

Kielce 30.06.2022

Politechnika Świętokrzyska w Kielcach

Wydział Elektrotechniki, Automatyki i Informatyki

# **Programowanie w języku Java**

## **Projekt “Odtwarzacz muzyki”**

Przemysław Cedro  
Rok III, grupa dziekańska 3ID14A

# 1. Opis ogólny

Tematem projektu było stworzenie prostego odtwarzacza muzyki mogącego wczytać pliki audio z systemu.

Projekt został napisany w języku Java w wersji 17 przy pomocy platformy do tworzenia interfejsów użytkownika JavaFX. Wykorzystana została również biblioteka org.json do przetwarzania plików JSON. Do testowania wykorzystany został framework JUnit 5. Jako narzędzie do budowania i zarządzania zależnościami w projekcie wybrane zostało narzędzie Maven.

Projekt tworzony i testowany był na platformie Linux w dystrybucji Ubuntu.

Można go znaleźć w zdalnych repozytoriach git pod adresami:

[https://orkan.tu.kielce.pl/gitlab/cedro\\_p/MusicPlayer](https://orkan.tu.kielce.pl/gitlab/cedro_p/MusicPlayer)

<https://github.com/SpontanCombust/MusicPlayer>

## 2. Funkcjonalności programu

Program może przeskanować wskazany dla niego folder na dysku i znaleźć w nim pliki audio. Odnaleziona muzyka wypisana jest w panelu “Biblioteka” w zakładce “Wszystkie utwory” oraz w zakładce “Albumy”. Każdy album odpowiada folderowi, w którym odnaleziono zostały pliki audio. Utwory przy pomocy menu kontekstowego można dodawać do playlisty (kolejki) jak i organizować je we własnych kolekcjach utworów. Albumom oraz własnym kolekcjom można zmieniać nazwę, okładkę, dodawać przypisane do nich utwory do kolejki. Je same można też usuwać jeśli zachodzi taka potrzeba. Dane o albumach i własnych kolekcjach można zapisać do pliku konfiguracyjnego, z którego przy następnym uruchomieniu programu można na powrót wczytać stan biblioteki.

Panel “Odtwarzacza” zawiera playlistę z wszystkimi dodanymi wcześniej do niej utworami oraz interfejs z elementami dobrze znanymi z innych odtwarzaczy typu przycisk odtwarzania i pauzy, przechodzenia do poprzedniego i następnego utworu, oś czasu utworu i suwak ustawiania głośności.

### 3. Uruchomienie i obsługa

Do uruchomienia potrzebne jest Java SE 17 oraz oprogramowanie Maven. W głównym folderze projektu należy wykonać komendę

```
mvn javafx:run -f ./pom.xml
```

Zaraz po uruchomieniu zostanie ukazana zakładka “Odtwarzacza”. Kolejka utworów będzie początkowo pusta. Aby dodać do niej utwory należy najpierw dać znać programowi o tym gdzie są one dostępne na dysku. Aby to zrobić w górnym menu “Plik” należy wybrać opcję “Wczytaj muzykę z folderu” i wskazać folder z muzyką.

Aby zobaczyć wszystkie wczytane utwory należy skierować się do panelu “Biblioteka”. Tam można zobaczyć wszystkie dostępne utwory w zakładce “Wszystkie utwory” lub pogrupowane według folderów, w których się znajdowały w zakładce “Albumy”. Zakładka “Własne kolekcje” zawiera grupy utworów dodanych własnoręcznie. Aby zobaczyć utwory znajdujące się wewnątrz albumu lub własnej kolekcji należy kliknąć podwójnie na dany kafelek. Pojawi się wtedy lista z nazwami utworów i przycisk powrotu. Pojedyncze utwory przy pomocy menu kontekstowego można dodawać i usuwać z kolejki i własnych kolekcji. Dla całych albumów i własnych kolekcji natomiast menu kontekstowe pozwala na zmianę nazwy, obrazu używanego jako okładka i usunięcie tej kolekcji.

Po dodaniu utworów do kolejki pojawią się one na liście w panelu “Odtwarzacz”. Od tego momentu można wybrać dowolny wpis w tej liście i zacząć odtwarzać muzykę. Interfejs odtwarzacza będzie dostarczał informacji o nazwie utworu, okładce albumu do którego należy, czas jaki upłynął od rozpoczęcia utworu i głośności muzyki. Zawiera również przyciski wyboru kolejnego i następnego utworu, odtwarzania i pauzy oraz przełączniki do automatycznego odtwarzania kolejnych utworów i zapętlania.

## 4. Specyfikacja kodu

Klasy reprezentujące dane o przechowywanej muzyce to kolejno `MusicTrack`, `MusicCollection` i `MusicAlbum`.

`MusicTrack` zawiera metodę statyczną `fromFile`, która bierze ścieżkę do pliku audio jako parametr i zwraca obiekt tej klasy jeśli plik jest poprawny lub null jeśli tak nie jest. Metody dostępne tej klasy to:

- `getFilePath` - zwraca pełną ścieżkę do pliku audio,
- `getName` - zwraca nazwę pliku,
- `getParentAlbumPath` - zwraca pełną ścieżkę do folderu, w którym utwor się znajduje,
- `getParentAlbum` - zwraca obiekt albumu, w którym znajduje się utwór zglądając do bazy albumów `MusicDatabase`.

`MusicCollection` jest klasą bazową dla `MusicAlbum`. Sama w sobie wykorzystywana jest do reprezentacji własnych kolekcji użytkownika. Posiada informacje o:

- nazwie kolekcji - dostęp przy pomocy `setName` i `getName`
- obrazie używanym jako okładka kolekcji - dostęp przy pomocy `setCoverImagePath`, `getCoverImagePath` i `getCoverImage`
- utworach, do których odwołuje się kolekcja - dostęp przy pomocy `addTracks`, `removeTracks`, `getTracksPaths` i `getTracks`.

Klasa posiada również metody `toJSON` i `fromJSON` przetwarzające obiekt klasy z lub w obiekt typu `JSONObject` biblioteki `org.json`.

`MusicAlbum` rozszerza klasę `MusicCollection` i dodatkowo zawiera informacje o ścieżce do folderu, do którego album się odwołuje. Do utworzenia obiektu używane są metody statyczne `fromDirectory` i `fromDirectoryRecurse`. Dodatkowo nadpisuje metody `toJSON` i `fromJSON`.

Aby mieć dostęp do obiektów powyższych klas stworzona została klasa `MusicDatabase`, która w czasie trwania programu zawiera informacje o wszystkich utworach, które użytkownik będzie chciał. Do ładowania i zapisywania na później biblioteki utworów służą jej `saveToConfigurationFile`, `loadFromConfigurationFile` i `loadFromFileSystem` oraz metody pomocnicze do nich z przedrostkiem `request`, które zamiast przyjmować ścieżki do plików jako argument pokazują okno dialogowe, w którym użytkownik sam wybierze tę ścieżkę. Klasa oczywiście posiada metody dostępne typu `clearAlbums`, `addAlbums` i `getAlbumMap`.

Klasa `Jukebox` jest klasą typu singleton. Odpowiedzialna jest za przechowywanie obecnej kolejki utworów i ich odtwarzanie. Zawiera pola typu `Property`, by elementy interfejsu mogły reagować lub wpływać na stan klasy nasłuchując zmiany ich stanu. Zawiera również instancję klasy `MusicDatabase`.

Klasa `TrackListView` jest już klasą wykorzystywaną bezpośrednio w konstruowaniu UI. Rozszerza klasę `ListView` z biblioteki `JavaFX`. Każda linijka listy odpowiada utworowi, do której ta lista się odwołuje. Rozszerzają ją klasy:

- `PlaylistTrackListView` - lista używana jako kolejka w panelu Odtwarzacza,
- `AllTracksTrackListView` - lista używana w panelu Biblioteka w zakładce Wszystkie utwory,
- `CollectionTrackListView` - lista dla kafelek albumów i kolekcji użytkownika.

Każda z nich odpowiada szczególnemu przypadkowi użycia elementu interfejsu. Dla `TrackListView` istnieje klasa `TrackListContextMenu`, która określa jak wygląda i działa menu kontekstowe dla elementów listy.

Kolejną szczególną klasą UI jest `LibraryCollectionTile`. Odpowiada ona kafelkowi wyświetlanemu w zakładkach "Albumy" i "Własne kolekcje" w panelu biblioteki. Reprezentuje wizualnie obiekty typu `MusicCollection`. Menu kontekstowe dla tej klasy określa klasa `LibraryCollectionTileContextMenu`. Po naciśnięciu dwukrotnie na kafelek pojawia się lista `TrackListView` z utworami, do jakiej odwołuje się kolekcja, którą reprezentuje kafelek.

Innymi kontrolerami układu interfejsu są klasy:

- `LibraryModelController` - klasa kontrolująca działanie elementów panelu biblioteki
- `PlayerHud` - kontroluje działaniem elementów interfejsu odtwarzacza w panelu "Odtwarzacz"
- `PlayerHudAndTrackList` - łączy elementy `PlayerHud` i `PlaylistTrackList`
- `RootController` - kontroluje działaniem "korzenia" interfejsu, czyli przyciskami do przełączania między odtwarzaczem a biblioteką oraz za działanie menu u góry ekranu.

Pozostałymi pomniejszych klasami są:

- `Localization` - posiada statyczną metodę `getString` zwracającą tekst odpowiadający językowi systemu
- `MusicPlayerApplication` - klasa z metodą startową dla aplikacji

## **5. Rozkład pracy**

Przemysław Cedro - cały projekt