

StatsSpotify

Generated by Doxygen 1.9.3



<b>1 Class Index</b>	<b>1</b>
1.1 Class List	1
<b>2 File Index</b>	<b>3</b>
2.1 File List	3
<b>3 Class Documentation</b>	<b>5</b>
3.1 SpotifyStats Class Reference	5
3.1.1 Detailed Description	6
3.1.2 Member Function Documentation	6
3.1.2.1 ArithmeticMeanOfListens()	6
3.1.2.2 ArtistSorter()	7
3.1.2.3 GenerateReport()	7
3.1.2.4 LeastPopularArtist()	9
3.1.2.5 LeastPopularSong()	10
3.1.2.6 MostPopularArtist()	10
3.1.2.7 MostPopularSong()	11
3.1.2.8 Readfile()	12
3.1.2.9 SaveDataToFile()	12
3.1.2.10 SpearmanKorelation()	12
3.1.3 Friends And Related Function Documentation	13
3.1.3.1 RaportsGeneratorInterface	13
<b>4 File Documentation</b>	<b>15</b>
4.1 Projekt_Zaliczeniowy_Spotify/StatSpotify.h File Reference	15
4.1.1 Detailed Description	16
4.2 StatSpotify.h	16
<b>Index</b>	<b>17</b>



# Chapter 1

## Class Index

### 1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">SpotifyStats</a>	
Klasa zawierajaca typy parametrow: . . . . .	<a href="#">5</a>



## Chapter 2

# File Index

### 2.1 File List

Here is a list of all documented files with brief descriptions:

Projekt_Zaliczeniowy_Spotify/ <a href="#">StatSpotify.h</a>	
StatsSpotify.h . . . . .	15





## Chapter 3

# Class Documentation

### 3.1 SpotifyStats Class Reference

Klasa zawierająca typy parametrow:

```
#include <StatSpotify.h>
```

#### Public Member Functions

- bool [Readfile](#) (std::list< [SpotifyStats](#) > &spotifystatslist, std::string &fl)  
*Funkcja odczytująca z pliku bazy danych parametry piosenek.*
- void [SaveDataToFile](#) (std::list< [SpotifyStats](#) > &spotifystatslist, std::string &fl)  
*Funkcja zapisująca do pliku bazy danych.*
- void [GenerateReport](#) (std::list< [SpotifyStats](#) > &spotifystatslist, std::string &fl, std::string &reg)  
*Funkcja generująca raport ze statystykami piosenek.*
- std::vector< std::pair< std::string, int > > [MostPopularArtist](#) (std::list< [SpotifyStats](#) > &spotifystatslist)  
*Funkcja wyznaczająca najpopularniejszego artystę.*
- std::vector< std::pair< std::string, int > > [LeastPopularArtist](#) (std::list< [SpotifyStats](#) > &spotifystatslist)  
*Funkcja wyznaczająca najmniej popularnego artystę.*
- std::vector< [SpotifyStats](#) > [MostPopularSong](#) (std::list< [SpotifyStats](#) > &spotifystatslist)  
*Funkcja wyznaczająca najpopularniejszą piosenkę.*
- std::vector< [SpotifyStats](#) > [LeastPopularSong](#) (std::list< [SpotifyStats](#) > &spotifystatslist)  
*Funkcja wyznaczająca najmniej popularną piosenkę.*
- std::list< [SpotifyStats](#) > [ArtistSorter](#) (std::list< [SpotifyStats](#) > &spotifystatslist)  
*Funkcja sortująca Artystów względem odsłuchan ich piosenek.*
- int [ArithmeticMeanOfListens](#) (std::list< [SpotifyStats](#) > &spotifystatslist)  
*Funkcja wyznaczająca średnia arytmetyczna odsłuchan.*
- double [SpearmanKorelation](#) (std::list< [SpotifyStats](#) > &spotifystatslist)  
*Funkcja wyznaczająca współczynniki korelacji rang Spearmana między regionem a ilością odsłuchan.*

#### Private Attributes

- std::string **Author\_name**
- std::string **Author\_lastname**
- std::string **Title**
- std::string **Continent**
- int **playscount** = 0
- double **Songtime** = 0

## Friends

- void [RaportsGeneratorInterface](#) (std::list< [SpotifyStats](#) > &spotifystatslist)

*Funkcja interfejsu wyswietlajaca nam opcje generowania raportow.*

### 3.1.1 Detailed Description

Klasa zawierajaca typy parametrow:

#### Copyright

Copyrigh 2023 Piotr Kraska

#### Parameters

<i>Author_name</i>	Imie Autora.
<i>Author_lastname</i>	Nazwisko Autora. Author_name i Author_last name sa rowniez uzywane do zapisu pseudonimow artystycznych.
<i>Title</i>	Nazwa utworu
<i>Continent</i>	Region swiata.
<i>playscount</i>	Ilosc odsluchan
<i>Songtime</i>	Dlugosc piosenki

### 3.1.2 Member Function Documentation

#### 3.1.2.1 ArithmeticMeanOfListens()

```
int SpotifyStats::ArithmeticMeanOfListens (
    std::list< SpotifyStats > & spotifystatslist )
```

Funkcja wyznaczajaca srednia arytmetyczna odsluchan.

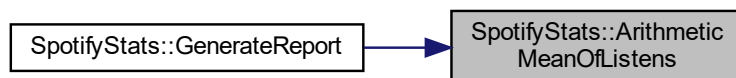
#### Parameters

<i>spotifystatslist</i>	Lista piosenek.
-------------------------	-----------------

**Returns**

Wartosc sredniej arytmetycznej.

Here is the caller graph for this function:

**3.1.2.2 ArtistSorter()**

```
std::list< SpotifyStats > SpotifyStats::ArtistSorter (  
    std::list< SpotifyStats > & spotifystatslist )
```

Funkcja sortująca Artystow względem odsłuchan ich piosenek.

**Parameters**

<i>spotifystatslist</i>	Lista piosenek.
-------------------------	-----------------

**Returns**

Posortowana liste piosenek.

Here is the caller graph for this function:

**3.1.2.3 GenerateReport()**

```
void SpotifyStats::GenerateReport (  
    std::list< SpotifyStats > & spotifystatslist,
```

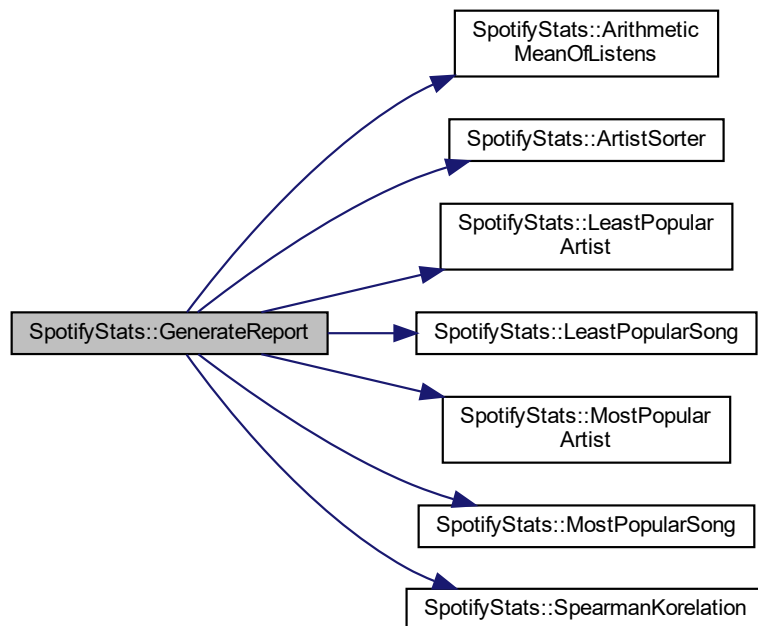
```
std::string & fl,  
std::string & reg )
```

Funkcja generujaca raport ze statystykami piosenek.

## Parameters

<i>spotifystatslist</i>	lista piosenek.
<i>fl</i>	Sciezka do pliku.
<i>reg</i>	Region z ktorego chcemy wygenerowac raport.

Here is the call graph for this function:



#### 3.1.2.4 LeastPopularArtist()

```
std::vector< std::pair< std::string, int > > SpotifyStats::LeastPopularArtist (
    std::list< SpotifyStats > & spotifystatslist )
```

Funkcja wyznaczająca najmniej popularnego artystę.

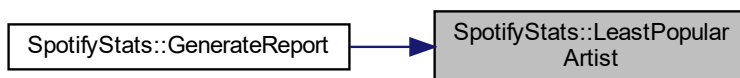
## Parameters

<i>spotifystatslist</i>	lista piosenek.
-------------------------	-----------------

## Returns

Wektor par imienia i nazwiska autora oraz sumy wyswietlen.

Here is the caller graph for this function:



### 3.1.2.5 LeastPopularSong()

```
std::vector< SpotifyStats > SpotifyStats::LeastPopularSong (
    std::list< SpotifyStats > & spotifystatslist )
```

Funkcja wyznaczająca najmniej popularną piosenkę.

#### Parameters

<i>spotifystatslist</i>	Lista piosenek.
-------------------------	-----------------

#### Returns

Wektor zawierający dane o Autorze najmniej popularnej piosenki.

Here is the caller graph for this function:



### 3.1.2.6 MostPopularArtist()

```
std::vector< std::pair< std::string, int > > SpotifyStats::MostPopularArtist (
    std::list< SpotifyStats > & spotifystatslist )
```

Funkcja wyznaczająca najpopularniejszego artystę.

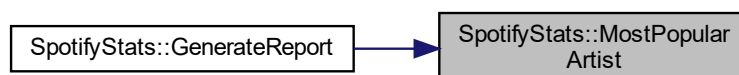
## Parameters

<i>spotifystatslist</i>	Lista piosenek.
-------------------------	-----------------

## Returns

Wektor par imienia i nazwiska autora oraz sumy wyswietlen.

Here is the caller graph for this function:



### 3.1.2.7 MostPopularSong()

```
std::vector< SpotifyStats > SpotifyStats::MostPopularSong (
    std::list< SpotifyStats > & spotifystatslist )
```

Funkcja wyznaczająca najpopularniejsza piosenke.

## Parameters

<i>spotifystatslist</i>	Lista piosenek.
-------------------------	-----------------

## Returns

Wektor zawierajacy dane o Autorze najpopularniejszej piosenki.

Here is the caller graph for this function:



### 3.1.2.8 Readfile()

```
bool SpotifyStats::Readfile (
    std::list< SpotifyStats > & spotifystatslist,
    std::string & fl )
```

Funkcja odczytująca z pliku bazy danych parametry piosenek.

#### Parameters

<i>spotifystatslist</i>	Lista z parametrami.
<i>fl</i>	Sciezka do pliku bazy danych.

#### Returns

True jesli wszystko zadziala poprawnie, false gdy cos pojdzie nie tak

### 3.1.2.9 SaveDataToFile()

```
void SpotifyStats::SaveDataToFile (
    std::list< SpotifyStats > & spotifystatslist,
    std::string & fl )
```

Funkcja zapisująca do pliku bazy danych.

#### Parameters

<i>spotifystatslist</i>	Lista z parametrami.
<i>fl</i>	Sciezka do pliku.

### 3.1.2.10 SpearmanKorelation()

```
double SpotifyStats::SpearmanKorelation (
    std::list< SpotifyStats > & spotifystatslist )
```

Funkcja wyznaczająca współczynniki korelacji rang Spearmana między regionem a ilością odsłuchan.

#### Parameters

<i>spotifystatslist</i>	Lista piosenek.
-------------------------	-----------------

#### Returns

Wartosc współczynnika korelacji.



Here is the caller graph for this function:



### 3.1.3 Friends And Related Function Documentation

#### 3.1.3.1 RaportsGeneratorInterface

```
void RaportsGeneratorInterface (
    std::list< SpotifyStats > & spotifystatslist ) [friend]
```

Funkcja interfejsu wyswietlajaca nam opcje generowania raportow.

##### Parameters

<i>spotifystatslist</i>	Lista piosenek.
-------------------------	-----------------

The documentation for this class was generated from the following files:

- Projekt\_Zaliczeniowy\_Spotify/[StatSpotify.h](#)
- Projekt\_Zaliczeniowy\_Spotify/StatSpotify.cpp



## Chapter 4

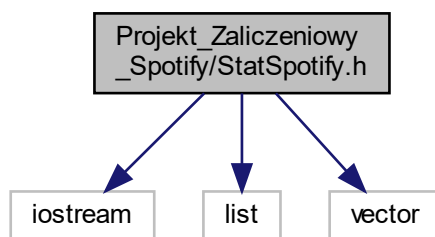
# File Documentation

### 4.1 Projekt\_Zaliczeniowy\_Spotify/StatSpotify.h File Reference

StatsSpotify.h.

```
#include <iostream>
#include <list>
#include <vector>
```

Include dependency graph for StatSpotify.h:



#### Classes

- class [SpotifyStats](#)

*Klasa zawierająca typy parametrow:*

#### Functions

- void **Help** ()

*Funkcja wyświetlająca nam pomoc.*

### 4.1.1 Detailed Description

StatsSpotify.h.

## 4.2 StatSpotify.h

[Go to the documentation of this file.](#)

```
1
2 #pragma once
3 #include <iostream>
4 #include <list>
5 #include <vector>
17 class SpotifyStats {
18     std::string Author_name;
19     std::string Author_lastname;
20     std::string Title;
21     std::string Continent;
22     int playscount = 0;
23     double Songtime = 0;
24
25 public:
30     bool Readfile(std::list<SpotifyStats> &spotifystatslist, std::string &fl);
34     void SaveDataToFile(std::list<SpotifyStats>& spotifystatslist, std::string &fl);
39     void GenerateReport(std::list<SpotifyStats>& spotifystatslist, std::string& fl, std::string& reg);
43     std::vector<std::pair<std::string, int> MostPopularArtist(std::list<SpotifyStats>& spotifystatslist);
47     std::vector<std::pair<std::string, int> LeastPopularArtist(std::list<SpotifyStats>&
spotifystatslist);
51     std::vector<SpotifyStats> MostPopularSong(std::list<SpotifyStats>& spotifystatslist);
55     std::vector<SpotifyStats> LeastPopularSong(std::list<SpotifyStats>& spotifystatslist);
59     std::list<SpotifyStats> ArtistSorter(std::list<SpotifyStats>& spotifystatslist);
63     int ArithmeticMeanOfListens(std::list<SpotifyStats>& spotifystatslist);
67     double SpearmanKorelation(std::list<SpotifyStats>& spotifystatslist);
70     friend void RaportsGeneratorInterface(std::list<SpotifyStats>& spotifystatslist);
71 };
73 void Help();
```

# Index

ArithmeticMeanOfListens

SpotifyStats, [6](#)

ArtistSorter

SpotifyStats, [7](#)

GenerateReport

SpotifyStats, [7](#)

LeastPopularArtist

SpotifyStats, [9](#)

LeastPopularSong

SpotifyStats, [10](#)

MostPopularArtist

SpotifyStats, [10](#)

MostPopularSong

SpotifyStats, [11](#)

Projekt\_Zaliczeniowy\_Spotify/StatSpotify.h, [15](#), [16](#)

RaportsGeneratorInterface

SpotifyStats, [13](#)

Readfile

SpotifyStats, [11](#)

SaveDataToFile

SpotifyStats, [12](#)

SpearmanKorelation

SpotifyStats, [12](#)

SpotifyStats, [5](#)

ArithmeticMeanOfListens, [6](#)

ArtistSorter, [7](#)

GenerateReport, [7](#)

LeastPopularArtist, [9](#)

LeastPopularSong, [10](#)

MostPopularArtist, [10](#)

MostPopularSong, [11](#)

RaportsGeneratorInterface, [13](#)

Readfile, [11](#)

SaveDataToFile, [12](#)

SpearmanKorelation, [12](#)