Języki i paradygmaty programowania

**Sprawozdanie z pracy projektowej**

**Program na zaliczenie**

Adrian Zalewski 10041

**Informatyka, sem. III**

**2017/ 2018**

* **Najważniejsze klasy warstwy danych projektu**

**Class Database**

public class **Database**

extends java.lang.Object

Klasa Database, w której zainicjowana jest ArrayLista, która jest bufforem danych w programie.

**Author:** Adrian Zalewski

* + ***Constructor Summary***

|  |
| --- |
| **Constructors** |
| **Constructor and Description** |
| [**Database**](file:///C:\Users\zalew\eclipse-workspace\Projekt\doc\Database.html#Database--)() |

* + ***Method Summary***

|  |  |
| --- | --- |
| **All Methods**[**Instance Methods**](javascript:show(2);)[**Concrete Methods**](javascript:show(8);) | |
| **Modifier and Type** | **Method and Description** |
| java.lang.String | [**toString**](file:///C:\Users\zalew\eclipse-workspace\Projekt\doc\Database.html#toString--)()  Metoda toString, która odpowiada za wygląd bazy danych w oknie konsoli. |

* + ***Constructor Detail***
    - **Database**

public Database()

* + ***Method Detail***
    - **toString**

public java.lang.String toString()

Metoda toString, która odpowiada za wygląd bazy danych w oknie konsoli.

**Overrides:**

toString in class java.lang.Object

## Class Muzyka

Klasa Muzyka to jedna z najważniejszych warstw danych w moim programie. Zainicjowane są tu typy danych jakie przechodzą przez buffor. Są tu zdefiniowane metody getterów i setterów.

Author:Adrian Zalewski

* + ***Constructor Summary***

|  |
| --- |
| **Constructors** |
| **Constructor and Description** |
| [Muzyka](file:///C:\Users\zalew\eclipse-workspace\Projekt\doc\Muzyka.html#Muzyka--)() |
| [Muzyka](file:///C:\Users\zalew\eclipse-workspace\Projekt\doc\Muzyka.html#Muzyka-int-java.lang.String-java.lang.String-java.lang.String-int-)(int ID, java.lang.String title, java.lang.String artist, java.lang.String album, int year) |

* + ***Method Summary***

|  |  |
| --- | --- |
| **All Methods**[**Instance Methods**](javascript:show(2);)[**Concrete Methods**](javascript:show(8);) | |
| **Modifier and Type** | **Method and Description** |
| int | [compareTo](file:///C:\Users\zalew\eclipse-workspace\Projekt\doc\Muzyka.html#compareTo-Muzyka-)([**Muzyka**](file:///C:\Users\zalew\eclipse-workspace\Projekt\doc\Muzyka.html) a)  compareTo porównuje ID (ASC) |
| java.lang.String | [getAlbum](file:///C:\Users\zalew\eclipse-workspace\Projekt\doc\Muzyka.html#getAlbum--)() |
| java.lang.String | [getArtist](file:///C:\Users\zalew\eclipse-workspace\Projekt\doc\Muzyka.html#getArtist--)() |
| int | [getID](file:///C:\Users\zalew\eclipse-workspace\Projekt\doc\Muzyka.html#getID--)() |
| java.lang.String | [getTitle](file:///C:\Users\zalew\eclipse-workspace\Projekt\doc\Muzyka.html#getTitle--)() |
| int | [getYear](file:///C:\Users\zalew\eclipse-workspace\Projekt\doc\Muzyka.html#getYear--)() |
| void | [setAlbum](file:///C:\Users\zalew\eclipse-workspace\Projekt\doc\Muzyka.html#setAlbum-java.lang.String-)(java.lang.String album) |
| void | [setArtist](file:///C:\Users\zalew\eclipse-workspace\Projekt\doc\Muzyka.html#setArtist-java.lang.String-)(java.lang.String artist) |
| void | [setID](file:///C:\Users\zalew\eclipse-workspace\Projekt\doc\Muzyka.html#setID-int-)(int ID) |
| void | [setTitle](file:///C:\Users\zalew\eclipse-workspace\Projekt\doc\Muzyka.html#setTitle-java.lang.String-)(java.lang.String title) |
| void | [setYear](file:///C:\Users\zalew\eclipse-workspace\Projekt\doc\Muzyka.html#setYear-int-)(int year) |
| java.lang.String | [toString](file:///C:\Users\zalew\eclipse-workspace\Projekt\doc\Muzyka.html#toString--)()  Metoda toString, która definiuje postać jednej linjki bazy danych w oknie konsoli. |

* + ***Constructor Detail***
    - **Muzyka**
    - public Muzyka(int ID,
    - java.lang.String title,
    - java.lang.String artist,
    - java.lang.String album,

int year)

* + - **Muzyka**

public Muzyka()

* + ***Method Detail***
    - **getID**

public int getID()

* + - **setID**

public void setID(int ID)

* + - **getTitle**

public java.lang.String getTitle()

* + - **setTitle**

public void setTitle(java.lang.String title)

* + - **getArtist**

public java.lang.String getArtist()

* + - **setArtist**

public void setArtist(java.lang.String artist)

* + - **getAlbum**

public java.lang.String getAlbum()

* + - **setAlbum**

public void setAlbum(java.lang.String album)

* + - **getYear**

public int getYear()

* + - **setYear**

public void setYear(int year)

* + - **compareTo**

public int compareTo([Muzyka](file:///C:\Users\zalew\eclipse-workspace\Projekt\doc\Muzyka.html) a)

compareTo porównuje ID (ASC)

Specified by:

compareTo in interface java.lang.Comparable<[Muzyka](file:///C:\Users\zalew\eclipse-workspace\Projekt\doc\Muzyka.html)>

* + - **toString**

public java.lang.String toString()

Metoda toString, która definiuje postać jednej linjki bazy danych w oknie konsoli.

Overrides:

toString in class java.lang.Object

**Class Methods**

Wszystkie Metody

**Author:** Adrian Zalewski

* + ***Constructor Summary***

|  |
| --- |
| **Constructors** |
| **Constructor and Description** |
| [**Methods**](file:///C:\Users\zalew\eclipse-workspace\Projekt\doc\Methods.html#Methods--)() |

* + ***Method Summary***

|  |  |
| --- | --- |
| **All Methods**[**Instance Methods**](javascript:show(2);)[**Concrete Methods**](javascript:show(8);) | |
| **Modifier and Type** | **Method and Description** |
| void | [**AboutApp**](file:///C:\Users\zalew\eclipse-workspace\Projekt\doc\Methods.html#AboutApp--)()  Metoda do otworzenia okienka "O Programie" |
| void | [**AddEmptyRow**](file:///C:\Users\zalew\eclipse-workspace\Projekt\doc\Methods.html#AddEmptyRow-javax.swing.JTable-)(javax.swing.JTable table)  Dodanie pustego wiersza. |
| void | [**AddObject**](file:///C:\Users\zalew\eclipse-workspace\Projekt\doc\Methods.html#AddObject--)()  Dodawanie Objektu w trybie tekstowy. |
| void | [**AddRowToJTable**](file:///C:\Users\zalew\eclipse-workspace\Projekt\doc\Methods.html#AddRowToJTable-javax.swing.JTable-javax.swing.JTable-)(javax.swing.JTable table, javax.swing.JTable table\_1)  Dodawanie rekordów do tabeli. |
| void | [**ClearTable**](file:///C:\Users\zalew\eclipse-workspace\Projekt\doc\Methods.html#ClearTable-javax.swing.JTable-javax.swing.JTable-)(javax.swing.JTable table, javax.swing.JTable table\_1)  Czyszczenie tabeli |
| void | [**ConnectSQL**](file:///C:\Users\zalew\eclipse-workspace\Projekt\doc\Methods.html#ConnectSQL--)()  Pobieranie danych z bazy SQL do ArrayListy. |
| void | [**Log**](file:///C:\Users\zalew\eclipse-workspace\Projekt\doc\Methods.html#Log--)()  Metoda Logowania do trybu gragicznego |
| void | [**ReadFromXML**](file:///C:\Users\zalew\eclipse-workspace\Projekt\doc\Methods.html#ReadFromXML--)()  Odczyt z XML |
| void | [**RemoveObject**](file:///C:\Users\zalew\eclipse-workspace\Projekt\doc\Methods.html#RemoveObject--)()  Usuwanie obiektu w trybie tekstowym. |
| void | [**RemoveSelectedRow**](file:///C:\Users\zalew\eclipse-workspace\Projekt\doc\Methods.html#RemoveSelectedRow-javax.swing.JTable-)(javax.swing.JTable table)  Usuwanie zaznaczonego wiersza |
| void | [**SaveToArrayList**](file:///C:\Users\zalew\eclipse-workspace\Projekt\doc\Methods.html#SaveToArrayList-javax.swing.JTable-)(javax.swing.JTable table)  Zapis do ArrayListy. |
| void | [**SaveToSQL**](file:///C:\Users\zalew\eclipse-workspace\Projekt\doc\Methods.html#SaveToSQL--)()  Zapis do SQL |
| void | [**SaveToXML**](file:///C:\Users\zalew\eclipse-workspace\Projekt\doc\Methods.html#SaveToXML--)()  Zapis do XML |
| void | [**ShowDataBase**](file:///C:\Users\zalew\eclipse-workspace\Projekt\doc\Methods.html#ShowDataBase--)()  Pokazywanie bazy danych w trybie tekstowym. |

* + ***Constructor Detail***
    - **Methods**

public Methods()

* + ***Method Detail***
    - **Log**

public void Log()

Metoda Logowania do trybu gragicznego

* + - **AddRowToJTable**
    - public void AddRowToJTable(javax.swing.JTable table,

javax.swing.JTable table\_1)

Dodawanie rekordów do tabeli.

**Parameters:**

table -

table\_1 -

* + - **ConnectSQL**

public void ConnectSQL()

Pobieranie danych z bazy SQL do ArrayListy.

* + - **ReadFromXML**

public void ReadFromXML()

Odczyt z XML

* + - **AddEmptyRow**

public void AddEmptyRow(javax.swing.JTable table)

Dodanie pustego wiersza.

**Parameters:**

table -

* + - **SaveToArrayList**

public void SaveToArrayList(javax.swing.JTable table)

Zapis do ArrayListy.

**Parameters:**

table -

* + - **SaveToSQL**

public void SaveToSQL()

Zapis do SQL

* + - **SaveToXML**

public void SaveToXML()

Zapis do XML

* + - **ClearTable**
    - public void ClearTable(javax.swing.JTable table,

javax.swing.JTable table\_1)

Czyszczenie tabeli

**Parameters:**

table -

table\_1 -

* + - **RemoveSelectedRow**

public void RemoveSelectedRow(javax.swing.JTable table)

Usuwanie zaznaczonego wiersza

**Parameters:**

table -

* + - **AboutApp**

public void AboutApp()

Metoda do otworzenia okienka "O Programie"

* + - **ShowDataBase**

public void ShowDataBase()

Pokazywanie bazy danych w trybie tekstowym.

* + - **AddObject**

public void AddObject()

Dodawanie Objektu w trybie tekstowy.

* + - **RemoveObject**

public void RemoveObject()

Usuwanie obiektu w trybie tekstowym.

* **Najważniejsze klasy warstwy logiki**

W moim programie są 4 klasy z warstwy logiki, lecz są one wszystkie podobne, różnią się tylko atrybutem sortowania, dlatego zamieszczam tylko JavaDoc tylko z jednej klasy.

## Class CompAl

* java.lang.Object
  + CompAl
* **All Implemented Interfaces:**

java.util.Comparator<[Muzyka](file:///C:\Users\zalew\eclipse-workspace\Projekt\doc\Muzyka.html)>

public class CompAl

extends java.lang.Object

implements java.util.Comparator<[Muzyka](file:///C:\Users\zalew\eclipse-workspace\Projekt\doc\Muzyka.html)>

Klasa warstwy logicznej, która odpowiada za sortowanie według nazwy Albumu.

Author:

Adrian Zalewski

* + ***Constructor Summary***

|  |
| --- |
| **Constructors** |
| **Constructor and Description** |
| [CompAl](file:///C:\Users\zalew\eclipse-workspace\Projekt\doc\CompAl.html#CompAl--)() |

* + ***Method Summary***

|  |  |
| --- | --- |
| **All Methods**[**Instance Methods**](javascript:show(2);)[**Concrete Methods**](javascript:show(8);) | |
| **Modifier and Type** | **Method and Description** |
| int | [compare](file:///C:\Users\zalew\eclipse-workspace\Projekt\doc\CompAl.html#compare-Muzyka-Muzyka-)([**Muzyka**](file:///C:\Users\zalew\eclipse-workspace\Projekt\doc\Muzyka.html) o1, [**Muzyka**](file:///C:\Users\zalew\eclipse-workspace\Projekt\doc\Muzyka.html) o2) |

* + - **CompAl**

public CompAl()

* + ***Method Detail***
    - **compare**
    - public int compare([Muzyka](file:///C:\Users\zalew\eclipse-workspace\Projekt\doc\Muzyka.html) o1,

[Muzyka](file:///C:\Users\zalew\eclipse-workspace\Projekt\doc\Muzyka.html) o2)

Specified by:

compare in interface java.util.Comparator<[Muzyka](file:///C:\Users\zalew\eclipse-workspace\Projekt\doc\Muzyka.html)>

* **Najważniejsze klasy warstwy interfejsu**

## Class Window

public class Window

extends javax.swing.JFrame

Klasa Window to główna klasa interfejsu graficznego, w której zawarte są komponenty całego wyglądu programu, oraz odwołania do metod.

Author:

Adrian Zalewski

See Also:

[Serialized Form](file:///C:\Users\zalew\eclipse-workspace\Projekt\doc\serialized-form.html#Window)

* + ***Constructor Summary***

|  |
| --- |
| **Constructors** |
| **Constructor and Description** |
| [Window](file:///C:\Users\zalew\eclipse-workspace\Projekt\doc\Window.html#Window--)()  Główna Metoda interfejsu graficznego, w której zawarte są wszystkie komponenty. |

* + ***Method Summary***

|  |  |
| --- | --- |
| **All Methods**[**Static Methods**](javascript:show(1);)[**Concrete Methods**](javascript:show(8);) | |
| **Modifier and Type** | **Method and Description** |
| static void | [main](file:///C:\Users\zalew\eclipse-workspace\Projekt\doc\Window.html#main-java.lang.String:A-)(java.lang.String[] args)  Główna metoda klasy Window, w której określony jest rozmiar, kolor okna itp. |

* + - **Window**
    - public Window()
    - throws javax.xml.parsers.ParserConfigurationException,
    - org.xml.sax.SAXException,

java.io.IOException

Główna Metoda interfejsu graficznego, w której zawarte są wszystkie komponenty.

Throws:

javax.xml.parsers.ParserConfigurationException

org.xml.sax.SAXException

java.io.IOException

* + ***Method Detail***
    - **main**

public static void main(java.lang.String[] args)

Główna metoda klasy Window, w której określony jest rozmiar, kolor okna itp.

## Class Login

Klasa rozruchowa, odpowiedzialna za logowanie i wybór trybu w jakim ma pracować program (tekstowy/graficzny).

Author:

Adrian Zalewski

See Also:

[Serialized Form](file:///C:\Users\zalew\eclipse-workspace\Projekt\doc\serialized-form.html#Login)

* ***Constructor Summary***

|  |
| --- |
| **Constructors** |
| **Constructor and Description** |
| [Login](file:///C:\Users\zalew\eclipse-workspace\Projekt\doc\Login.html#Login--)()  Metoda Login(), to metoda, w której się logujemy oraz dokonujemy wyboru pomiędzy trybem graficznym, a tekstowym. |

* + ***Method Summary***

|  |  |
| --- | --- |
| **All Methods**[**Static Methods**](javascript:show(1);)[**Concrete Methods**](javascript:show(8);) | |
| **Modifier and Type** | **Method and Description** |
| static void | [main](file:///C:\Users\zalew\eclipse-workspace\Projekt\doc\Login.html#main-java.lang.String:A-)(java.lang.String[] args)  Metoda main, odpowiada za uruchomienie okienka logowania. |

* + ***Constructor Detail***
    - **Login**

public Login()

Metoda Login(), to metoda, w której się logujemy oraz dokonujemy wyboru pomiędzytrybem graficznym, a tekstowym.

***Method Detail***

* + - **main**

public static void main(java.lang.String[] args)

Metoda main, odpowiada za uruchomienie okienka logowania. Znajduje się w niej metoda log(), która uruchamia tryb graficzny, jeżeli dokonamy takiego wyboru.

**Class Tekst**

Klasa interfejsu tekstowego.

**Author:** Adrian Zalewski

* + ***Constructor Summary***

|  |
| --- |
| **Constructors** |
| **Constructor and Description** |
| [**Tekst**](file:///C:\Users\zalew\eclipse-workspace\Projekt\doc\Tekst.html#Tekst--)() |

* + ***Method Summary***

|  |  |
| --- | --- |
| **All Methods**[**Instance Methods**](javascript:show(2);)[**Concrete Methods**](javascript:show(8);) | |
| **Modifier and Type** | **Method and Description** |
| void | [**Tekstowy**](file:///C:\Users\zalew\eclipse-workspace\Projekt\doc\Tekst.html#Tekstowy--)()  Główna metoda interfejsu tekstowego. |

* + ***Constructor Detail***
    - **Tekst**

public Tekst()

* + ***Method Detail***
    - **Tekstowy**

public void Tekstowy()

Główna metoda interfejsu tekstowego.

## Class About

Klasa okienka "O programie"

Author:Adrian Zalewski

See Also:[Serialized Form](file:///C:\Users\zalew\eclipse-workspace\Projekt\doc\serialized-form.html#About)

* + ***Summary***

|  |
| --- |
| **Constructors** |
| **Constructor and Description** |
| [About](file:///C:\Users\zalew\eclipse-workspace\Projekt\doc\About.html#About--)()  Metoda w której zawarte są graficzne komponenty okienka. |

* + ***Method Summary***

|  |  |
| --- | --- |
| **All Methods**[**Static Methods**](javascript:show(1);)[**Concrete Methods**](javascript:show(8);) | |
| **Modifier and Type** | **Method and Description** |
| static void | [main](file:///C:\Users\zalew\eclipse-workspace\Projekt\doc\About.html#main-java.lang.String:A-)(java.lang.String[] args)  Metoda uruchamiająca okienko. |

* + ***Constructor Detail***
    - **About**

public About()

Metoda w której zawarte są graficzne komponenty okienka.

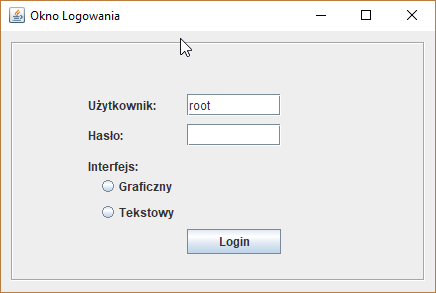
* + ***Method Detail***
    - **main**

public static void main(java.lang.String[] args)

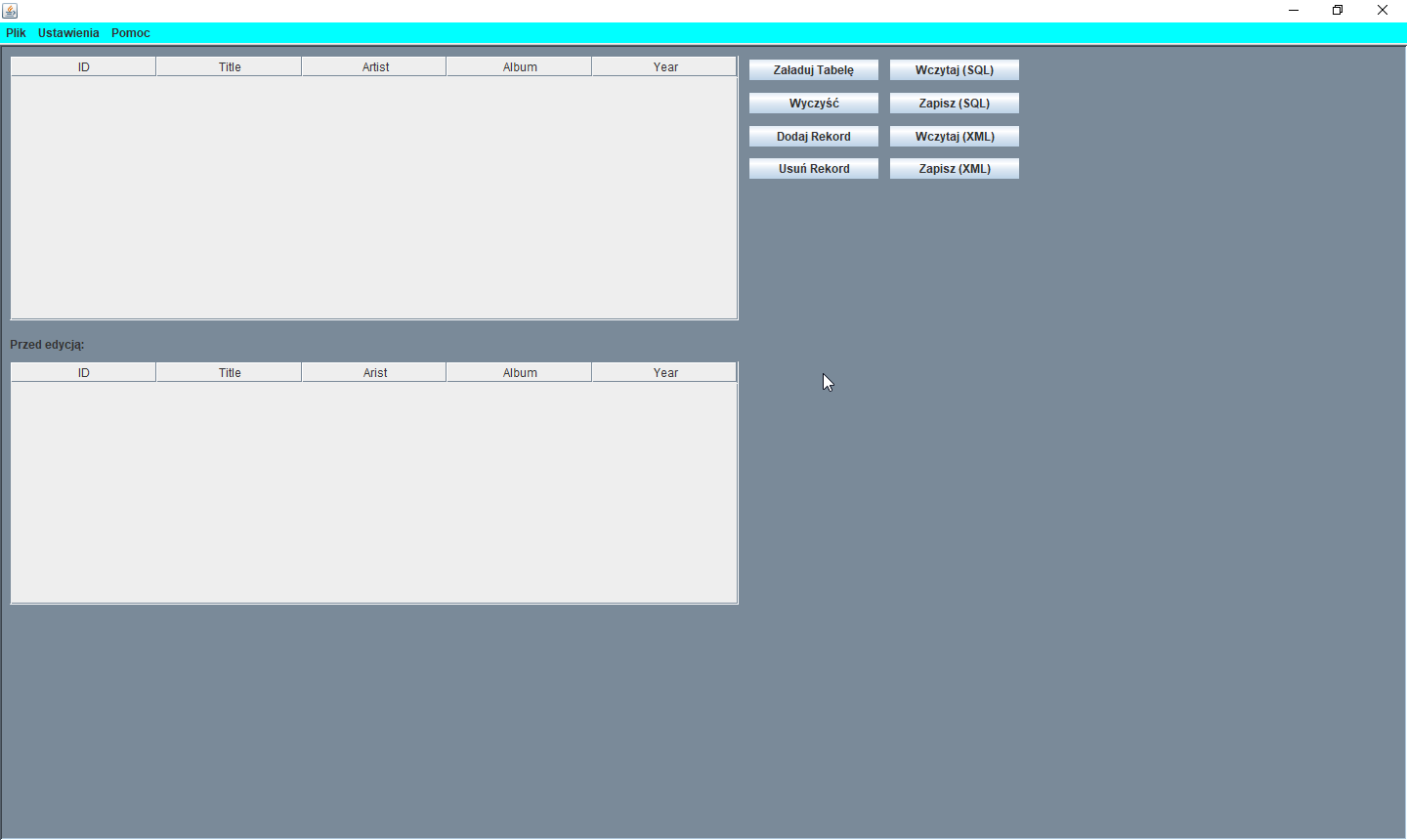
Metoda uruchamiająca okienko.

**Opis oraz zrzuty ekranu:**

Wszystko zaczyna się od okna Logowania, gdzie musimy wpisać nazwę użytkownika, oraz hasło (domyślnie jest to Root i brak hasła), a następnie wybrać tryb graficzny bądź tekstowy i nacisnąć login.



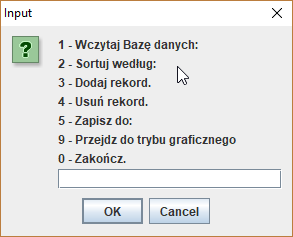
Jeżeli wybierzemy tryb graficzny naszym oczom ukaże się okno:



Pierwsza tabela, jest to ta, na której wykonujemy operacje, natomiast druga jest tylko do wglądu, aby móc porównać sobie to co załadowaliśmy na początku z tym co edytowaliśmy

Opis poszczególnych przycisków:

* Menu:
  + Plik
    - Wczytaj bazę danych z (SQL) (CTRL+F1) – wczytuje bazę danych z phpMyAdmin
    - Zapisz bazę danych do (SQL) (CTRL+F2) - zapisuje bazę danych do phpMyAdmin
    - Wczytaj z (XML) (CTRL+F3) – wczytuje bazę danych z pliku .xml
    - Zapisz do (XML) (CTRL+F4) – zapisuję bazę danych do pliku .xml
  + Ustawienia
    - Tryb tekstowy (CTRL+T) – wyłącza obecne okno graficzne, poczym włącza się tryb tekstowy
    - Wyloguj (CTRL+Q) – wyłącza obecne okno, i przełącza na okno logowania
  + Pomoc
    - O programie(CTRL + I) – wyświetla informacje o programie
* Przyciski obok tabeli
  + Załaduj tabelę – załadowuje wcześniej wczytane informacje do tabelki.
  + Wyczyść – czyści wszystkie rekordy w tabeli.
  + Dodaj rekord – dodaje pusty rekord z domyślnym następnym numerem ID
  + Usuń Rekord – usuwa zaznaczony rekord
  + Wczytaj (SQL) - wczytuje bazę danych z phpMyAdmin
  + Zapisz (SQL) - zapisuje bazę danych do phpMyAdmin
  + Wczytaj (XML) - wczytuje bazę danych z pliku .xml
  + Zapisz (XML) – zapisuję bazę danych do pliku .xml
* Nazwy kolumn – jeżeli klikniemy na nazwy kolumny to program zacznie sortować tabelę po danej kolumnie, pojawi się strzałka do wyboru sortowania (ASC/DESC)

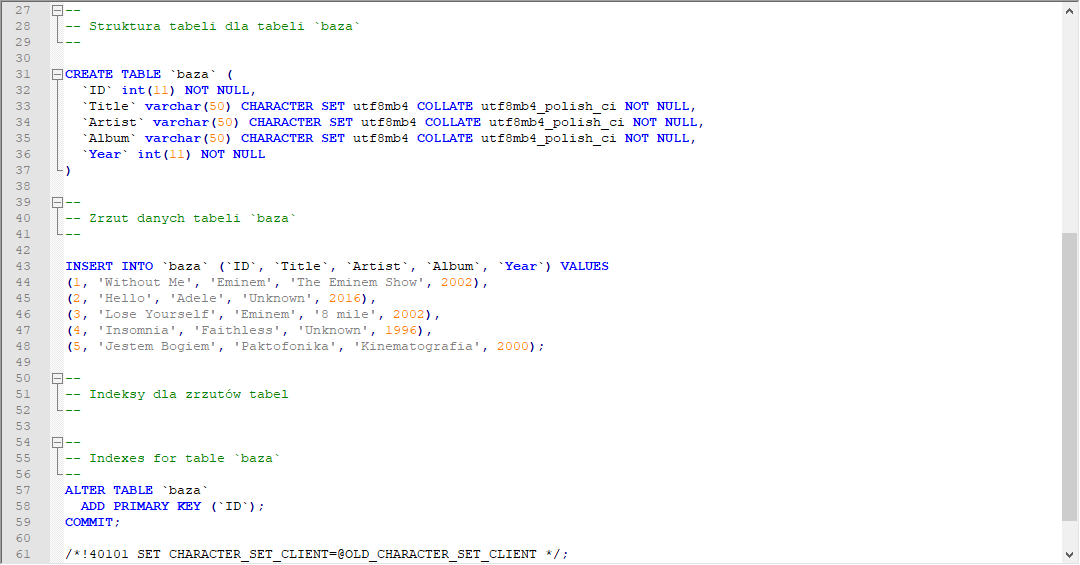
Jeżeli zaś wybierzemy tryb tekstowy to zobaczymy:  


Wpisanie:

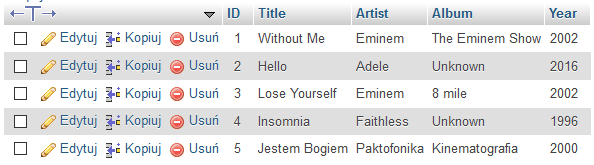
1. Pozwoli nam wczytać bazę (wybór SQL/XML lub cofnij)
2. Sortuj według: wybór atrybutu po, którym tabela ma być posortowany (ASC)
3. Dodanie rekordu (ID domyślnie większe o 1 od ostatniego + wpisywanie każdego atrybutu po kolei zgodnie z instrukcją.
4. Usuwanie rekordu wybierając jego ID
5. Zapisanie do (wybór SQL/XML lub cofnij)

9. Zamknięcie tego okienka i otwarcie trybu graficznego.  
0. Zakończenie programu

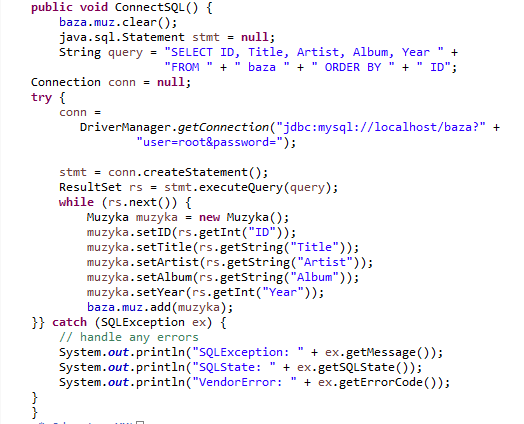
**4. Postać bazy danych z zapisanymi informacjami o zaprogramowanych stacjach. (zrzut ekranu, fragment kodu/tekstu itp.)**

Moja baza danych zaprojektowana jest w **MySQL**, z którą łączę się za pomocą programu **XAMPP.** Oto Fragment kodu jej tworzenia, oraz wprowadzenia 5 przykładowych rekordów (cały kod [tutaj](baza.sql))

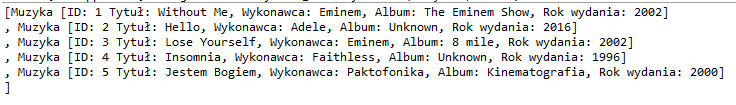
A oto zrzut ekranu, jak ona prezentuje się w narzędziu **phpMyAdmin:**



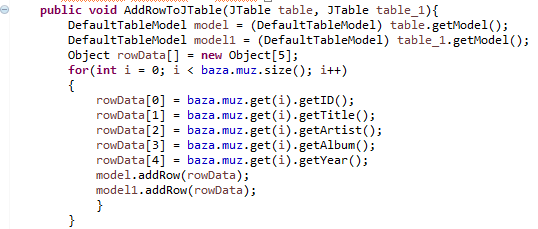
Prosto z bazy danych, konwertowana jest na ArrayListę



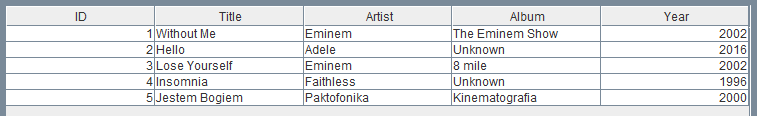
Która prezentuje się następująco w konsoli w trybie tekstowym:



Natomiast w trybie graficznym dane są wpisywane do tabelki.

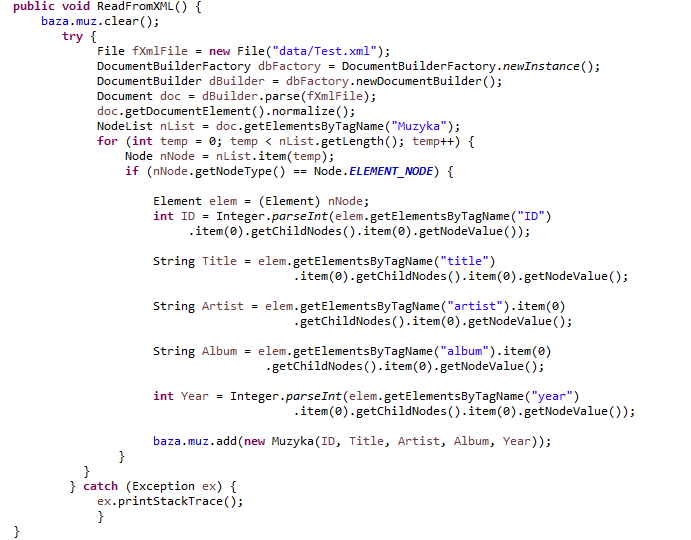


Prezentują się tam tak:

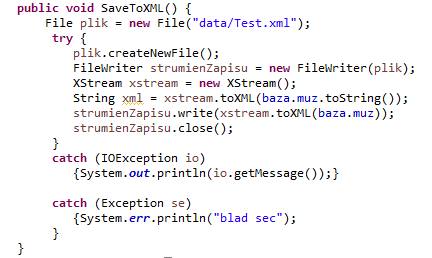


Istnieje również możliwość zapisu oraz wczytania do pliku z rozszerzeniem XML:

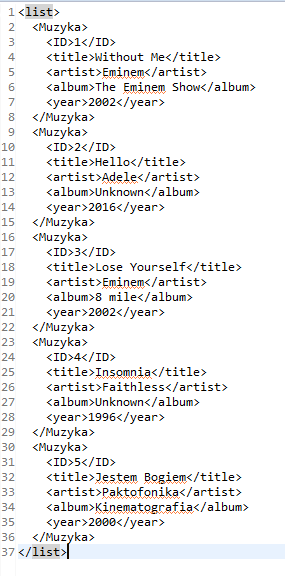
Wczytanie:



Oraz zapis (za pomocą XStreama):



Postać ta wygląda tak:



**5. Podsumowanie i ewentualne uwagi grupy nt. projektu.**

Reasumując, jestem zadowolony z tego co udało mi się już osiągnąć robiąc ten projekt, aczkolwiek brakuje mi paru punktów związanych z wymaganiami, dot. M.in. „StatusBara”.

Oczywiście przez najbliższy tydzień będę dążył do tego, aby wszystkie wymagania nt. projektu zostały spełnione. Oraz planuje rozszerzyć swój program.