

Projekt
Programowanie w języku JAVA

Wydział Elektrotechniki Automatyki i Informatyki
Politechnika Świętokrzyska

Studia: Stacjonarne I stopnia	Kierunek: Informatyka	Rok: 2016/2017
Data oddania: 28.06.2017	Semestr: IV	Grupa: 2ID12A
Temat projektu: Odtwarzacz MP3, OGG (temat 5)	<ol style="list-style-type: none">1. Kamil Domagała2. Damian Gawęda	

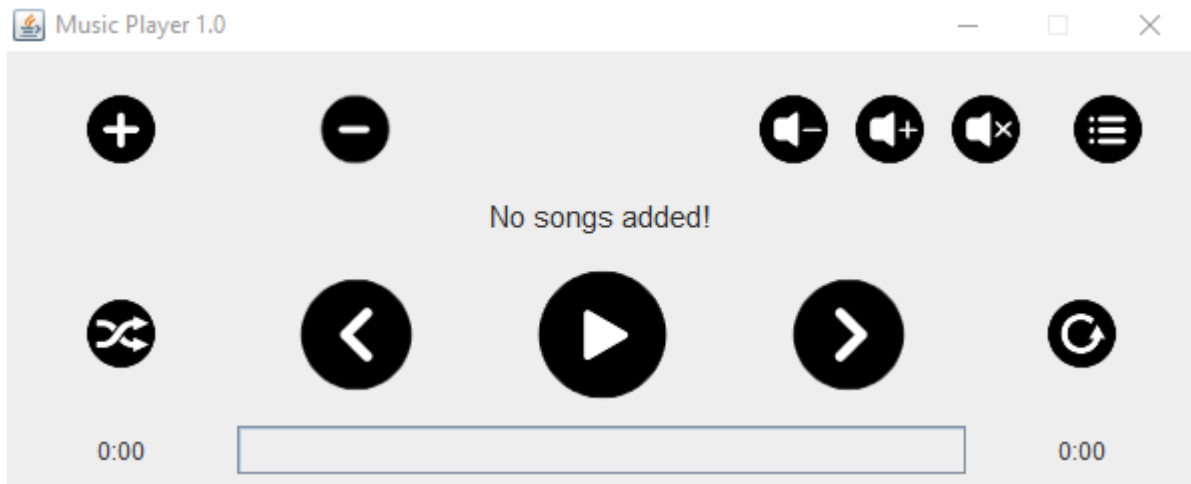
Wstęp

Tematem projektu było stworzenie aplikacji w języku JAVA która ma odtwarzać pliki .mp3 i .ogg. W tym celu skorzystaliśmy z środowiska IntelliJ IDEA.



Całość projektu podzieliśmy na kilka plików źródłowych, które zawierają klasy i metody niezbędne do prawidłowego funkcjonowania aplikacji. W pliku nagłówkowym „InterfaceMain.java” znajduje się implementacja interfejsu, który możemy zobaczyć po uruchomieniu aplikacji. Plik „FileManagement.java” zawiera klasę dzięki której możemy wybrać ścieżkę do katalogu z muzyką, którą chcemy dodać do naszej listy odtwarzania – którą obsługuje plik „Playlist.java”. Po wybraniu katalogu klasa MP3Parser wyciąga z tagów informacje o utworach i wypełnia nimi okno interfejsu. Plik „PlayerMain.java” zajmuje się kontrolą nad wątkiem ‘playera’ znajdującym się w pliku „PlayerThread.java”.

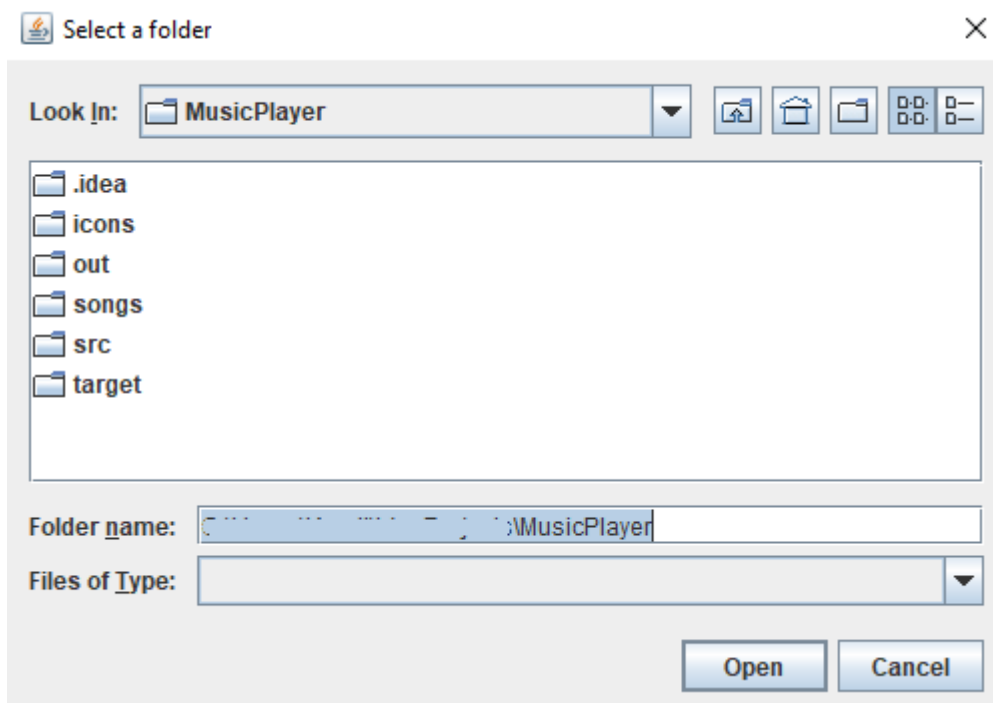
Proces uruchamiania oraz sterowanie

Aby uruchomić program należy skorzystać z pliku `MusicPlayer.jar`. Aplikacja uruchomi się i pokaże się okno główne aplikacji.



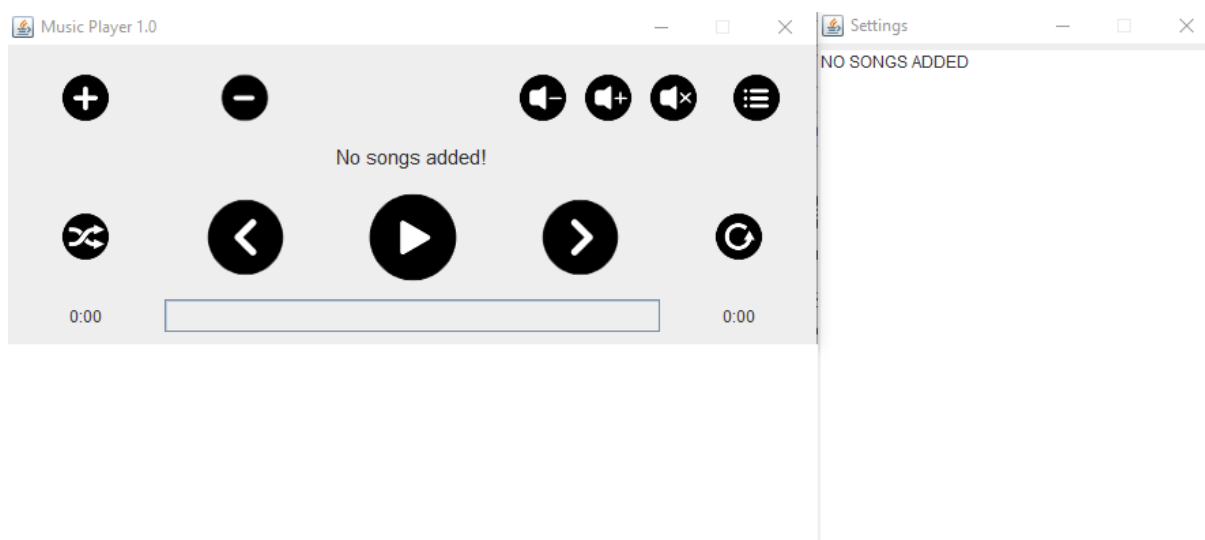
Główne okno aplikacji

Użytkownik po pierwszym uruchomieniu powinien dodać folder z którego aplikacja ma pobrać pliki .mp3 lub .ogg przyciskiem . Przy kolejnych uruchomieniach nie trzeba tego robić, ponieważ aplikacja zapamiętuje wybraną ścieżkę. Jeśli chcemy ją zresetować należy wcisnąć przycisk .



Okno wyboru ścieżki do utworów

Po wybraniu ścieżki możemy zobaczyć dodane utwory klikając przycisk . Ukaże się nowe okno w którym będą wypisane pliki które udało się odczytać z wybranego folderu.



Okno aplikacji wraz z 'Playlistą'

Aby uruchomić odtwarzanie należy wcisnąć lub aby je zatrzymać. Jeśli chcemy zmienić aktualnie grany utwór musimy wcisnąć przyciski znajdujące się obok przycisku 'play' - i .

Poziom głośności możemy modyfikować poprzez kliknięcie na ikonę



- aby zwiększyć głośność




- aby zmniejszyć głośność




- aby wyciszyć dźwięk ,



- aby włączyć go ponownie

Ponadto jeśli chcemy aby kolejność utworów była losowa musimy wcisnąć przycisk  ,

natomiast aby zapętlić utwór należy wcisnąć  .

Aplikację wyłączamy jak każdy inny program – np. na Windowsie przyciskiem „X” znajdującym się w prawym górnym rogu okna.

Opis klas i metod

- **App** – zawiera main'a, wywołujemy w niej interfejs.
- **InterfaceMain** – Główna klasa, w niej zaimplementowany jest cały interfejs wizualny i jego obsługa.
 - Jej metody ustawiają 'składniki' (tekst, przyciski) aplikacji na swoich miejscach i obsługują je
- **FileManagement** – Pozwala na dodanie katalogu z utworami.
 - `setListOfFiles()` – pobiera listę plików z folderu
 - `wasItAddedBefore(String path)` – Sprawdza czy ścieżka podana jako argument jest już dodana
 - `isEmpty()` – sprawdza czy plik jest pusty
 - `saveFiles()` – zapisuje pliki do playlisty
 - `loadFiles()` – odczytuje playlistę
 - `sortList(boolean ascending)` – sortuje playlistę wg. parametru
 - `shuffleList()` – miesza utwory na playliście
 - `getSongList()` – zwraca listę utworów
 - `moveToNextSong()` – przełącza na następny utwór
 - `moveToPrevSong()` – przełącza na poprzedni utwór
 - `getCurrentSongPath()` – zwraca ścieżkę aktualnie granego utworu
 - `getSongName()` – zwraca tytuł aktualnie granego utworu
 - `getSongDuration()` – zwraca czas trwania utworu
 - `timeToInt()` – konwertuje czas utworu (String) na int
 - `wipePlaylist()` – czyści playlistę
- **MP3Parser** – Odczytuje tagi z pliku.
 - `getName(String fl)` – odczytuje tytuł utworu o ścieżce podanej przez parametr
 - `getDuration(String fl)` – odczytuje długość utworu o ścieżce podanej przez parametr
- **Playlist** – Lista utworów
 - `setFile(Iterator<String> ite)` – ustawia listę utworów
 - `convertToText()` – zapisuje zawartość iteratora do jednego String'a
 - `createWindow()` – tworzy okno playlisty
 - `closeWindow()` – zamyka okno playlisty
- **PlayerMain** – Kontroluje wątek odtwarzacza poprzez ustawianie odpowiednich flag które są odczytywane w **PlayerThread**.
 - `startSong(FileManagement files)` – Tworzy nowy wątek jeśli nie ma innego działającego i odtwarza aktualny utwór
 - `Autoplay(...)` – Automatycznie odtwarza następny utwór po zakończeniu aktualnego
 - `stopCurrent()` – Zatrzymuje aktualnie odtwarzany utwór
 - `moveToNext(...)` – Przechodzi do odtwarzania następnego utworu
 - `moveToPrev(...)` – Przechodzi do odtwarzania poprzedniego utworu

- `increaseVolume()` – zwiększa głośność
- `decreaseVolume()` – zmniejsza głośność
- `mute()` – wycisza dźwięk
- `unmute()` – wyłącza wyciszenie

- `PlayerThread` – Wątek wykonawczy, wykonuje operacje które dał mu `PlayerMain`.