

System zarządzania biblioteką

Wygenerowano przez Doxygen 1.8.17

1 Indeks hierarchiczny	1
1.1 Hierarchia klas	1
2 Indeks klas	3
2.1 Lista klas	3
3 Indeks plików	5
3.1 Lista plików	5
4 Dokumentacja klas	7
4.1 Dokumentacja klasy Book	7
4.1.1 Dokumentacja konstruktora i destruktora	7
4.1.1.1 Book() [1/2]	8
4.1.1.2 Book() [2/2]	8
4.1.1.3 ~Book()	8
4.1.2 Dokumentacja funkcji składowych	8
4.1.2.1 saveItem()	8
4.1.2.2 showAvailable()	8
4.1.2.3 showItem()	9
4.1.2.4 search()	9
4.1.2.5 byType()	9
4.1.2.6 byGenre()	9
4.1.2.7 getType()	9
4.1.2.8 getAvailable()	10
4.2 Dokumentacja klasy Element	10
4.2.1 Opis szczegółowy	10
4.2.2 Dokumentacja konstruktora i destruktora	11
4.2.2.1 Element()	11
4.2.3 Dokumentacja funkcji składowych	11
4.2.3.1 showItem()	11
4.2.3.2 saveItem()	11
4.2.3.3 showAvailable()	11
4.2.3.4 byType()	12
4.2.3.5 byGenre()	12
4.2.3.6 search()	12
4.2.3.7 getType()	12
4.2.3.8 getRating()	12
4.2.3.9 getAvailable()	13
4.2.3.10 printByRate()	13
4.2.3.11 returnId()	13
4.2.3.12 changeAvailable()	13
4.2.4 Dokumentacja atrybutów składowych	13
4.2.4.1 ID	13

4.2.4.2 info	13
4.3 Dokumentacja klasy Film	14
4.3.1 Dokumentacja konstruktora i destruktora	14
4.3.1.1 Film() [1/2]	14
4.3.1.2 Film() [2/2]	14
4.3.1.3 ~Film()	15
4.3.2 Dokumentacja funkcji składowych	15
4.3.2.1 showItem()	15
4.3.2.2 saveItem()	15
4.3.2.3 showAvailable()	15
4.3.2.4 search()	15
4.3.2.5 byType()	16
4.3.2.6 getRating()	16
4.3.2.7 getType()	16
4.3.2.8 printByRate()	16
4.3.2.9 getAvailable()	16
4.4 Dokumentacja klasy Music	17
4.4.1 Dokumentacja konstruktora i destruktora	17
4.4.1.1 Music() [1/2]	17
4.4.1.2 Music() [2/2]	17
4.4.1.3 ~Music()	18
4.4.2 Dokumentacja funkcji składowych	18
4.4.2.1 showItem()	18
4.4.2.2 saveItem()	18
4.4.2.3 showAvailable()	18
4.4.2.4 search()	18
4.4.2.5 byType()	19
4.4.2.6 getType()	19
4.4.2.7 getAvailable()	19
4.5 Dokumentacja klasy Person	19
4.5.1 Dokumentacja konstruktora i destruktora	20
4.5.1.1 Person() [1/3]	20
4.5.1.2 Person() [2/3]	20
4.5.1.3 Person() [3/3]	20
4.5.1.4 ~Person()	20
4.5.2 Dokumentacja funkcji składowych	20
4.5.2.1 saveUser()	20
4.5.2.2 borrow()	21
4.5.2.3 showUserElement()	21
4.5.3 Dokumentacja atrybutów składowych	21
4.5.3.1 id	21
4.5.3.2 first_name	21

4.5.3.3 last_name	21
4.5.3.4 borrowed	21
5 Dokumentacja plików	23
5.1 Dokumentacja pliku Biblioteka/Book.h	23
5.2 Dokumentacja pliku Biblioteka/Element.h	23
5.2.1 Dokumentacja typów wyliczanych	23
5.2.1.1 Type	23
5.3 Dokumentacja pliku Biblioteka/Film.h	24
5.4 Dokumentacja pliku Biblioteka/functions.h	24
5.4.1 Dokumentacja funkcji	25
5.4.1.1 return_index_user()	25
5.4.1.2 return_index_ele()	25
5.4.1.3 type_to_string()	25
5.4.1.4 load_items()	25
5.4.1.5 save_items()	26
5.4.1.6 load_users()	26
5.4.1.7 save_users()	26
5.4.1.8 add()	26
5.4.1.9 show_user()	26
5.4.1.10 show_users()	27
5.4.1.11 show()	27
5.4.1.12 person_id()	27
5.4.1.13 borrow_item()	27
5.4.1.14 borrowed_repeat()	27
5.4.1.15 return_items()	28
5.4.1.16 delete_()	28
5.4.1.17 sort_by_rate()	28
5.5 Dokumentacja pliku Biblioteka/Music.h	28
5.6 Dokumentacja pliku Biblioteka/Person.h	28
Indeks	29

Rozdział 1

Indeks hierarchiczny

1.1 Hierarchia klas

Ta lista dziedziczenia posortowana jest z grubsza, choć nie całkowicie, alfabetycznie:

Element	10
Book	7
Film	14
Music	17
Person	19

Rozdział 2

Indeks klas

2.1 Lista klas

Tutaj znajdują się klasy, struktury, unie i interfejsy wraz z ich krótkimi opisami:

Book	7
Element	10
Film	14
Music	17
Person	19

Rozdział 3

Indeks plików

3.1 Lista plików

Tutaj znajduje się lista wszystkich plików z ich krótkimi opisami:

Biblioteka/ Book.h	23
Biblioteka/ Element.h	23
Biblioteka/ Film.h	24
Biblioteka/ functions.h	24
Biblioteka/ Music.h	28
Biblioteka/ Person.h	28

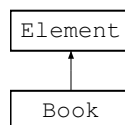
Rozdział 4

Dokumentacja klas

4.1 Dokumentacja klasy Book

```
#include <Book.h>
```

Diagram dziedziczenia dla Book



Metody publiczne

- `Book` (int, `Type`, string, string, string)
- `Book` (int index)
- `~Book` ()
- virtual string `saveItem` ()
- virtual void `showAvailable` ()
- virtual void `showItem` ()
- virtual void `search` (string)
- virtual void `byType` (`Type`)
- virtual void `byGenre` (string)
- virtual `Type` `getType` ()
- virtual string `getAvailable` ()

Dodatkowe Dziedziczone Składowe

4.1.1 Dokumentacja konstruktora i destruktor

4.1.1.1 Book() [1/2]

```
Book::Book (
    int ,
    Type ,
    string ,
    string ,
    string ,
    string )
```

4.1.1.2 Book() [2/2]

```
Book::Book (
    int index )
```

4.1.1.3 ~Book()

```
Book::~Book ( )
```

4.1.2 Dokumentacja funkcji składowych

4.1.2.1 saveItem()

```
virtual string Book::saveItem ( ) [virtual]
```

sdUzywana do zapisywania danych w pliku elementow.

Implementuje [Element](#).

4.1.2.2 showAvailable()

```
virtual void Book::showAvailable ( ) [virtual]
```

Metoda virtualna wyswietlajca element jesli jest on dostepny do wypozyczenia.

Implementuje [Element](#).

4.1.2.3 showItem()

```
virtual void Book::showItem ( ) [virtual]
```

Metoda virtualna wyswietlajca poszczegolne elementy.

Implementuje [Element](#).

4.1.2.4 search()

```
virtual void Book::search (
    string ) [virtual]
```

Metoda virtualna przerzukujaca elementy pod katem szukanej frazy.

Reimplementowana z [Element](#).

4.1.2.5 byType()

```
virtual void Book::byType (
    Type ) [virtual]
```

Metoda virtualna wyswietlajca poszczegolne elementy jesli sa one zgodne z podanym typem.

Reimplementowana z [Element](#).

4.1.2.6 byGenre()

```
virtual void Book::byGenre (
    string ) [virtual]
```

Metoda virtualna potrzebna do wyswietlania ksiazek na podstawie gatunkow.

Reimplementowana z [Element](#).

4.1.2.7 getType()

```
virtual Type Book::getType ( ) [virtual]
```

Metoda virtualna zwracajaca typ elemntu.

Implementuje [Element](#).

4.1.2.8 getAvailable()

```
virtual string Book::getAvailable ( ) [virtual]
```

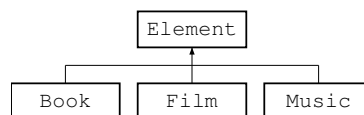
Metoda virtualna potrzebna do uzyskania informacji o dostępności elementu.

Implementuje [Element](#).

4.2 Dokumentacja klasy Element

```
#include <Element.h>
```

Diagram dziedziczenia dla Element



Metody publiczne

- [Element](#) (int i, string info)
- virtual void [showItem](#) ()=0
- virtual string [saveItem](#) ()=0
- virtual void [showAvailable](#) ()=0
- virtual void [byType](#) (Type)
- virtual void [byGenre](#) (string)
- virtual void [search](#) (string)
- virtual Type [getType](#) ()=0
- virtual float [getRating](#) ()
- virtual string [getAvailable](#) ()=0
- virtual void [printByRate](#) (float)
- int [returnId](#) ()
- void [changeAvailable](#) ()

Atrybuty chronione

- int [ID](#)
- string [info](#)

4.2.1 Opis szczegółowy

Klasa [Element](#) jest klasą bazową abstrakcyjną wszystkich przedmiotów możliwych do wypożyczenia.

Parametry

<i>index</i>	Jest to numer ID elementu.
<i>info</i>	Zawiera informacje o tym czy dany element jest dostępny do wypożyczenia.

4.2.2 Dokumentacja konstruktora i destruktora

4.2.2.1 Element()

```
Element::Element (
    int i,
    string info )
```

Konstruktor potrzebny do tworzenia obiektów klas pochodnych.

Parametry

<i>i</i>	ID elementu.
<i>in</i>	Informacja o dostępności.

4.2.3 Dokumentacja funkcji składowych

4.2.3.1 showItem()

```
virtual void Element::showItem ( ) [pure virtual]
```

Metoda virtualna wyświetlająca poszczególne elementy.

Implementowany w [Book](#), [Film](#) i [Music](#).

4.2.3.2 saveItem()

```
virtual string Element::saveItem ( ) [pure virtual]
```

sdUżywana do zapisywania danych w pliku elementów.

Implementowany w [Book](#), [Film](#) i [Music](#).

4.2.3.3 showAvailable()

```
virtual void Element::showAvailable ( ) [pure virtual]
```

Metoda virtualna wyświetlająca element jeśli jest on dostępny do wypożyczenia.

Implementowany w [Book](#), [Film](#) i [Music](#).

4.2.3.4 byType()

```
virtual void Element::byType (
    Type ) [virtual]
```

Metoda virtualna wyswietlajca poszczególne elementy jesli sa one zgodne z podanym typem.

Reimplementowana w [Book](#), [Film](#) i [Music](#).

4.2.3.5 byGenre()

```
virtual void Element::byGenre (
    string ) [virtual]
```

Metoda virtualna potrzebna do wyswietlania ksiazek na podstawie gatunkow.

Reimplementowana w [Book](#).

4.2.3.6 search()

```
virtual void Element::search (
    string ) [virtual]
```

Metoda virtualna przeszukujaca elementy pod katem szukanej frazy.

Reimplementowana w [Book](#), [Film](#) i [Music](#).

4.2.3.7 getType()

```
virtual Type Element::getType ( ) [pure virtual]
```

Metoda virtualna zwracajaca typ elemntu.

Implementowany w [Book](#), [Film](#) i [Music](#).

4.2.3.8 getRaiting()

```
virtual float Element::getRaiting ( ) [virtual]
```

Metoda virtualna potrzebna do wyswietlania posortowanych filmow na podstawie ich ocen.

Reimplementowana w [Film](#).

4.2.3.9 getAvailable()

```
virtual string Element::getAvailable ( ) [pure virtual]
```

Metoda virtualna potrzebna do uzyskania informacji o dostępności elementu.

Implementowany w [Book](#), [Film](#) i [Music](#).

4.2.3.10 printByRate()

```
virtual void Element::printByRate (
    float ) [virtual]
```

Metoda virtualna potrzebna do wyświetlania książek na podstawie gatunków.

Reimplementowana w [Film](#).

4.2.3.11 returnId()

```
int Element::returnId ( )
```

Metoda potrzebna do uzyskania dostępu do ID elementu.

4.2.3.12 changeAvailable()

```
void Element::changeAvailable ( )
```

Metoda zmieniająca dostępność elementów.

4.2.4 Dokumentacja atrybutów składowych

4.2.4.1 ID

```
int Element::ID [protected]
```

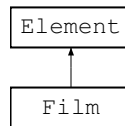
4.2.4.2 info

```
string Element::info [protected]
```

4.3 Dokumentacja klasy Film

```
#include <Film.h>
```

Diagram dziedziczenia dla Film



Metody publiczne

- [Film](#) (int, [Type](#), string, string, float, string)
- [Film](#) (int index)
- [~Film](#) ()
- virtual void [showItem](#) ()
- virtual string [saveItem](#) ()
- virtual void [showAvailable](#) ()
- virtual void [search](#) (string)
- virtual void [byType](#) ([Type](#))
- virtual float [getRaiting](#) ()
- virtual [Type](#) [getType](#) ()
- virtual void [printByRate](#) (float)
- virtual string [getAvailable](#) ()

Dodatkowe Dziedziczone Składowe

4.3.1 Dokumentacja konstruktora i destruktora

4.3.1.1 Film() [1/2]

```
Film::Film (  
    int ,  
    Type ,  
    string ,  
    string ,  
    float ,  
    string )
```

4.3.1.2 Film() [2/2]

```
Film::Film (  
    int index )
```

4.3.1.3 ~Film()

```
Film::~~Film ( )
```

4.3.2 Dokumentacja funkcji składowych

4.3.2.1 showItem()

```
virtual void Film::showItem ( ) [virtual]
```

Metoda virtualna wyświetlająca poszczególne elementy.

Implementuje [Element](#).

4.3.2.2 saveItem()

```
virtual string Film::saveItem ( ) [virtual]
```

sdUżywana do zapisywania danych w pliku elementow.

Implementuje [Element](#).

4.3.2.3 showAvailable()

```
virtual void Film::showAvailable ( ) [virtual]
```

Metoda virtualna wyświetlająca element jeśli jest on dostępny do wypożyczenia.

Implementuje [Element](#).

4.3.2.4 search()

```
virtual void Film::search (
    string ) [virtual]
```

Metoda virtualna przeszukująca elementy pod kątem szukanej frazy.

Reimplementowana z [Element](#).

4.3.2.5 byType()

```
virtual void Film::byType (
    Type ) [virtual]
```

Metoda virtualna wyswietlajca poszczegolne elementy jesli sa one zgodne z podanym typem.

Reimplementowana z [Element](#).

4.3.2.6 getRaiting()

```
virtual float Film::getRaiting ( ) [virtual]
```

Metoda virtualna potrzebna do wyswietlania posortowanych filmow na podstawie ich ocen.

Reimplementowana z [Element](#).

4.3.2.7 getType()

```
virtual Type Film::getType ( ) [virtual]
```

Metoda virtualna zwracajaca typ elemntu.

Implementuje [Element](#).

4.3.2.8 printByRate()

```
virtual void Film::printByRate (
    float ) [virtual]
```

Metoda virtualna potrzebna do wyswietlania ksiazek na podstawie gatunkow.

Reimplementowana z [Element](#).

4.3.2.9 getAvailable()

```
virtual string Film::getAvailable ( ) [virtual]
```

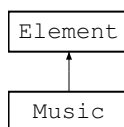
Metoda virtualna potrzebna do uzyskawiania informacji o dostepnosci elementu.

Implementuje [Element](#).

4.4 Dokumentacja klasy Music

```
#include <Music.h>
```

Diagram dziedziczenia dla Music



Metody publiczne

- `Music` (int, `Type`, string, string, string, string)
- `Music` (int index)
- `~Music` ()
- virtual void `showItem` ()
- virtual string `savelItem` ()
- virtual void `showAvailable` ()
- virtual void `search` (string)
- virtual void `byType` (`Type`)
- virtual `Type` `getType` ()
- virtual string `getAvailable` ()

Dodatkowe Dziedziczone Składowe

4.4.1 Dokumentacja konstruktora i destruktor

4.4.1.1 `Music()` [1/2]

```
Music::Music (  
    int ,  
    Type ,  
    string ,  
    string ,  
    string ,  
    string )
```

4.4.1.2 `Music()` [2/2]

```
Music::Music (  
    int index )
```

4.4.1.3 ~Music()

```
Music::~~Music ( )
```

4.4.2 Dokumentacja funkcji składowych

4.4.2.1 showItem()

```
virtual void Music::showItem ( ) [virtual]
```

Metoda virtualna wyswietlajca poszczególne elementy.

Implementuje [Element](#).

4.4.2.2 saveItem()

```
virtual string Music::saveItem ( ) [virtual]
```

sdUzywana do zapisywania danych w pliku elementow.

Implementuje [Element](#).

4.4.2.3 showAvailable()

```
virtual void Music::showAvailable ( ) [virtual]
```

Metoda virtualna wyswietlajca element jesli jest on dostepny do wypozyczenia.

Implementuje [Element](#).

4.4.2.4 search()

```
virtual void Music::search (
    string ) [virtual]
```

Metoda virtualna przerzukujaca elementy pod katem szukanej frazy.

Reimplementowana z [Element](#).

4.4.2.5 byType()

```
virtual void Music::byType (
    Type ) [virtual]
```

Metoda virtualna wyswietlajca poszczególne elementy jesli sa one zgodne z podanym typem.

Reimplementowana z [Element](#).

4.4.2.6 getType()

```
virtual Type Music::getType ( ) [virtual]
```

Metoda virtualna zwracajaca typ elemntu.

Implementuje [Element](#).

4.4.2.7 getAvailable()

```
virtual string Music::getAvailable ( ) [virtual]
```

Metoda virtualna potrzebna do uzyskawinia informacji o dostepnosci elementu.

Implementuje [Element](#).

4.5 Dokumentacja klasy Person

```
#include <Person.h>
```

Metody publiczne

- [Person](#) (int, string, string, std::vector< int >)
- [Person](#) (int, string, string)
- [Person](#) (int id)
- string [saveUser](#) ()
- void [borrow](#) (int)
- void [showUserElement](#) (const std::vector< [Element](#) * > element, bool &check)
- [~Person](#) ()

Atrybuty publiczne

- int [id](#)
- string [first_name](#)
- string [last_name](#)
- std::vector< int > [borrowed](#)

4.5.1 Dokumentacja konstruktora i destruktora

4.5.1.1 Person() [1/3]

```
Person::Person (
    int ,
    string ,
    string ,
    std::vector< int > )
```

4.5.1.2 Person() [2/3]

```
Person::Person (
    int ,
    string ,
    string )
```

4.5.1.3 Person() [3/3]

```
Person::Person (
    int id )
```

4.5.1.4 ~Person()

```
Person::~~Person ( )
```

4.5.2 Dokumentacja funkcji składowych

4.5.2.1 saveUser()

```
string Person::saveUser ( )
```

4.5.2.2 borrow()

```
void Person::borrow (
    int )
```

4.5.2.3 showUserElement()

```
void Person::showUserElement (
    const std::vector< Element * > element,
    bool & check )
```

4.5.3 Dokumentacja atrybutów składowych

4.5.3.1 id

```
int Person::id
```

4.5.3.2 first_name

```
string Person::first_name
```

4.5.3.3 last_name

```
string Person::last_name
```

4.5.3.4 borrowed

```
std::vector<int> Person::borrowed
```

Rozdział 5

Dokumentacja plików

5.1 Dokumentacja pliku Biblioteka/Book.h

```
#include <string>
#include "Element.h"
```

Komponenty

- class [Book](#)

5.2 Dokumentacja pliku Biblioteka/Element.h

```
#include <string>
```

Komponenty

- class [Element](#)

Wyliczenia

- enum [Type](#) { [BOOK](#), [FILM](#), [MUSIC](#) }

5.2.1 Dokumentacja typów wyliczanych

5.2.1.1 Type

```
enum Type
```

Wartości wyliczeń

BOOK	
FILM	
MUSIC	

5.3 Dokumentacja pliku Biblioteka/Film.h

```
#include <string>
#include "Element.h"
```

Komponenty

- class [Film](#)

5.4 Dokumentacja pliku Biblioteka/functions.h

```
#include <vector>
#include <fstream>
#include <sstream>
#include <iostream>
#include <algorithm>
#include <iomanip>
#include "Element.h"
#include "Person.h"
#include "Book.h"
#include "Film.h"
#include "Music.h"
```

Funkcje

- int [return_index_user](#) (int ID, std::vector< [Person](#) > person)
- int [return_index_ele](#) (int ID, std::vector< [Element](#) * > person)
- std::string [type_to_string](#) (const [Type](#) type)
- void [load_items](#) (const std::string name_file, std::vector< [Element](#) * > &element)
- void [save_items](#) (const std::string name_file, const std::vector< [Element](#) * > element) noexcept
- void [load_users](#) (const std::string file_name, std::vector< [Person](#) > &person)
- void [save_users](#) (const std::string name_file, const std::vector< [Person](#) > person) noexcept
- void [add](#) (std::vector< [Person](#) > &person, std::vector< [Element](#) * > &element, const std::string file_↵ elements, const std::string file_users)
- void [show_user](#) (const int ID, const std::vector< [Person](#) > person) noexcept
- void [show_users](#) (const std::vector< [Person](#) > person) noexcept
- void [show](#) (std::vector< [Person](#) > person, const std::vector< [Element](#) * > element)
- int [person_id](#) (const std::vector< [Person](#) > person)
- void [borrow_item](#) (std::vector< [Person](#) > &person, std::vector< [Element](#) * > &element, const std::string file_elements, const std::string file_users)

- void `borrowed_repeat` (std::vector< `Element` * > &, std::vector< `Person` > &, int ID_person, int &element↵_index)
- void `return_items` (std::vector< `Person` > &, std::vector< `Element` * > &, std::string file_elements, std::string file_users)
- void `delete_` (std::vector< `Person` > &, std::vector< `Element` * > &, const std::string file_elements, const std::string file_users)
- void `sort_by_rate` (const std::vector< `Element` * > element) noexcept

5.4.1 Dokumentacja funkcji

5.4.1.1 return_index_user()

```
int return_index_user (
    int ID,
    std::vector< Person > person )
```

Funkcja potrzebna do uzyskania indexu wektora danego uzytkownika na podstawie ID.

5.4.1.2 return_index_ele()

```
int return_index_ele (
    int ID,
    std::vector< Element * > person )
```

Funkcja potrzebna do uzyskania indexu wektora danego przedmiotu na podstawie ID.

5.4.1.3 type_to_string()

```
std::string type_to_string (
    const Type type )
```

Funkcja zamieniajaca typ wyliczeniowy na string.

5.4.1.4 load_items()

```
void load_items (
    const std::string name_file,
    std::vector< Element * > & element )
```

Funkcja pobierajaca dane z pliku elementow wprowadzajaca je do programu (wektora).

Parametry

<i>name_file</i>	Nazwa pliku zawierajace dane elementow.
<i>element</i>	Vector elementow do przechowywania danych.

5.4.1.5 save_items()

```
void save_items (
    const std::string name_file,
    const std::vector< Element * > element ) [noexcept]
```

Funkcja zapisujące dane przechowywane w vectorze do pliku elementow.

Parametry

<i>name_file</i>	Nazwa pliku zawierające dane elementow.
<i>element</i>	Vector elementow do przechowywania danych.

5.4.1.6 load_users()

```
void load_users (
    const std::string file_name,
    std::vector< Person > & person )
```

Funkcja pobierająca dane z pliku osob wprowadzająca je do programu (vectora).

5.4.1.7 save_users()

```
void save_users (
    const std::string name_file,
    const std::vector< Person > person ) [noexcept]
```

Funkcja zapisujące dane przechowywane w vectorze do pliku osob.

5.4.1.8 add()

```
void add (
    std::vector< Person > & person,
    std::vector< Element * > & element,
    const std::string file_elements,
    const std::string file_users )
```

Funkcja dodająca element lub osobe do bazy elementow lub osob oraz zpisująca je w oryginalnym pliku.

5.4.1.9 show_user()

```
void show_user (
    const int ID,
    const std::vector< Person > person ) [noexcept]
```

Funkcja wyświetlająca imie mi nazwisko danej osoby na podstawie id.

5.4.1.10 show_users()

```
void show_users (
    const std::vector< Person > person ) [noexcept]
```

Funkcja wyświetlająca ID, Imię oraz nazwisko wszystkich osób.

5.4.1.11 show()

```
void show (
    std::vector< Person > person,
    const std::vector< Element * > element )
```

Funkcja wyświetlająca użytkowników lub liste elementów w zależności od wyboru. Pozwala na wyświetlenie wszystkich elementów jak i również wybranych. Można wyświetlić płyty, książki oraz filmy jak i konkretne gatunki książek lub posortowane filmy na podstawie ich ocen.

5.4.1.12 person_id()

```
int person_id (
    const std::vector< Person > person )
```

Funkcja sprawdza imię oraz nazwisko użytkownika zwracając jego ID.

5.4.1.13 borrow_item()

```
void borrow_item (
    std::vector< Person > & person,
    std::vector< Element * > & element,
    const std::string file_elements,
    const std::string file_users )
```

Funkcja umożliwia wypożyczenie użytkownikowi wybranego przedmiotu.

5.4.1.14 borrowed_repeat()

```
void borrowed_repeat (
    std::vector< Element * > & ,
    std::vector< Person > & ,
    int ID_person,
    int & element_index )
```

Funkcja która jest używana w funkcji borrow, odpowiada za wypożyczenie przedmiotu bez konieczności pisanie jej dwa razy w funkcji borrow.

5.4.1.15 return_items()

```
void return_items (
    std::vector< Person > & ,
    std::vector< Element * > & ,
    std::string file_elements,
    std::string file_users )
```

Funkcja pozwalająca na zwracanie przez użytkownika wypożyczonych przedmiotów.

5.4.1.16 delete_()

```
void delete_ (
    std::vector< Person > & ,
    std::vector< Element * > & ,
    const std::string file_elements,
    const std::string file_users )
```

Funkcja pozwalająca na usunięcie elementów lub użytkowników z plików danych oraz wektorów, bez zmieniania jego ID.

5.4.1.17 sort_by_rate()

```
void sort_by_rate (
    const std::vector< Element * > element ) [noexcept]
```

Funkcja sortująca filmy na podstawie ocen.

5.5 Dokumentacja pliku Biblioteka/Music.h

```
#include "Element.h"
#include <string>
```

Komponenty

- class [Music](#)

5.6 Dokumentacja pliku Biblioteka/Person.h

```
#include <string>
#include <vector>
#include "Element.h"
```

Komponenty

- class [Person](#)