

# Fourier's Phone - projekt techniczny

Frederic Grabowski

Bartłomiej Karasek  
Andrzej Swatowski

Wojciech Przybyszewski

21 maja 2018  
ver. 1.1

## Spis treści

1	Opis projektu	3
2	Wykorzystanie aplikacji	3
3	Architektura aplikacji	3
4	Rozwiązania techniczne/technologie	4
5	Algorytmy	4

## 1 Opis projektu

Celem projektu jest zbudowanie aplikacji na telefony z systemem Android w wersji 5.0 lub wyższej pozwalającej na przesyłanie różnego rodzaju danych między telefonami za pomocą dźwięku.

## 2 Wykorzystanie aplikacji



Użytkownicy korzystają z aplikacji na swoich telefonach z zainstalowanym systemem Android w wersji 5.0 lub wyższej.

Zależnie od wybranego trybu (nadawania lub odbierania), aplikacja korzysta z odpowiedniej klasy, która to za pomocą zaimplementowanych na niższym poziomie abstrakcji metod albo wysyła dźwięki o określonych częstotliwościach, albo je odbiera. Aplikacja w trybie odbierania następnie przekształca odebrane dźwięki w tekst i wyświetla na ekranie telefonu.

## 3 Architektura aplikacji

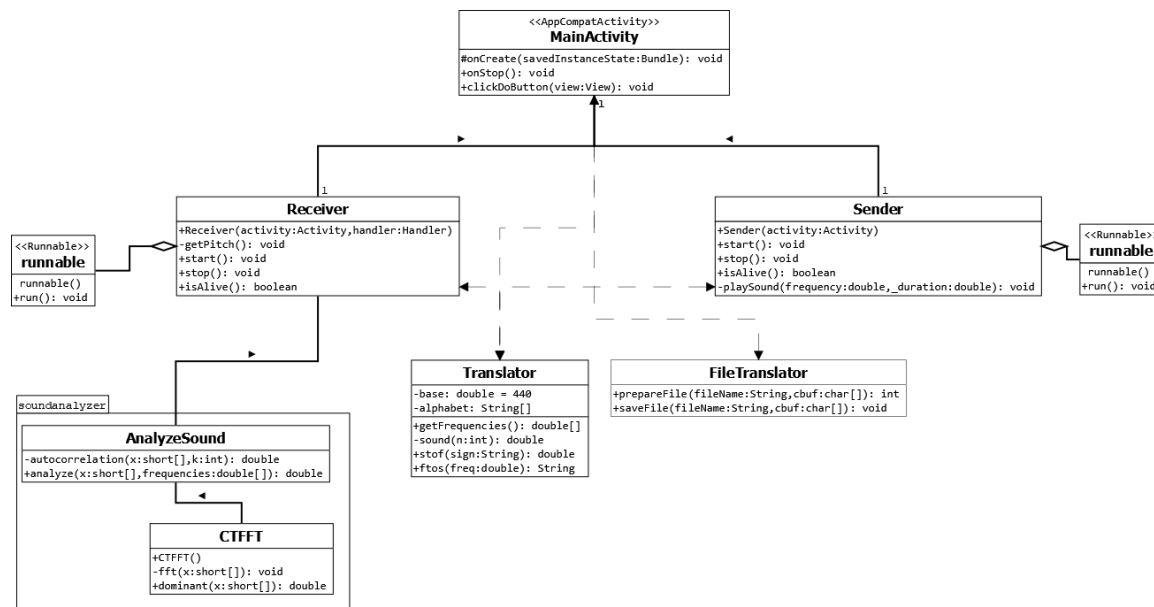
Aplikacja ma jeden ekran, implementowany przez klasę `MainActivity`, dziedziczącą po `AppCompatActivity`.

Zależnie od wybranej opcji (za pomocą stosownego przełącznika), `MainActivity` korzysta z metod zaimplementowanych w klasach `Receiver` (tryb odbierania) oraz `Sender` (tryb nadawania). Klasy te wykorzystują wewnętrzne metody Androida, by kontrolować mikrofon oraz głośniki. Do tłumaczenia tekstu na dźwięk lub odebranych dźwięków na tekst służą publiczne metody `stof` i `ftos` z klasy `Translator`.

Klasa `Receiver` korzysta z klas i metod umieszczonych w wewnętrznej paczce `soundanalyzer`. Wewnętrzna implementacja korzysta z algorytmów operujących na dźwiękach i funkcjach fal, takich jak FFT.

Aplikacja oprócz trybu wysyłania tekstu oferuje także tryb wysyłania plików. Do operacji na plikach zostanie wykorzystana osobna klasa, `FileTranslator`, która to oferować będzie publiczne metody

*prepareFile* (do wysyłania pliku) oraz *saveFile* (do tłumaczenia odebranych przez aplikację znaków na binarną reprezentację pliku).



## 4 Rozwiązania techniczne/technologie

Całość aplikacji jest napisana w Javie, korzystając z bibliotek systemowych Androida. Android musi być w wersji 5.0 lub wyższej, bowiem aplikacja korzysta z metod i funkcjonalności, które pojawiły się dopiero w tej wersji API.

## 5 Algorytmy

W trosce tak o wydajność rozwiązania, jak i poprawność odbieranych danych należy skorzystać z odpowiednich algorytmów.

Podstawowym algorytmem z jakiego aplikacja będzie korzystać (skąd wynika także jej nazwa) jest FFT - Szybka Transformata Fouriera. Ponieważ algorytm jest dobrze znany i krótki, zastosujemy własną implementację (działającą w  $O(n \log n)$ ). Zmniejsza to ilość wykorzystywanych bibliotek, a co za tym idzie także rozmiar aplikacji. Pozwala też na większą kontrolę nad szczegółami technicznymi, takimi jak format danych.

Prócz tego zakłada się wykorzystanie innych algorytmów (takich jak autokorelacja), które przyspieszą działanie aplikacji oraz zapewnią jej niezawodność lub też radykalnie obniżą prawdopodobieństwo błędu. Wybór, ewentualna implementacja i przetestowanie tych rozwiązań wchodzi w zakres obowiązków programistów.