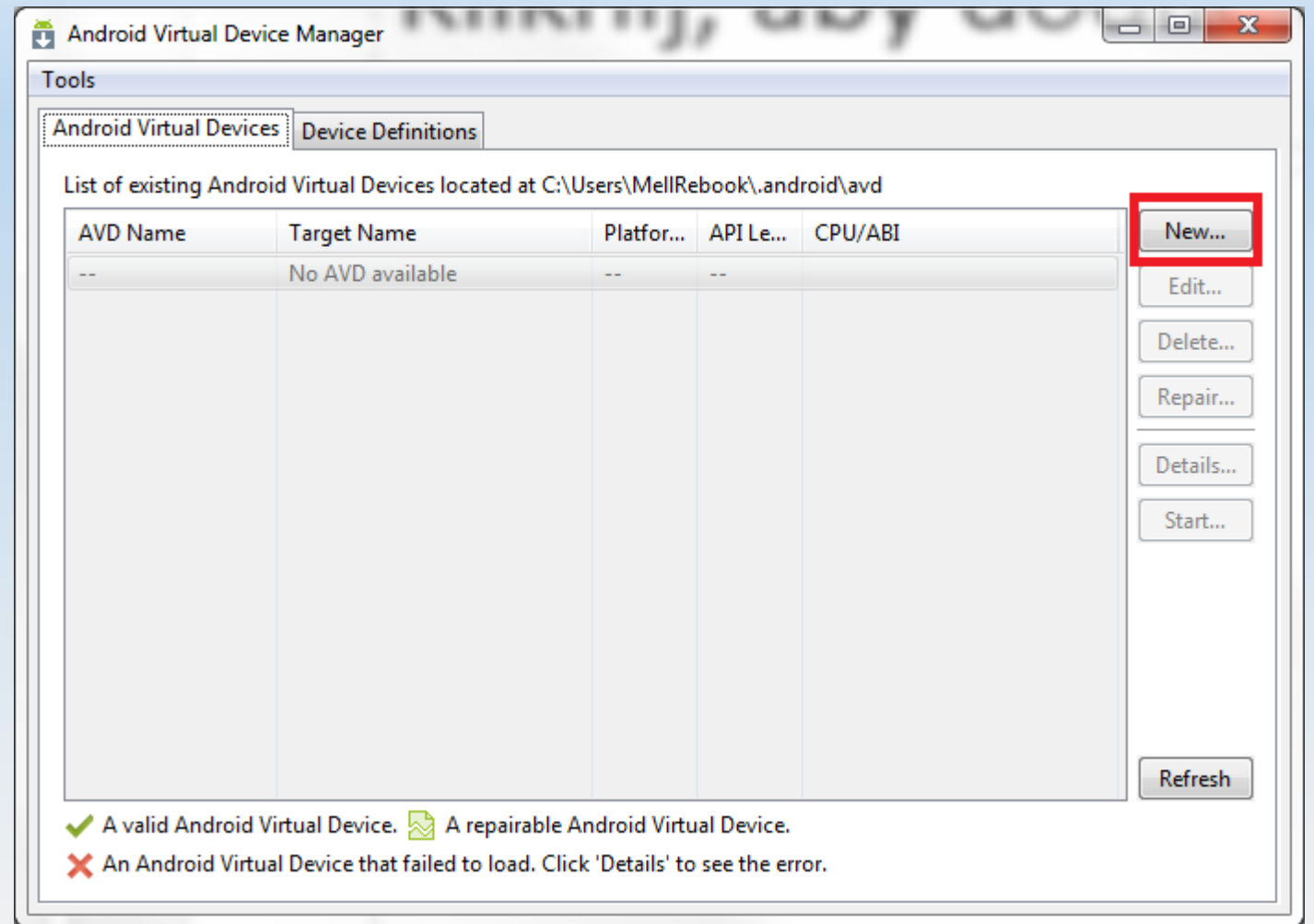


W wierszu poleceń wpisujemy „android avd”



```
C:\Windows\system32\cmd.exe - android avd  
C:\Users\MellRebook>android avd
```

Wybieramy „New...”



1. Nazwa naszej maszyny
2. Profil sprzętowy (można wybrać profil bazujący na bardziej znanych modelach, bądź uzależnić się od rozmiaru ekranu i jego rozdzielczości), polecam wybrać jakiś mniejszy rozmiar dla słabszych komputerów
3. Wersja Androida – dzisiaj będziemy pracować na najnowszej wersji 4.4.2 „KitKat”
4. Procesor na którym emulator będzie się opierać (polecam Intel Atom, po zainstalowaniu odpowiedniej paczki, jednakże ARM też powinien zrobić swoje)
5. „Przyciski ekranowe” – czyli emulowane przyciski hardwarowe dostępne w komórce (menu, cofnij itp.)
6. Skórka emulatora
7. Kamera (oprócz emulowania obrazu jest możliwość podpięcia kamery internetowej)
8. Ilość pamięci RAM
9. Pamięć wewnętrzna komórki
10. Ilość dostępnego miejsca na karcie pamięci

Create new Android Virtual Device (AVD)

AVD Name: 1 NazwaMaszyny

Device: 2 4" WVGA (Nexus S) (480 × 800: hdpi)

Target: 3 Android 4.4.2 - API Level 19

CPU/ABI: 4 Intel Atom (x86)

Keyboard: 5 ☒ Hardware keyboard present

Skin: 6 Skin with dynamic hardware controls

Front Camera: None

Back Camera: 7 None

Memory Options: 8 RAM: 512 VM Heap: 32

Internal Storage: 9 200 MiB

SD Card: 10 ☒ Size: 64 MiB ☐ File: Browse...

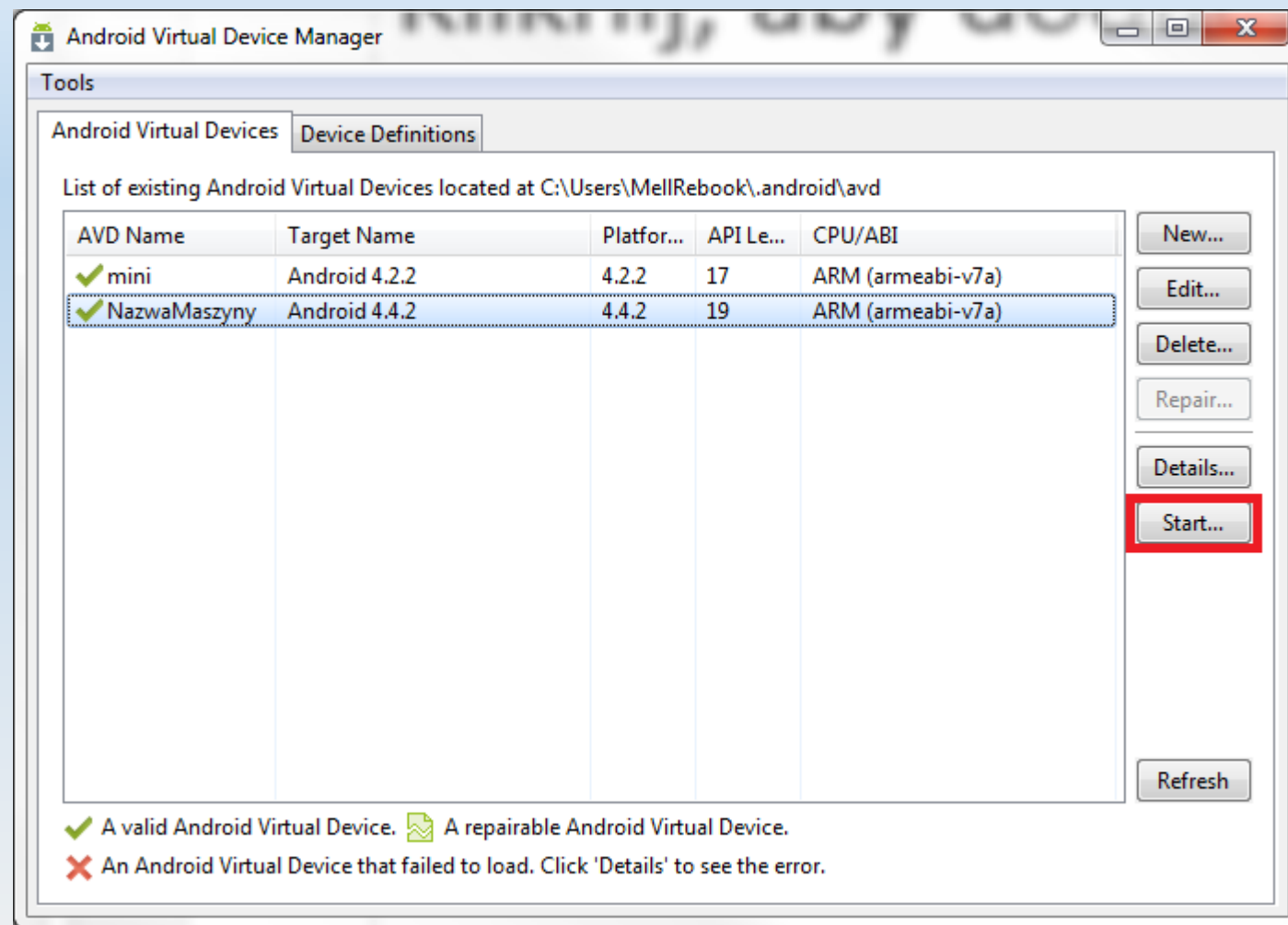
Emulation Options: ☐ Snapshot ☐ Use Host GPU

☐ Override the existing AVD with the same name

OK Cancel

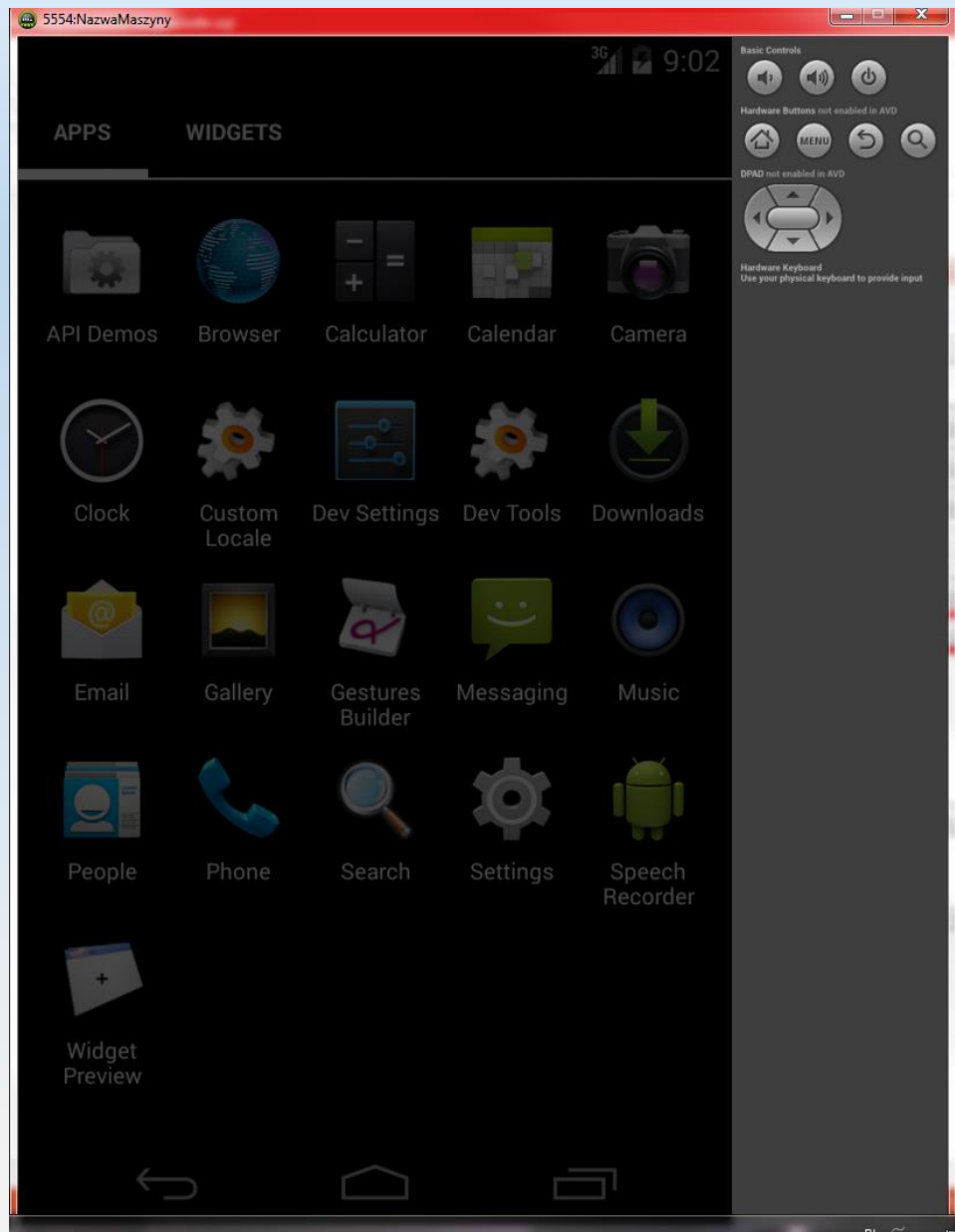
Pozostało nam tylko wybrać maszynę i ją uruchomić.

W razie problemów z uruchomieniem emulatora, możliwe że będzie potrzebne poeksperymentowanie z ustawieniami procesora, profilu czy też ilości pamięci RAM. Wszystko to zależy od mocy naszego komputera.



Gratulacje! Właśnie skonfigurowałeś emulator i go uruchomiłeś!

Emulator potrafi się włączać dość długo, więc polecam nie wyłączać go podczas pracy nad aplikacją.

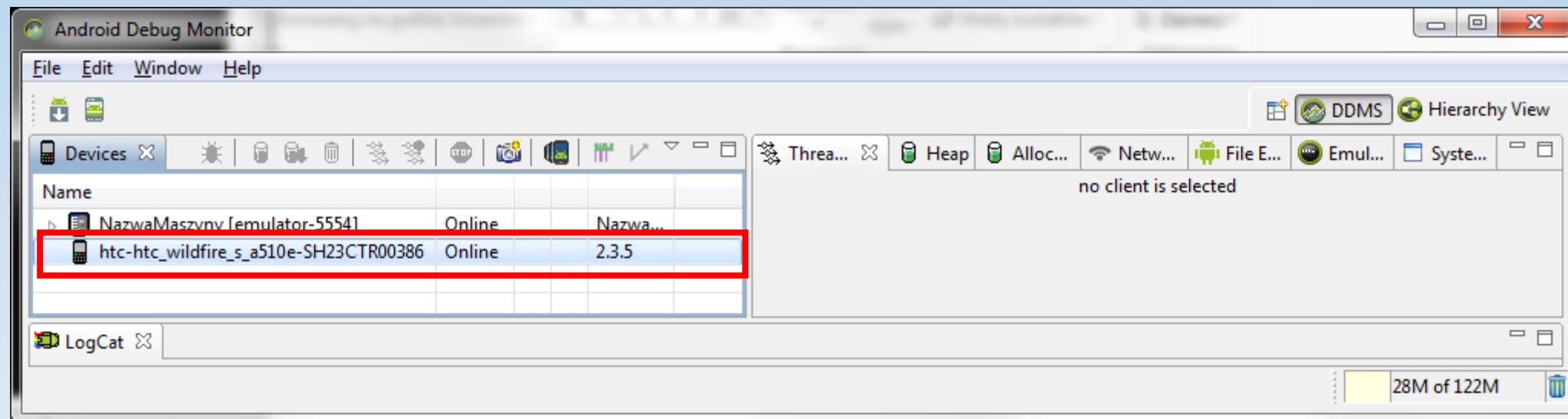


Debugowanie aplikacji na prawdziwym telefonie:

Oczywiście jest taka możliwość, trzeba jednak zrobić kilka rzeczy:

- Ustawienie opcji debugowania USB w telefonie (w różnych wersjach Androida znajduje się to w różnych miejscach np.: Ustawienia->Aplikacje->Tworzenie->Debugowanie USB)
- Podpięcie telefonu poprzez USB
- Zainstalowanie odpowiednich sterowników (jest to największy problem, przez który zdecydowaliśmy się użyć emulatorów. Praktycznie każdy model wymaga znalezienia sterowników dostarczonych przez producenta, a znalezienie ich czasem zajmuje sporo czasu i nerwów)
- Sprawdzenie czy monitor Androida widzi nasz telefon (w cmd wpisujemy „monitor”) – jeśli nie to powtarzamy punkt wyżej.
- Jeśli wszystko zrobiliśmy poprawnie, to instalowanie aplikacji na telefonie wygląda tak samo jak na emulatorze.

Tak wygląda nasz telefon w monitorze



Tworzenie przykładowego projektu w PhoneGapie



Zanim zaczniemy:

- Potrzebujemy PhoneGapa – powinien znajdować się w folderze z zasobami. Kopiujemy go gdzieś w wygodne miejsce.
- Następnie tworzymy sobie folder gdzie będziemy przechowywać nasze projekty, polecam utworzyć go gdzieś bezpośrednio na dysku C:\ tak aby łatwo można było wpisać ścieżkę w wierszu poleceń.

Otwieramy wiersz poleceń, i przechodzimy do folderu PhoneGapa. Następnie dostajemy się do folderu lib->android->bin

Można również ułatwić sobie życie klikając prawym przyciskiem myszy na folder bin w folderze android z wciśniętym klawiszem shift, a następnie wybieramy opcję „Otwórz okno poleceń tutaj”

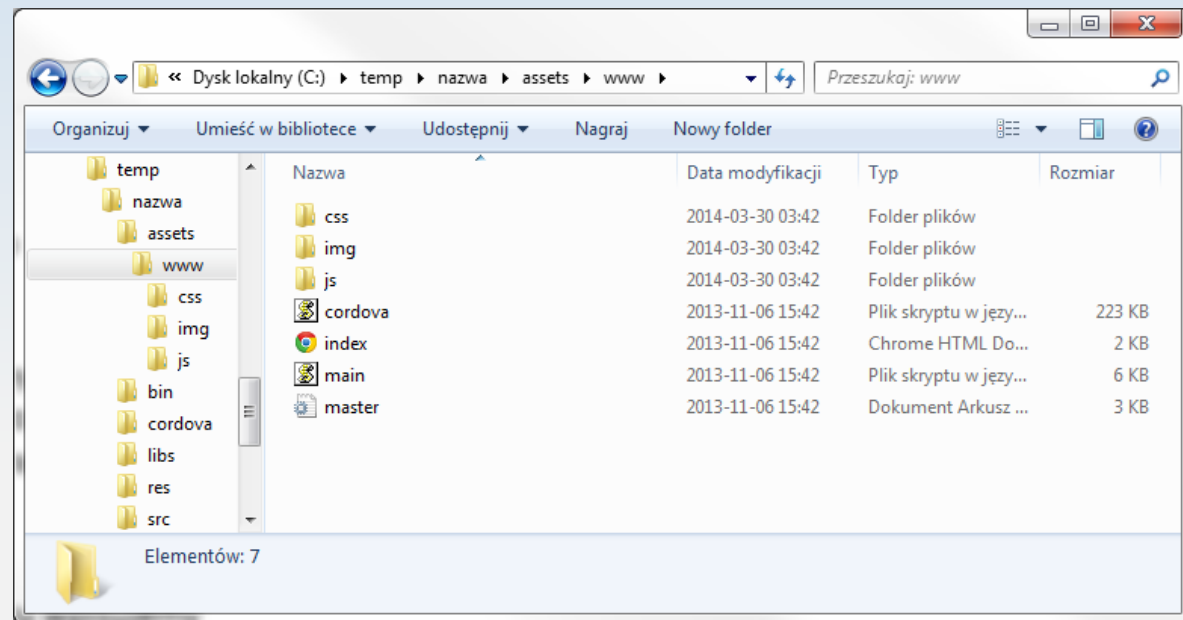
```
C:\Windows\system32\cmd.exe

C:\Users\MellRebook\Desktop\phonegap-2.9.1\lib\android\bin>create C:\temp\nazwa
com.example.nazwaProjektu nazwaProjektu
Creating new android project...
Copying template files...
Copying js, jar & config.xml files...
Copying cordova command tools...
Updating AndroidManifest.xml and Main Activity...

C:\Users\MellRebook\Desktop\phonegap-2.9.1\lib\android\bin>
```

Kiedy już znajdujemy się w odpowiednim miejscu wpisujemy „create ścieżka_do_naszego_folderu_na_projekty\nazwa_folderu com.example.nazwaProjektu nazwaProjektu”

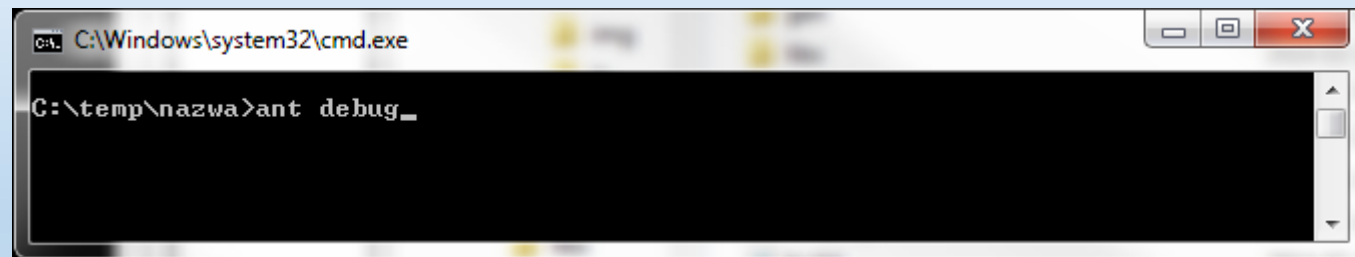
Po tym zabiegu w naszym folderze, powinny pojawić się pliki projektu.



Instalowanie aplikacji na telefonie



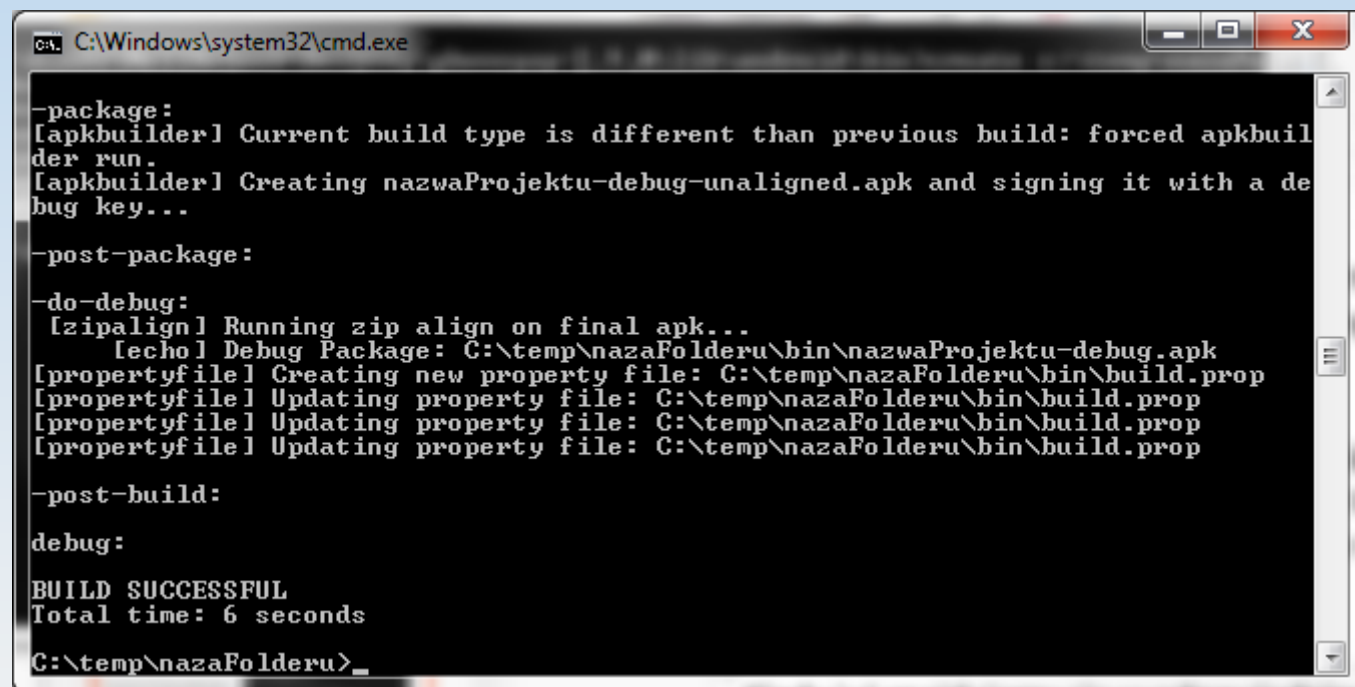
W folderze projektu otwieramy okno poleceń i wpisujemy „ant debug”



```
C:\Windows\system32\cmd.exe
C:\temp\nazwa>ant debug_
```

Jeśli zrobiliśmy wszystko poprawnie powinno nam się pojawić napis:

„BUILD SUCCESSFUL”



```
C:\Windows\system32\cmd.exe
- package:
[apkbuilder] Current build type is different than previous build: forced apkbui
der run.
[apkbuilder] Creating nazwaProjektu-debug-unaligned.apk and signing it with a de
bug key...
- post-package:
- do-debug:
[zipalign] Running zip align on final apk...
[echo] Debug Package: C:\temp\nazaFolderu\bin\nazwaProjektu-debug.apk
[propertyfile] Creating new property file: C:\temp\nazaFolderu\bin\build.prop
[propertyfile] Updating property file: C:\temp\nazaFolderu\bin\build.prop
[propertyfile] Updating property file: C:\temp\nazaFolderu\bin\build.prop
[propertyfile] Updating property file: C:\temp\nazaFolderu\bin\build.prop
- post-build:

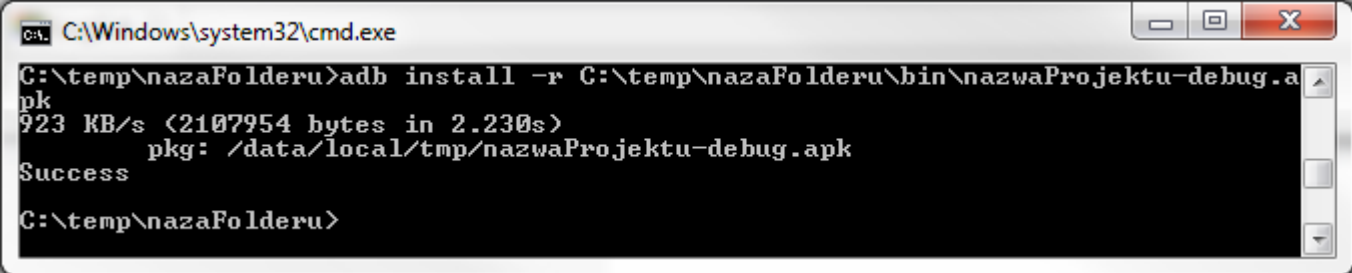
debug:
BUILD SUCCESSFUL
Total time: 6 seconds
C:\temp\nazaFolderu>_
```


Instalacja Aplikacji na telefonie/emulatorze



W wierszu poleceń wpisujemy

„adb install -r ścieżka_do_folderu_projektu\bin\nazwaProjektu-debug.apk

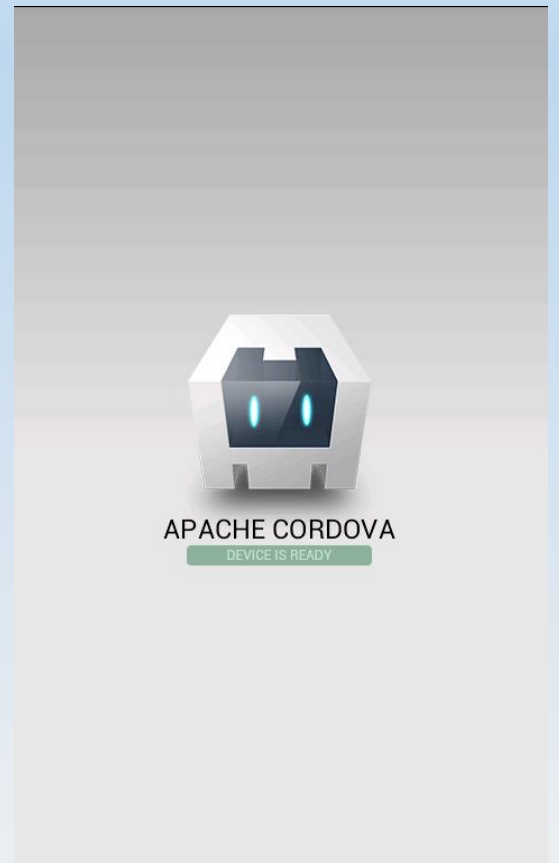
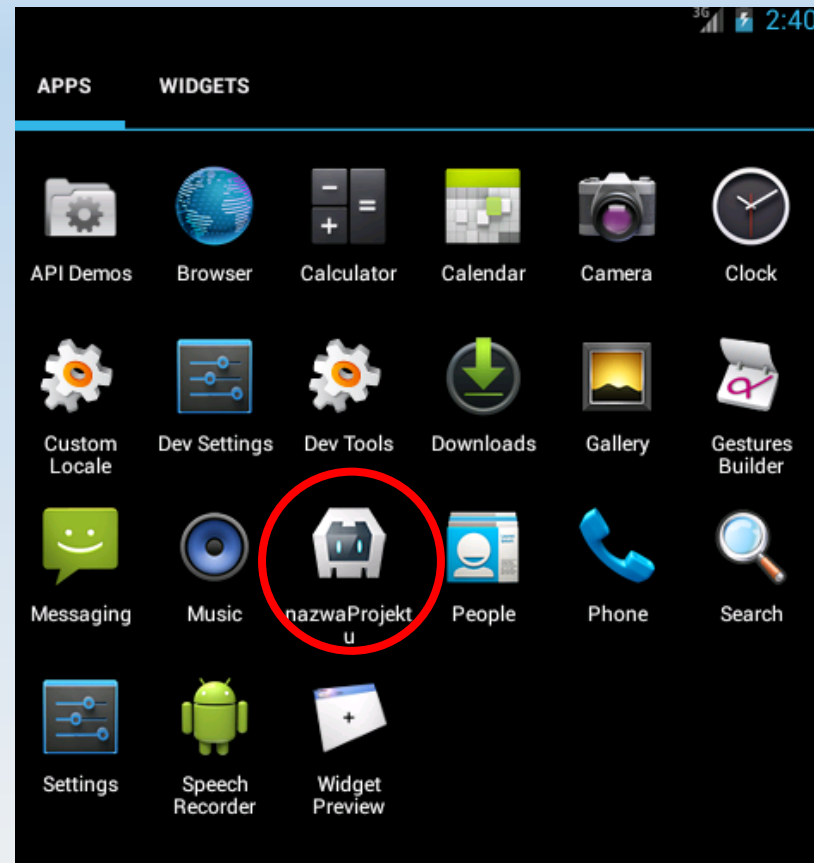


```
C:\Windows\system32\cmd.exe
C:\temp\nazaFolderu>adb install -r C:\temp\nazaFolderu\bin\nazwaProjektu-debug.apk
923 KB/s (2107954 bytes in 2.230s)
Success
C:\temp\nazaFolderu>
```

W przypadku instalacji aplikacji na prawdziwy telefon wygląda to dokładnie tak samo.

Gratulacje!

Właśnie na telefonie pojawiła się nasza aplikacja!



**Skoro już mamy skonfigurowane środowisko i emulator,
to możemy przejść do tego co tygryski lubią najbardziej.**

