

Dokumentacja PWIR

Odtwarzacz MP3

Maciej Żądło, Filip Szoldra

Wydział Elektrotechniki Automatyki Informatyki i Inżynierii Biomedycznej

kierunek Informatyka 3 rok,

grupa 2b, wtorek 14:30-16:00

1. **Tytuł** - "Odtwarzacz MP3"

2. **Dane studentów**

- a. Maciej Żądło, 3 rok Informatyka, 2a (na zajęcia uczęszczam z grupą 2b)
- b. Filip Szoldra, 3 rok Informatyka, 2b

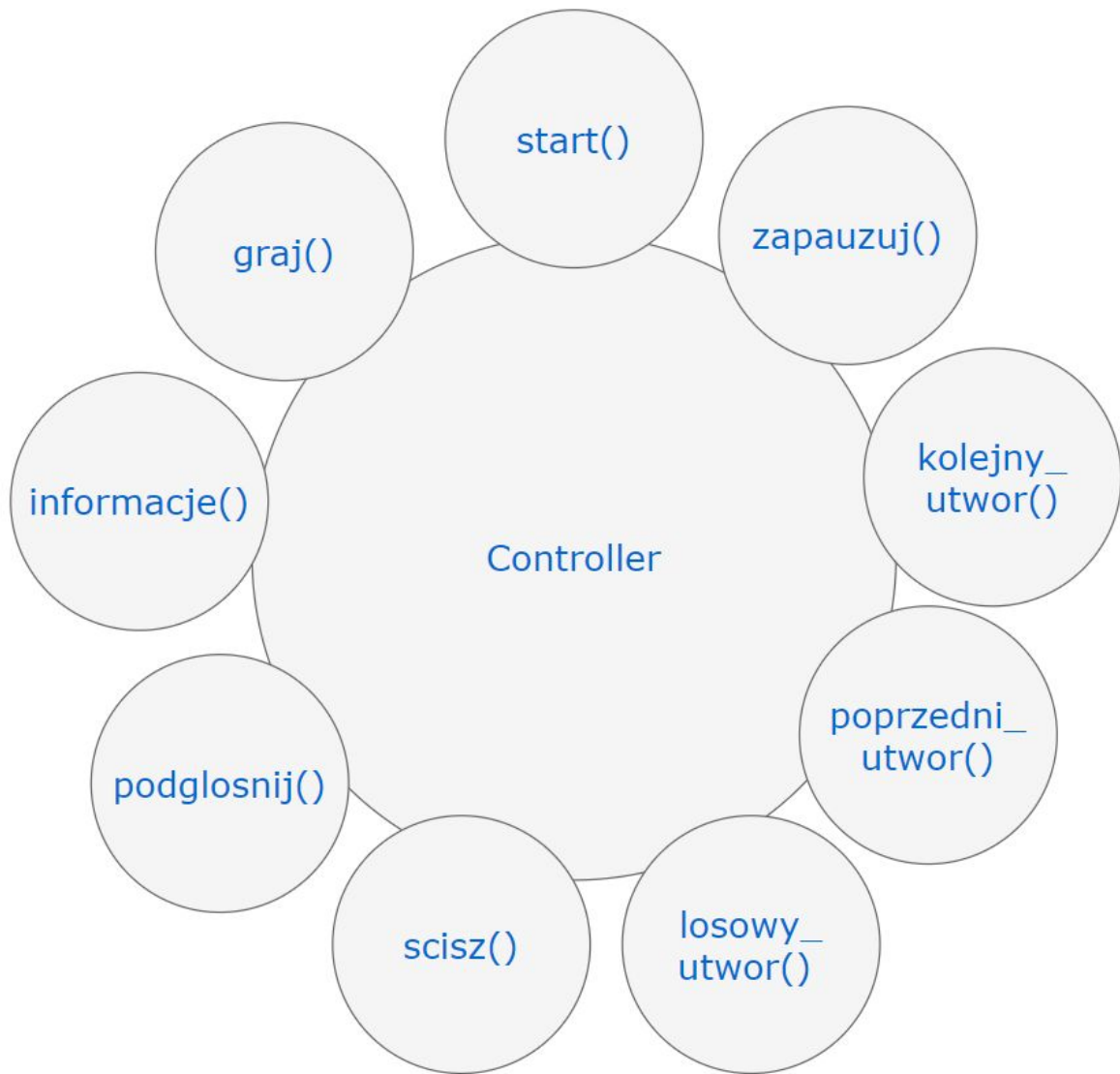
3. **Data oddania programu:** 18.01.2021 rok,

4. **Cel programu** - projekt ma na celu stworzenie działającej symulacji odtwarzacza MP3, który do komunikacji i przekazywania poleceń między kontrolerem, a komponentami urządzenia wykorzystuje w swoim działaniu współbieżność języka Erlang

5. **Opis i schemat struktury zadaniowej programu (diagram kooperacji UML, lub podobny)**

W naszym programie rozróżniamy kilka funkcji służących do obsługi naszego playera MP3. Skupiliśmy się na tym, aby były one deskryptywne i przekazywały informacje o tym czego dotyczą bez potrzeby komentowania kodu. Oto one:

- controller(Parameters) - służy do obsługi programu
- graj() - odtwarzanie muzyki
- zapauzuj() - zastopuj odtwarzanie muzyki
- kolejny_utwor() - odtwarzanie kolejnego utworu z listy
- poprzedni_utwor() - odtwarzanie poprzedniego utworu z listy
- losowy_utwor() - odtwarzanie losowego utworu z listy
- podglosnij() - zwiększenie głośności odtwarzania
- scisz() - zmniejszenie głośności odtwarzania
- informacje() - wyświetlanie informacji o aktualnej głośności oraz numerze aktualnie odtwarzanego utworu
- start() - rozpoczęcie symulacji zachowania odtwarzacza wraz z zachowaniem użytkownika



6. Informacje o stosowanych pakietach zewnętrznych

(niestandardowych) - w naszym projekcie nie skorzystaliśmy z pakietów zewnętrznych.

7. Informacje o zastosowaniu specyficznych metod rozwiązania

problemu - w naszym wypadku specyfiką rozwiązania jest posiadanie funkcji controller - jej zadaniem jest zarządzanie innymi procesami. Znaczy to mniej więcej tyle, że obsługuje ona każde zachowanie użytkownika - podgłaśnianie, ściszenie, zmianę utworu itd. Odtwarzacz ten przedstawia pewnego rodzaju symulację - tj. jako, że testowanie tego poprzez pojedyncze wywołania funkcji byłoby pracochłonne oraz ciężkie do zwalidowania, to nasz program dokonuje operacje z opóźnieniem równym 0.75 sekundy.

8. Krótka instrukcja obsługi,

W celu skorzystania z naszego odtwarzacza najpierw cały plik należy skompilować. W konsoli należy wpisać `erl` - odpali to emulator erlanga. Następnie poprzez polecenie `c(<<NAZWA_PLIKU>>)` dokonujemy kompilacji programu. Ostatecznie musimy odpalić naszą symulację poprzez wpisanie w konsoli `mp3player:start()`. Od teraz na naszym ekranie będą wyświetlać się informacje w formacie `~~ czynność ~~`, oraz `#### Wątek ####`.

9. Testy, przykłady.

Jednym z przykładów działania naszego programu będzie pokazanie tego jak działają niektóre z naszych funkcji. Stąd dla przykładu:

- a. walidacja wyświetlanych aktualnych informacji o głośności oraz aktualnie granym utworze (funkcja `informacje()`):

```
3> ~~ Zagraj losowy utwor: 25 ~~  
3> ~~ Aktualna glosnosc = 30 ~~  
3> -----  
Informacje:  
Piosenka: 25  
Glosnosc: 30  
-----
```

Jak widać na powyższym zdjęciu informacje wyświetlane są poprawnie.

- b. test zmiany utworu na poprzedni (funkcja `poprzedni_utwor()`) -

```
~~ Odtwarzanie poprzedniego utworu: 24 ~~  
~~ Odtwarzanie poprzedniego utworu: 23 ~~
```

- c. sprawdzenie czy można zmniejszyć głośność aktualnie słuchanego utworu (funkcja `scisz()`) -

```
~~ Aktualna glosnosc = 40 ~~  
~~ Aktualna glosnosc = 30 ~~
```

- d. sprawdzenie czy można zwiększyć głośność aktualnie słuchanego utworu (funkcja `podglosnij()`)

```
7> ~~ Aktualna wartosc glosnosci = 50 ~~  
7> ~~ Aktualna wartosc glosnosci = 60 ~~  
7> ~~~~~
```

- e. Pauza oraz rozpoczęcie odtwarzania utworu (funkcje `graj()` oraz `zapauzuj()`)

```
7> ~~ Muzyka zapauzowana ~~  
7> -----  
Informacje:  
Piosenka: 2  
Glosnosc: 40  
-----  
7> ~~ Aktualna wartosc glosnosci = 50 ~~  
7> ~~ Muzyka zaczela grac ~~
```

f. Kończenie działania wątków:

```
/> ##### Wylaczenie odtwarzacza MP3 #####  
7> ##### Watek odtwarzacza zakonczony #####  
7> ##### Watek pauzy muzyki zakonczony #####  
7> ##### Zagraj kolejny utwor: watek zakonczony #####  
7> ##### Zagraj poprzedni utwor: watek zakonczony #####  
7> ##### Zagraj losowy utwor: watek zakonczony #####
```

g. rozpoczęcie działania programu (funkcja start())

```
6> mp3player:start().  
Rozpoczecie symulacji MP3  
~~ Aktualna glosnosc = 40 ~~
```

10. Możliwe rozszerzenia programu:

W naszym programie możliwe jest zastosowanie wielu rozszerzeń funkcjonalnych - głównie skupiających się na poprawie jakości odczuwania osoby użytkującej program.

Do takich możliwości rozszerzenia programu należą między innymi:

- dodanie ciekawostek dotyczących najbardziej znanych utworów - wyświetlałyby one się w trakcie słuchania muzyki;
- opracowanie systemu podpowiadającego o podobnych utworach do tego który jest aktualnie odsłuchiwany
- dodanie okresowych statystyk mówiących o ilości godzin spędzonych na słuchaniu muzyki w ciągu dnia/tygodnia/miesiąca/roku.
- oraz wiele wiele innych.

11. Ograniczenia programu np. maksymalna liczba obsługiwanych plików, max. czas symulacji itp.,

Projekt odtwarzacza MP3 charakteryzuje się tym, że albo działa albo nie. Stąd nie ma specjalnego problemu z ograniczeniami dotyczącymi problemu. Za takie ograniczenie można jedynie uznać brak jakichś funkcjonalności, jednak w naszym projekcie one nie występują.