# Wstęp

## Problematyka i zakres pracy

W pierwszym kwartale 2014 roku trzy na pięć sprzedaży telefonów komórkowych nosiło nazwę „inteligentny” [1]. Smartphone[[1]](#footnote-1), jak potocznie   
na urządzenia tego typu się mówi, nie tylko zawładnęły niemałym rynkiem telefonów bezprzewodowych, ale także znacznie poprawiły komfort naszego życia. Od obsługi wiadomości SMS, po zarządzanie pocztą elektroniczną,   
do sterowania temperaturą w naszych mieszkaniach – lista rzeczy, które nie potrafią te urządzenia z każdym dniem zmniejszą jeszcze bardziej.

Nic dziwnego, że tym rynkiem, który jest wart miliardy dolarów rocznie, zainteresowały się wielkie firmy. Swoje rozwiązania w tej dziedzinie technologii   
na przestrzeni ostatnich lat pokazały takie formy jak: Microsoft, Apple czy Google. W swojej pracy zapoznam się jedną z części systemu firmy z Mountain View odpowiedzialną za komunikacji miedzy komponentami zainstalowanymi   
na urządzeniu. Dzięki systemowi nadajników i odbiorników program, które będzie odbierał wiadomości SMS nie musi posiadać dodatkowego wątku odpowiedzialnego za sprawdzenie czy wiadomość tekstowa przyszła, tylko potrzebuje wyczekiwać informacji od części systemu odpowiedzialnej   
za odbieranie krótkich wiadomości. Taki sposób rozwiązania komunikacji między aplikacja w Android przysporzył się do znacznego ograniczenia używanych zasobów przez oprogramowanie działające na tym systemie.

Częściom badawczą mojej pracy było wykonanie aplikacji przedstawiającej działanie tego mechanizmu w Androidzie. Program ten wykorzystuje interfejsy programistyczne, któryezostały udostępnione przez firmę Google. Sama aplikacja jest napisana w wersji standardowej języka Java. Niemniej jednak sposób, w jaki został zaprojektowany interfejs programistyczny Androida różni się od tego zaproponowanego przez Oracle. Programiści znający już podstawową edycje Javy zaczynający swoją przygodę z systemem od firmy z Mountain View muszą zapoznać się nie tylko z nazwami udostępnionych klas i metod, ale z sposobem tworzenia interfejsu użytkownika czy komunikacji między systemem a aplikacją. Z tego powodu uważam, że moja praca nie tylko zapoznała czytelnika z systemem rozsyłania i odzierania informacji w Androidzie, ale także pokażała   
go z podstawowymi aspektami tworzenia oprogramowania na ten system.

W części teoretycznej zostaną opisanego takie zagodnienia jak:

* Jak korzystać z powyszch interfejsów
* Komponenty systemu Android, z których można pobierać lub wysyłać informacje potrzebne w działaniu programu
* Klasy poboczne (tj. **IntentFilter**), które są potrzebne w korzystaniu   
  z tych interfejsów
* Zasięg działania obiektów typu **Intent** oraz **BroadcastReceiver**

## Cele i zakres pracy

Głównym celem pracy jest przedstawienie w języku polskim mechanizmu komunikacji między kompomentami w systemie Android. Postanowiłem wykorzystać moją prace dyplomową do zaprezentowania tego zagadnienia, ponieważ podczas zaczęcia nauki programowania na ten system natrafiłem   
na dużą, dla sporej grupy kludzi, bariere – bariere językową.

Z powodu przedstawionego w poprzednim akapicie, moja praca skupi się   
na opisanie jednych z najważniejszych mechanizmów w systemie od Google, mianowicie na klasach **Intent** oraz **BroadcastReceiver**. Te obiekty ułatwiają komunikacje miedzy kompomentami aplikacji. Trudno wyobrazić sobie scenariusz gdzie, mając na uwadze jak najmniejesze zużycie procesora i pamieci operacyjnej, na jednoczestnym istnieniu dwóch programów na tym samym urządzeniu, które korzystają z osobnych wątków do wykonania **tej samej długiej operacji**.

Pomniejszym celem mojej pracy dyplomowej jest przedstawienie także takich części programu pisanego pod Adroida jak:

* Application manifest
* Activities
* Zarządzenie i rodzaje zasodów, z ktrórych korzystają aplikacje Androidowe
* Android background processing
* Android pemissions

Powyższe komponenty są częścią sporej ilości programów mobilnych. Bez znajomości tych technologi trudno, by było zacząć pisać aplikacje Androidowe jak i opisać tytułowy mechanizm.

## Przegłąd literatury

Podstawowym źródłem informacji na temat tworzenia aplikacji na Androida jest jego documentacja [2]. Wiedza ukazania na tej stronie internetowej jest przeżysta oraz napisana łatwym do zrozumienia językiem. Stronie nie tylko zawiera opis API[[2]](#footnote-2) Androida , ale także przykłady zostasowań komponentów   
w niej zawartych [3]. O ile dokumentacja dostarczona przez Google jest bardzo dobrze napisana w większości obszarów, to tak niektóre nie zostały w wogóle wspomniane. Przykładem komponentu, który nie został opisany w tutorialach[[3]](#footnote-3), jest klasa **BroadcastReceiver**. Naszczeście to zagadnie zostało opisane przez osoby spoza firmy z Mountain View [4].

O ile wiedza na temat tworzenia aplikacji działający na Androidzie dostępną w Internecie jest duża i najczęściej darmowa, ale za to jest ona rozsiana po całej sieci. Czas spędzany na przeszukiwaniu Internetu w poszukiwaniu dobrej jakości artykułu na temat nas interesujący jest czasami ogromny, dlatego istnienie książek z tej domeny nauki jest bardzo potrzebny. Jednym z najlepszych podręczników mówiących o programowaniu na Androida mim zdaniem należy wolumen autorstwa Reto Meier [5]. Książka ta to jeden z najlepszych kompendiów wiedzy zawartej jednym podręcznym opakowaniu. Zawarte w min jest wszystko, co musi wiedzieć każdy, kto chce programować na system od Google.

Z Języku Polskim

1. Smartphone (z ang. *Inteligentny telefon*) – urządzenie elektroniczne, które można zmieścić w kieszenie posiadające możliwość wykonywania połączeń głosowych oraz pozwalające na instalowanie i uruchamianie oprogramowania firm trzecich. [↑](#footnote-ref-1)
2. API (skrót od *Application Programming Interface* – z ang. *interfejs programowania aplikacji*) – podprogramy, struktury danych, klasy obiektów, które udostępnia dany program, biblioteka, system operacyjny w celu komunikacji w innym oprogramowaniem. [↑](#footnote-ref-2)
3. Tutorial (z ang. *Samouczek*) – zestaw instrukcji pozwalający na łatwe nauczenie się okreśklonego zagadnienia. [↑](#footnote-ref-3)