



POLITECHNIKA
KOSZALIŃSKA

Zastosowania programowania komponentowego
Wyświetlacz słupkowy do miksera audio

Realizacja etapu 1

PAWEŁ SZELAĞ U-14871

SEBASTIAN TALAROWSKI U-14877

1. Realizacja etapu

Na podstawie wytycznych z zajęć zastosowania programowania komponentowego zostały zrealizowane następujące elementy w ramach etapu 1:

- Prezentacja użycia komponentu Java Beans w aplikacji testującej
- Prezentacja użycia komponentu .NET w aplikacji testującej
- Komentarz dotyczący porównania komponentów

2. Zrealizowane funkcjonalności

Głównym celem etapu było stworzenie prototypów o uproszczonej funkcjonalności w dwóch technologiach: Java Beans oraz .NET.

2.1. Cechy funkcjonalne

Wyświetlacz miksera audio według przyjętych założeń powinien wyświetlać stan następujących parametrów:

- Poziom głośności (*Volume level*)
- Poziom wzmocnienia tonów niskich (*Bass level*)
- Poziom wzmocnienia dźwięku na różnych częstotliwościach

2.2. Cechy нефunkcjonalne

W ramach cech pozafunkcyjnych komponent powinien cechować:

- Responsywność (zmiana skali słupków miksera)
- Kolor słupków
- Kolor tła
- Kolor ramki

W ramach etapu 1 funkcjonalności zostały uproszczone w celu stworzenia prototypu nadającego się do zastosowania w aplikacji testującej.

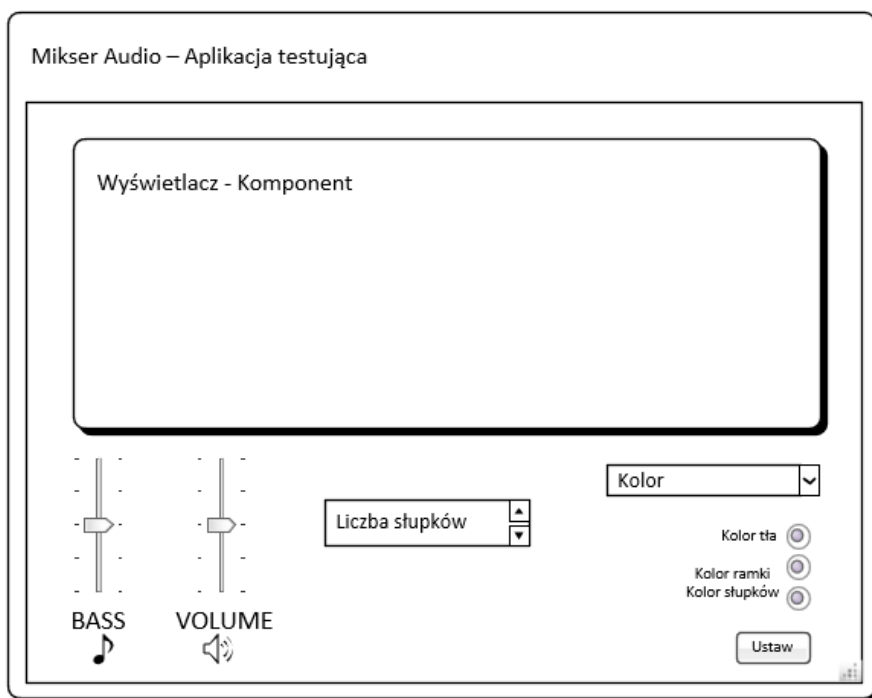
Zrealizowano w uproszczonej wersji m.in.:

- Regulację poziomu głośności oraz basu
- Regulację poziomu częstotliwości
- Zmianę kolorów tła, ramki oraz słupków

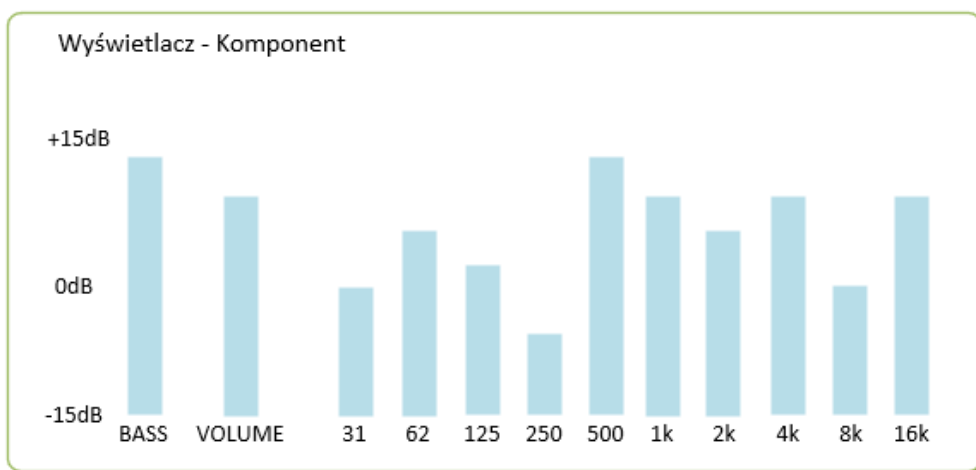
3. Prezentacja prototypu komponentu

3.1. Przegląd wizualny projektu

Podstawą stworzenia prototypu komponentu miksera audio był poniższy szkielet wizualny. Celem jest zaprojektowanie ujednoliconego projektu zarówno dla technologii Java Beans oraz .NET.



Rysunek 1. Osadzenie komponentu w aplikacji testującej



Rysunek 2. Przegląd wizualny komponentu – wyświetlacz miksera audio

3.2. Prototypowanie – technologia Java Beans

3.2.1. Zastosowane środowisko

Podczas prototypowania przyjęto środowisko programistyczne Netbeans. Utworzono dwa projekty: dla aplikacji testującej oraz komponentu. Projekt komponentu umożliwił wygenerowanie pliku o rozszerzeniu .JAR, który nadał się do zastosowania w aplikacji testującej.

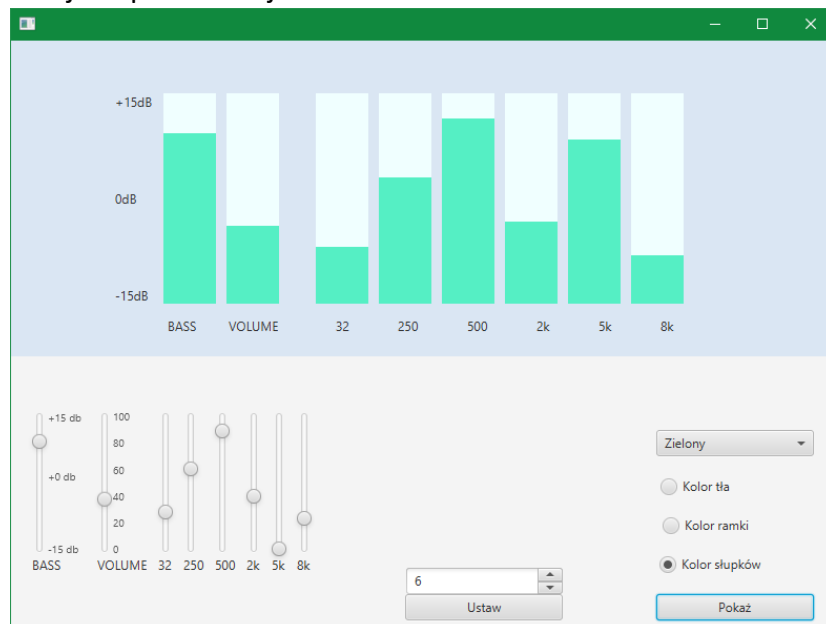
3.2.2. Implementacja komponentu

Przyjęto, że komponent składa się z następujących części składowych:

- Panel (*AudioDisplay*) – służy do osadzenia elementów na komponencie
- Słupek (*VerticalBar*) – reprezentuje pojedynczy słupek dla takich parametrów jak: poziom basu, głośności lub wzmocnienia częstotliwości
- Napis (*Label*) – zastosowany do opisu poszczególnych parametrów na komponencie
- Generator częstotliwości (*FrequencyGenerator*) – zastosowany do generowania poziomu częstotliwości dla różnej ilości słupków

Elementy tj. panel, słupki oraz napis zostały generowane we współpracy bibliotek graficznych Java.

3.2.3. Prezentacja implementacji



Rysunek 3. Implementacja prototypu w Java Beans

3.3. Prototypowanie – technologia .NET

3.3.1. Zastosowane środowisko

Podczas prototypowania przyjęto środowisko programistyczne VisualStudio 2019.

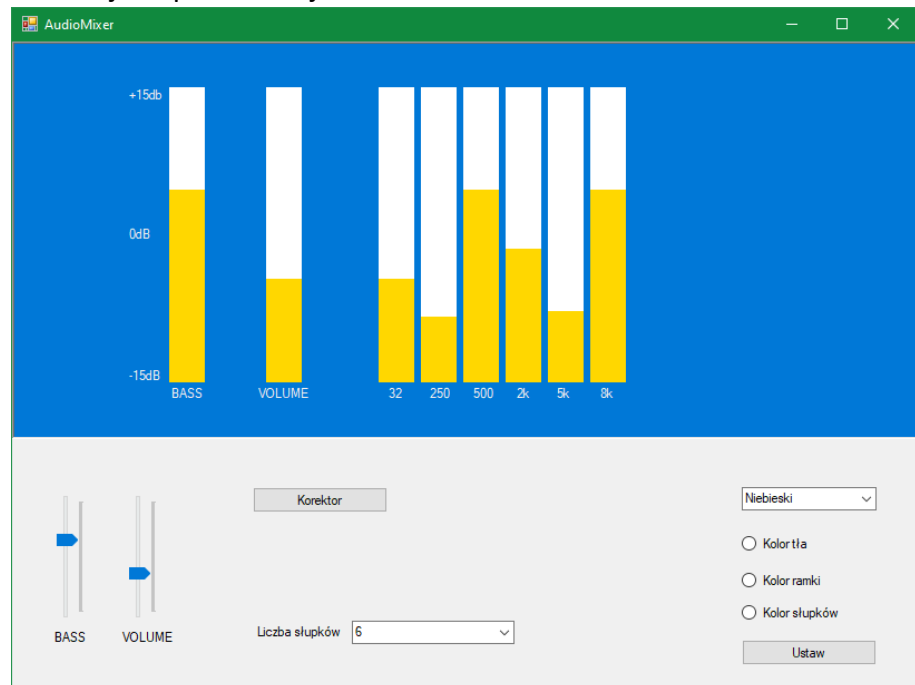
Utworzono dwa projekty: dla aplikacji testującej oraz komponentu.

Środowisko automatycznie wygenerowało plik o rozszerzeniu .dll, który nadał się do przeciągnięcia na formatkę w projekcie dla aplikacji testującej.

3.3.2. Implementacja

Elementy takie jak słupek zostały stworzone wraz z użyciem bibliotek graficznych .NET.

3.3.3. Prezentacja implementacji



Rysunek 4. Implementacja prototypu w .NET

4. Porównanie komponentów

Implementacja części składowych w obu technologiach jest zbliżona do siebie. (zbliżony model klas). Zarówno w .NET oraz w Java Beans zastosowano biblioteki graficzne, które definiują kształt oraz rozmiar słupków dla miksera audio.

Największą różnicą wydaje się wdrożenie komponentu do aplikacji. W technologii .NET należy utworzyć dowiązanie/zależność do pliku .dll lub zaimportować projekt do rozwiązania. W technologii Java Beans należy wskazać zewnętrzną bibliotekę w postaci pliku .JAR.

5. Repozytorium projektu

Rezultat oraz postęp projektu w ramach etapu 1 został udokumentowany w serwisie GitHub pod adresem:

<https://github.com/Pawlo2020/MikserAudio>