

## RAPORT Z POSTĘPU PRAC NAD PROJEKTEM

Temat projektu: odtwarzacz muzyczny

### Opis dotychczasowej metodyki wytwarzania kodu

- Dotychczas kod był tworzony porcjami, z pomocą literatury(\*) oraz dostępnych w internecie źródeł wiedzy i przykładów, m.in. z kursu Complete Java Masterclass na platformie udemy.com.
- Do odtwarzacza stopniowo dodawałem dodatkową funkcjonalność, dbając za każdym razem, aby dokładnie sprawdzić działanie dodawanego elementu (w myśl zasady: lepiej dodać mniej funkcji, które działają, niż więcej, z których zachowania połowy nie jestem w stanie przewidzieć).

### Szacunkowy procentowy stopień ukończenia projektu

Interfejs:	40%
Funkcjonalności:	50%
Testy:	10%

### Informacje dot. architektury projektu

- Punktem kluczowym jest w zasadzie stworzenie funkcjonalnego i czytelnego interfejsu, umożliwiającego wygodne korzystanie z dostępnej funkcjonalności.
- Każda z funkcji odtwarzacza jest rozwijana w oddzielnym pliku, a klasa Interface ma możliwość korzystania ze wszystkich klas implementujących funkcjonalność.

### Informacje dot. kluczowych funkcjonalności, które już działają

- Wybór odtwarzanego pliku z dysku
- Wyświetlenie nazwy wskazanego do odtwarzania pliku
- Odtwarzanie plików mp3
- Sterowanie czasem odtwarzania utworu za pomocą przycisków Play, Pause, Stop
- Wyświetlanie informacji o utworze (artysta, tytuł, album) z tagu ID3

### Streszczenie sposobu testowania

- W testowaniu funkcjonalności pomagają testy jednostkowe JUnit, implementowane w klasie Tests z paczki (package'u) testing, pozwalają bowiem stwierdzić, czy wynik wykonania danej czynności na odpowiednim pliku da taki sam wynik, jak przewidziany *a priori* przez programistę.
- Testów tych będzie przybywać wraz z przybywaniem kolejnych funkcji oraz naturalnie wraz z uświadamianiem sobie o konieczności kompleksowego przetestowania funkcji, które już w programie są.
- Każda nowa funkcjonalność dodawana do projektu jest przeze mnie dokładnie sprawdzana poprzez testy na różnych plikach - sprawdzanie, czy program zachowa się tak, jak programista (czyli ja) to przewidział, skoro oczekiwany wynik wywołania jest znany.

### Napotkane trudności projektowe i implementacyjne

- Zasadniczym problemem przy implementacji tego projektu jest konieczność odkrywania Javy zupełnie od zera, jako że to mój pierwszy kontakt z tym językiem, a także w ogóle jeden z pierwszych, większych projektów, nad którym samodzielnie pracuję (a już na pewno pierwszy, który wymaga i systemu kontroli wersji, i narzędzia Maven, i pisania testów jednostkowych, ale konieczność korzystania z powyższych jest dla mnie sensowna i

zrozumiała, widzę bowiem że to naprawdę przydatne mechanizmy i na pewno pomagające w przyszłej pracy informatyka).

- Zasadniczym problemem jest zbudowanie interfejsu, co wynika z faktu uczenia się od zupełnych podstaw JavyFX (nie będę ukrywał, że to pierwsza biblioteka graficzna z którą pracuje i w zasadzie pierwszy projekt, którego szata graficzna jest tak ważna).

#### Plan dalszej pracy nad projektem

- Projekt zamierzam nadal rozwijać według zamierzonego planu, tj. stopniowo dodawać do niego kolejną funkcjonalność.
- Planuję zacząć od dokończenia obsługi ID3 tagów poprzez dodanie obsługi okładek albumów.
- Następnie postaram się dodać pasek postępu czasu oraz suwak do regulacji głośności.
- Wiem też, że z już wspomnianego powodu poznawania dopiero JavyFX równolegle czeka mnie jeszcze wiele pracy nad interfejsem aplikacji, który docelowo ma prezentować się znacznie korzystniej niż obecna wersja.

Na koniec przypomnę link do repozytorium:

<https://github.com/Slighten22/Music-Player/>

(\*) Cay S. Horstmann, Gary Cornell "Java. Podstawy", wyd. Helion;  
Rogers Cadenhead, "Java w 21 dni", wyd. Helion