

Projekt z Programowania obiektowego:

Specyfikacja wstępna

Temat projektu projektu: aplikacja webowa jako generator fali akustycznej

Wykonująca: Anna Maria Dzieżyk, nr albumu: 318 506

Opiekun: dr inż. Marek Niewiński

Spis treści:

1. *Przeznaczenie projektu, sposób działania, jego efekty, potrzebne pliki*
2. *Diagram klas*
3. *Opis przeznaczenia poszczególnych klas*

Adnotacja 1. Przeznaczenie projektu, sposób działania, jego efekty, potrzebne pliki

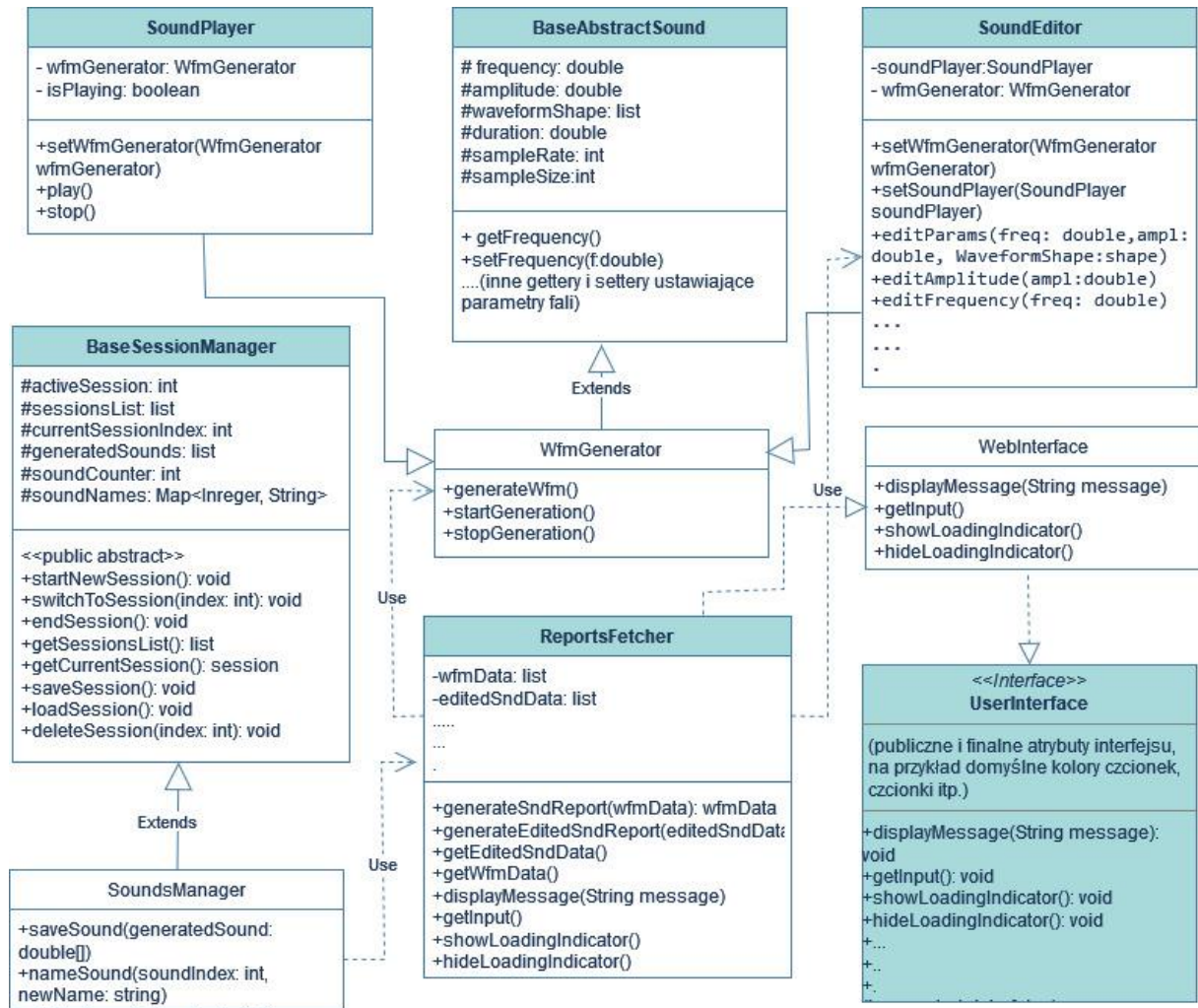
Projekt ma na celu wytworzenie fali akustycznej, o takich parametrach, jakie ustawia użytkownik. Przykładowi użytkownicy takiej aplikacji mogą skorzystać z niej, aby zapoznawać się z parametrami dźwięku – czyli dla edukacji, inni mogą testować projekty filtrów, poprzez podawanie sygnału dźwiękowego na wejście filtru. Generator dźwięku może posłużyć do badań naukowych, testowania słuchu (na przykład częstotliwości, których już nie słyszymy), czy dla twórców muzyki, którzy chcą eksperymentować ze znalezieniem pożądanego brzmienia.

Sposób działania i efekty: zależy mi na tym, aby fala nie tylko była generowana, ale chcę, by była wizualizowana na wykresie. Naturalnie użytkownik może ustawić dowolne parametry typu: częstotliwość, amplituda, kształt fali (np. sinusoidalna, prostokątna, trójkątna) czy czas trwania dźwięku. Następnie chcę, by użytkownik wcisnął przycisk „generuj” i żeby dźwięk trwał do momentu, gdy użytkownik wcisnie przycisk „stop”. Gdy to nastąpi, dźwięk ma się zapisać na liście dźwięków wytworzonych w danej sesji. Chcę też zaimplementować możliwość edytowania generowanego dźwięku w czasie rzeczywistym.

Potrzebne pliki:

- *Pliki źródłowe Javy z kodem aplikacji*
- *Zewnętrzne biblioteki potrzebne do implementacji obsługi interfejsu użytkownika (jeszcze nie zdecydowałam, jakie konkretnie, możliwe, że będą to JavaFX, ControlsFX, JFoenix czy ChartFX)*
- *Pliki zasobów niezbędne do wyświetlania interfejsu użytkownika: na przykład ikon lub teł*
- *Pliki konfiguracyjne: parametry zawierające ścieżki plików, język domyślny lub wygląd interfejsu*
- *Pliki interfejsu użytkownika: pliki szablonów (JavaScript, CSS, HTML), uważam, że łatwiej mi będzie skorzystać właśnie z szablonów, niż implementować coś od „zera”*

Adnotacja 2. Diagram klas (struktura podstawowa, do późniejszego uzupełnienia)



Adnotacja 3. Opis przeznaczenia poszczególnych klas

- *BaseAbstractSound* – klasa abstrakcyjna, służy do ustawiania parametrów dźwięków, jako metody posiada tylko gettery i setery, planuję stworzyć więcej klas dziedziczących po niej
- *SoundPlayer* – klasa dziedziczy po klasie *WfmGenerator*, służy do odtwarzania dźwięku na podstawie wygenerowanego sygnału w klasie *WfmGenerator*
- *SoundEditor* – klasa dziedziczy po klasie *WfmGenerator* i ma metody potrzebne do zmian parametrów fali w czasie rzeczywistym
- *BaseSessionManager* – klasa abstrakcyjna, służy do podstawowych operacji na sesjach tworzenia dźwięków
- *SoundManager* – dziedziczy po *BaseSessionManager*, służy do operacji na wybranych dźwiękach
- *ReportsFetcher* – zbiera informacje takie jak parametry ustawiane przez użytkownika, czy parametry wyników pracy danych metod i przechowuje je, aby inna klasa mogła z nich skorzystać, ponadto implementuje interfejs aplikacji w sieci.

Komentarz, podsumowanie:

Planuję zmienić strukturę klas, będą to raczej kosmetyczne zmiany typu dodanie nowych klas dziedziczących po klasach abstrakcyjnych, brakuje klasy, która implementuje na przykład rysowanie wykresów sygnałów fali, chcę dodać możliwość składania dźwięków, na przykład z sygnału trójkątnego i sinusoidalnego, dalej, klasa ReportFetcher na tym diagramie jest bardzo uproszczona i posiada za dużo różnych metod. Całe zagadnienie wychwytywania i gromadzenia informacji zaimplementuję prawdopodobnie z wykorzystaniem dziedziczenia po klasie abstrakcyjnej klas, które będą implementować jakieś metody interfejsów.