Last Ditch Player

Dokumentacja architekturalna

1. Wstęp

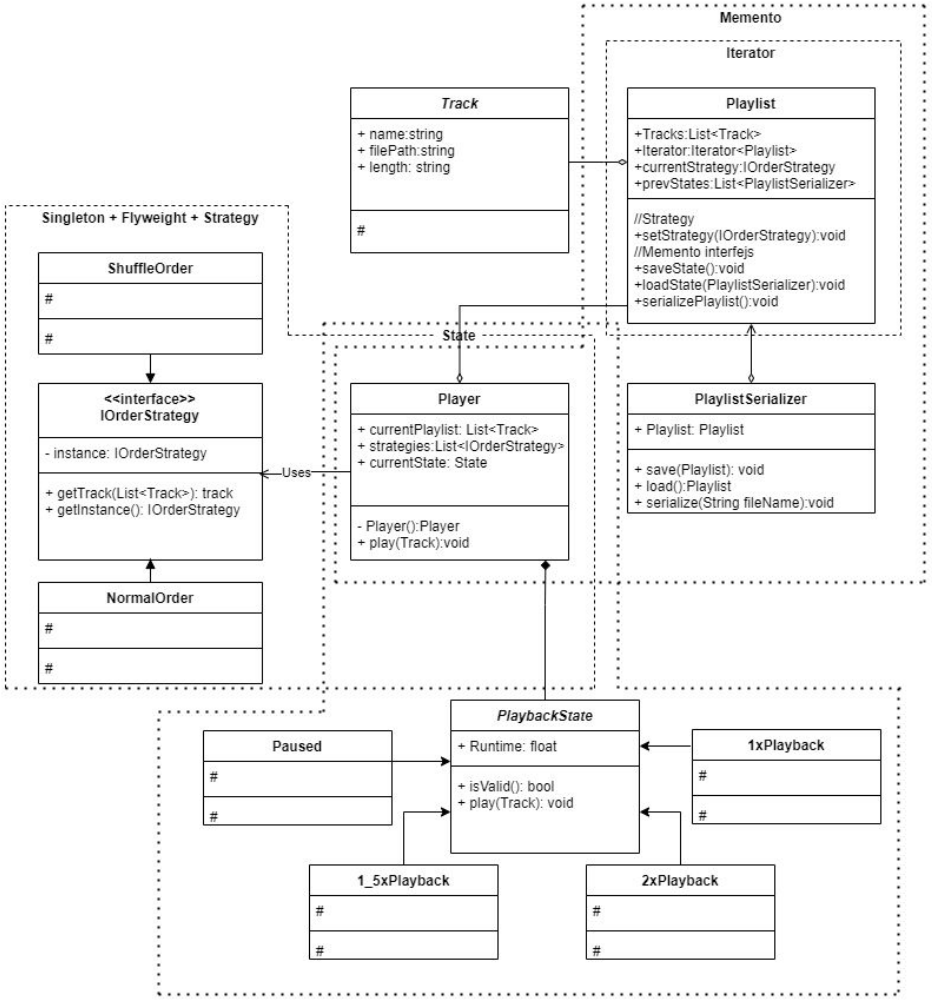
1.1. Opis

Last Ditch Player (w skrócie LDP) to prosty odtwarzacz plików audio. Umożliwiający odtworzenie kilku różnych typów audio, manipulację playlistami i ich zapisywanie.

1.2. Funkcjonalności

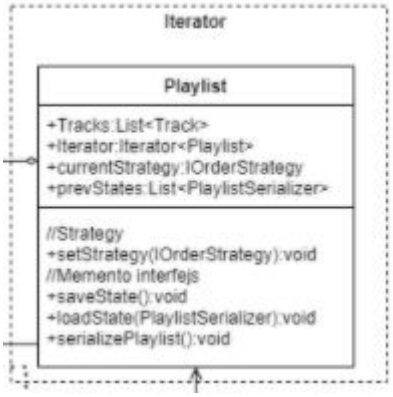
- Odtwarzanie plików audio (kilku formatów)
- Tworzenie playlist (dodawanie utworów)
- Możliwość zatrzymywania, odtwarzania w przyśpieszeniu
- Możliwość odtwarzania losowego z playlisty
- Możliwość zapisywania playlist na dysku i wczytywanie ich

2. Diagram klas



3. Opis wzorców

3.1. Iterator

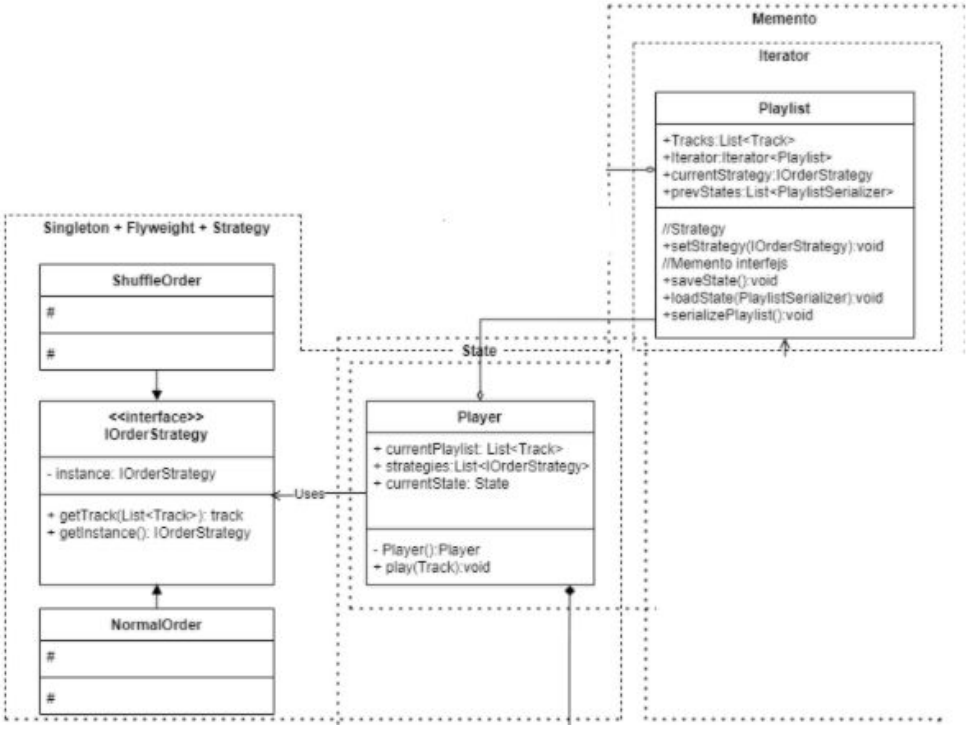


Iterator - zapewnia dostęp do elementów

Używany w naszym programie do iteracji po utworach składających się na playlistę.

Zastosowanie własnego iteratora zamiast domyślnych dekoratorów pozwala nam na zmianę strategii losowania następnego utworu.

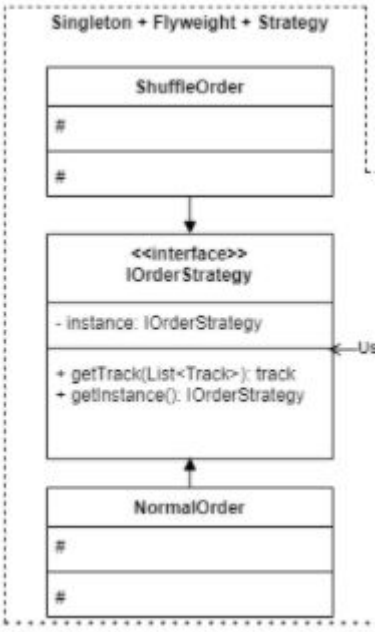
3.2. Strategy



Strategy - zapewnia możliwość zmiany implementacji fragmentu programu w trakcie jego pracy

Nasz LDP pozwala na zmianę strategii losowania piosenki z playlisty, do tego potrzebujemy możliwości zmiany implementacji za to odpowiedzialnej w trakcie działania programu.

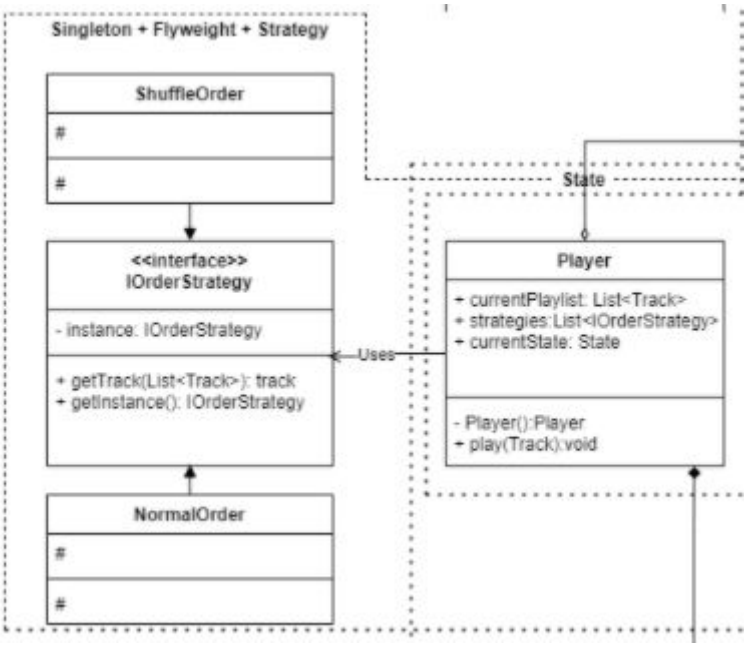
3.3. Singleton



Singleton - gwarantuje istnieje jedynie jednej instancji klasy

W naszym rozwiązaniu każda ze strategii jest singletonem, aby być pewnym że będzie występować tylko jedna instancja, ponieważ nie chcemy tworzyć nowych obiektów dla każdej playlisty.

3.4. Flyweight

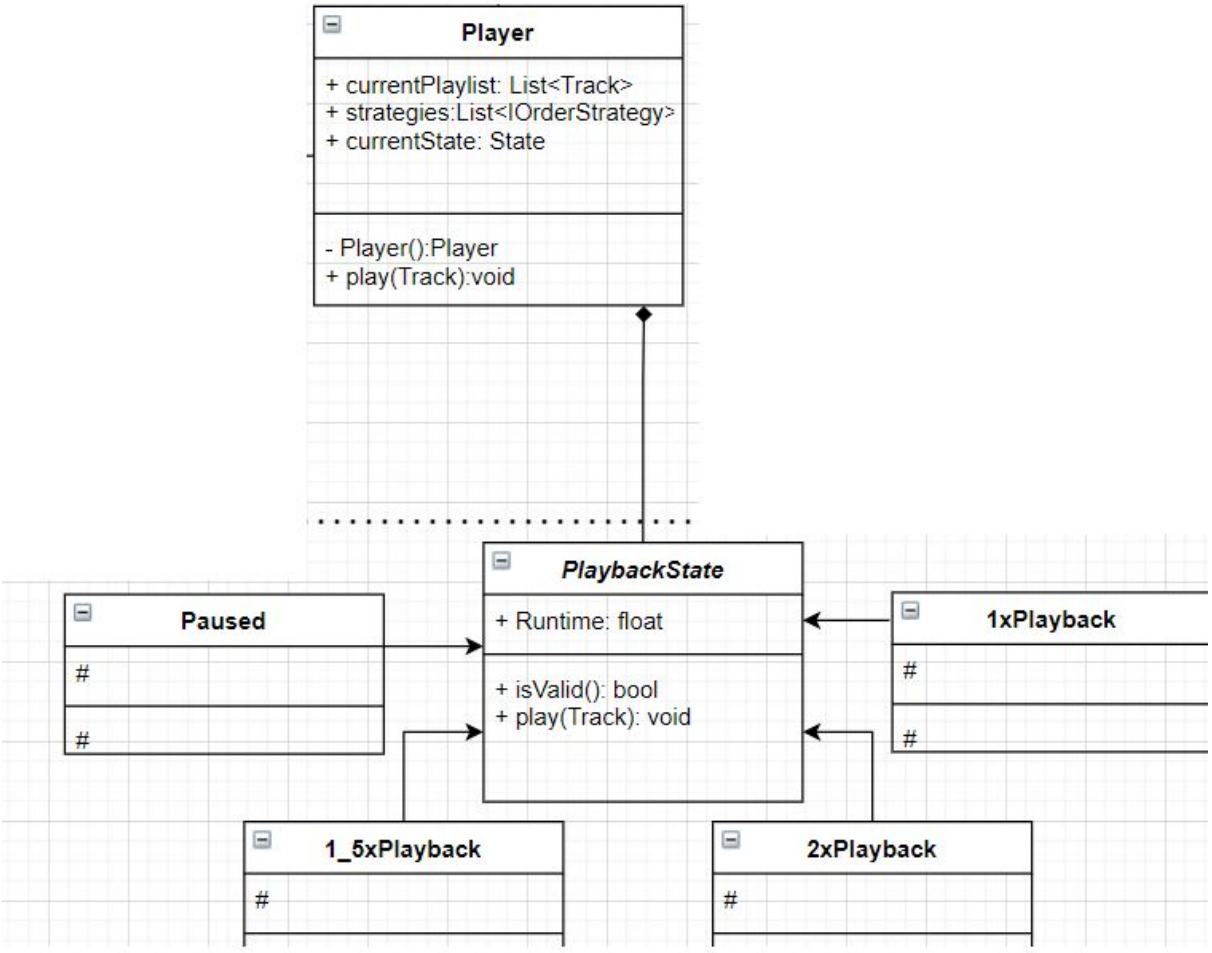


Flyweight - współdzielenie stanu wewnętrznego i oddzielenie od stanu zewnętrznego

W naszym rozwiązaniu flyweight pozwala nam na użycie jednej instancji strategii w wielu playlistach. Pozwala nam to zaoszczędzić pamięć, jak i przyśpieszyć działanie, ponieważ pomijamy tworzenie nowych obiektów.

3.5. State

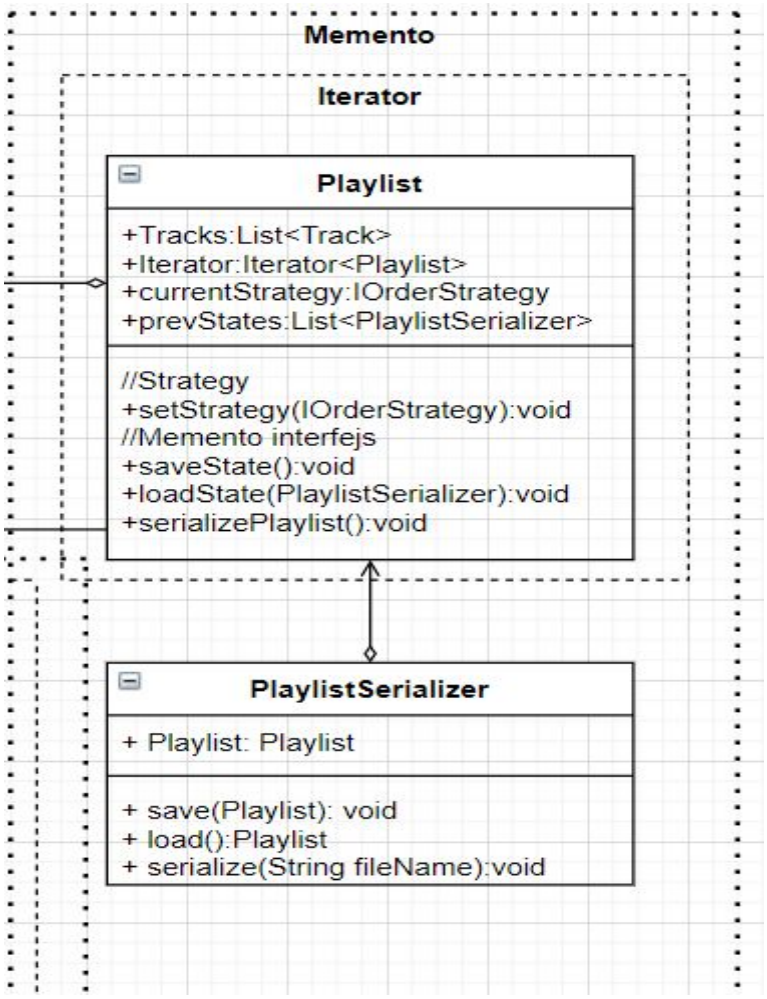
Wzorzec stan zmienia zachowanie klasy, gdy stan wewnętrzny obiektu się zmienia.



Stan kontekstu (klasy **Player)** wpływa na tempo próbkowania pliku dźwiękowego w zależności od akcji użytkownika.

3.6. Memento

Wzorzec memento pozwala na zapamiętanie obecnego stanu klasy, nie łamiąc przy tym zasady enkapsulacji.



W naszym programie wzorzec memento jest wykorzystywany do trzymania w pamięci stanu playlisty, gdy użytkownik zrezygnuje z wprowadzania w niej zmian. W zaimplementowanym wzorcu zarówno kontekstem jak i opiekunem memento jest klasa **Playlist**.