

Library

Generated by Doxygen 1.9.3



<b>1 Hierarchical Index</b>	<b>1</b>
1.1 Class Hierarchy	1
<b>2 Class Index</b>	<b>3</b>
2.1 Class List	3
<b>3 File Index</b>	<b>5</b>
3.1 File List	5
<b>4 Class Documentation</b>	<b>7</b>
4.1 Book Class Reference	7
4.1.1 Detailed Description	7
4.1.2 Member Function Documentation	8
4.1.2.1 display()	8
4.1.2.2 fromCsv()	8
4.1.2.3 getId()	8
4.1.2.4 getStatus()	8
4.1.2.5 rent()	9
4.1.2.6 requestData()	9
4.1.2.7 returnBack()	9
4.1.2.8 search()	9
4.1.2.9 toCsv()	9
4.2 Books Class Reference	10
4.2.1 Detailed Description	11
4.2.2 Constructor & Destructor Documentation	11
4.2.2.1 Books()	11
4.2.3 Member Function Documentation	11
4.2.3.1 add()	11
4.2.3.2 deserialize()	11
4.2.3.3 display()	11
4.2.3.4 exists()	11
4.2.3.5 get()	12
4.2.3.6 isAvailable()	12
4.2.3.7 remove()	12
4.2.3.8 rent()	13
4.2.3.9 returnBack()	13
4.2.3.10 search()	13
4.2.3.11 serialize()	13
4.3 Rental Class Reference	14
4.3.1 Detailed Description	14
4.3.2 Member Function Documentation	14
4.3.2.1 display()	14
4.3.2.2 fromCsv()	14

4.3.2.3	<a href="#">getBookId()</a>	15
4.3.2.4	<a href="#">getId()</a>	15
4.3.2.5	<a href="#">getStudentId()</a>	15
4.3.2.6	<a href="#">requestData()</a>	15
4.3.2.7	<a href="#">search()</a>	15
4.3.2.8	<a href="#">setId()</a>	16
4.3.2.9	<a href="#">toCsv()</a>	16
4.4	<a href="#">Rentals Class Reference</a>	16
4.4.1	<a href="#">Detailed Description</a>	17
4.4.2	<a href="#">Constructor &amp; Destructor Documentation</a>	17
4.4.2.1	<a href="#">Rentals()</a>	17
4.4.3	<a href="#">Member Function Documentation</a>	18
4.4.3.1	<a href="#">add()</a>	18
4.4.3.2	<a href="#">deserialize()</a>	18
4.4.3.3	<a href="#">display()</a>	18
4.4.3.4	<a href="#">get()</a>	18
4.4.3.5	<a href="#">remove()</a>	19
4.4.3.6	<a href="#">search()</a>	19
4.4.3.7	<a href="#">serialize()</a>	19
4.5	<a href="#">Serializer Class Reference</a>	19
4.5.1	<a href="#">Detailed Description</a>	20
4.5.2	<a href="#">Constructor &amp; Destructor Documentation</a>	20
4.5.2.1	<a href="#">Serializer()</a>	20
4.5.3	<a href="#">Member Function Documentation</a>	20
4.5.3.1	<a href="#">deserialize()</a>	20
4.5.3.2	<a href="#">getDelimiter()</a>	20
4.5.3.3	<a href="#">serialize()</a>	21
4.5.4	<a href="#">Member Data Documentation</a>	21
4.5.4.1	<a href="#">delimiter</a>	21
4.5.4.2	<a href="#">fileName</a>	21
4.6	<a href="#">Student Class Reference</a>	21
4.6.1	<a href="#">Detailed Description</a>	21
4.6.2	<a href="#">Member Function Documentation</a>	22
4.6.2.1	<a href="#">display()</a>	22
4.6.2.2	<a href="#">fromCsv()</a>	22
4.6.2.3	<a href="#">getId()</a>	22
4.6.2.4	<a href="#">requestData()</a>	22
4.6.2.5	<a href="#">search()</a>	22
4.6.2.6	<a href="#">toCsv()</a>	23
4.7	<a href="#">Students Class Reference</a>	23
4.7.1	<a href="#">Detailed Description</a>	24
4.7.2	<a href="#">Constructor &amp; Destructor Documentation</a>	24

4.7.2.1 Students()	24
4.7.3 Member Function Documentation	24
4.7.3.1 add()	24
4.7.3.2 deserialize()	25
4.7.3.3 display()	25
4.7.3.4 exists()	25
4.7.3.5 get()	25
4.7.3.6 remove()	26
4.7.3.7 search()	26
4.7.3.8 serialize()	26
<b>5 File Documentation</b>	<b>27</b>
5.1 C:/Users/Kuba/Tajne/Book.cpp File Reference	27
5.2 C:/Users/Kuba/Tajne/Book.h File Reference	27
5.3 Book.h	27
5.4 C:/Users/Kuba/Tajne/Books.cpp File Reference	28
5.5 C:/Users/Kuba/Tajne/Books.h File Reference	28
5.6 Books.h	28
5.7 C:/Users/Kuba/Tajne/main.cpp File Reference	29
5.7.1 Function Documentation	29
5.7.1.1 deserialize()	29
5.7.1.2 displayMenu()	30
5.7.1.3 main()	30
5.7.1.4 operationConfirmation()	30
5.8 C:/Users/Kuba/Tajne/Rental.cpp File Reference	30
5.8.1 Function Documentation	30
5.8.1.1 currentDateTime()	30
5.9 C:/Users/Kuba/Tajne/Rental.h File Reference	31
5.10 Rental.h	31
5.11 C:/Users/Kuba/Tajne/Rentals.cpp File Reference	31
5.12 C:/Users/Kuba/Tajne/Rentals.h File Reference	31
5.13 Rentals.h	32
5.14 C:/Users/Kuba/Tajne/Serializer.h File Reference	32
5.15 Serializer.h	33
5.16 C:/Users/Kuba/Tajne/Student.cpp File Reference	33
5.17 C:/Users/Kuba/Tajne/Student.h File Reference	33
5.18 Student.h	34
5.19 C:/Users/Kuba/Tajne/Students.cpp File Reference	34
5.20 C:/Users/Kuba/Tajne/Students.h File Reference	34
5.21 Students.h	35
<b>Index</b>	<b>37</b>



# Chapter 1

## Hierarchical Index

### 1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Book . . . . .	7
Rental . . . . .	14
Serializer . . . . .	19
Books . . . . .	10
Rentals . . . . .	16
Students . . . . .	23
Student . . . . .	21





## Chapter 2

# Class Index

### 2.1 Class List

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

<a href="#">Book</a>	Klasa ksiazka . . . . .	<a href="#">7</a>
<a href="#">Books</a>	Klasa do obslugi wielu ksiazek . . . . .	<a href="#">10</a>
<a href="#">Rental</a>	Klasa wypozyczenie . . . . .	<a href="#">14</a>
<a href="#">Rentals</a>	Klasa do obsugi wypozyczen . . . . .	<a href="#">16</a>
<a href="#">Serializer</a>	Klasa bazowa zapewniajaca mozliwosc zapisu i odczytu z pliku w formacie csv . . . . .	<a href="#">19</a>
<a href="#">Student</a>	Klasa uczen . . . . .	<a href="#">21</a>
<a href="#">Students</a>	Klasa do obslugi wielu uczniow . . . . .	<a href="#">23</a>



## Chapter 3

# File Index

### 3.1 File List

Here is a list of all files with brief descriptions:

C:/Users/Kuba/Tajne/ <a href="#">Book.cpp</a> . . . . .	27
C:/Users/Kuba/Tajne/ <a href="#">Book.h</a> . . . . .	27
C:/Users/Kuba/Tajne/ <a href="#">Books.cpp</a> . . . . .	28
C:/Users/Kuba/Tajne/ <a href="#">Books.h</a> . . . . .	28
C:/Users/Kuba/Tajne/ <a href="#">main.cpp</a> . . . . .	29
C:/Users/Kuba/Tajne/ <a href="#">Rental.cpp</a> . . . . .	30
C:/Users/Kuba/Tajne/ <a href="#">Rental.h</a> . . . . .	31
C:/Users/Kuba/Tajne/ <a href="#">Rentals.cpp</a> . . . . .	31
C:/Users/Kuba/Tajne/ <a href="#">Rentals.h</a> . . . . .	31
C:/Users/Kuba/Tajne/ <a href="#">Serializer.h</a> . . . . .	32
C:/Users/Kuba/Tajne/ <a href="#">Student.cpp</a> . . . . .	33
C:/Users/Kuba/Tajne/ <a href="#">Student.h</a> . . . . .	33
C:/Users/Kuba/Tajne/ <a href="#">Students.cpp</a> . . . . .	34
C:/Users/Kuba/Tajne/ <a href="#">Students.h</a> . . . . .	34



## Chapter 4

# Class Documentation

### 4.1 Book Class Reference

klasa `ksiazka`

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

#### Public Member Functions

- void `requestData` ()  
*funkcja wczytuje dane książki z klawiatury*
- int `getId` ()  
*funkcja podaje id książki*
- int `getStatus` ()  
*funkcja podaje status książki*
- void `rent` ()  
*funkcja obsługuje wypożyczenie książki*
- void `returnBack` ()  
*funkcja obsługuje zwrot książki*
- bool `search` (string text)  
*funkcja sprawdza czy tytuł lub autor książki pasuje do podanej frazy wyszukiwania*
- void `display` ()  
*funkcja wyświetla informacje o książce*
- string `toCsv` (string delimiter)  
*funkcja zamienia dane książki na napis w formacie csv*
- void `fromCsv` (string text, string delimiter)  
*funkcja odczytuje z napisu w formacie csv dane książki*

#### 4.1.1 Detailed Description

klasa `ksiazka`

## 4.1.2 Member Function Documentation

### 4.1.2.1 display()

```
void Book::display ( )
```

funkcja wyswietla informacje o ksiazce

### 4.1.2.2 fromCsv()

```
void Book::fromCsv (
    string text,
    string delimiter )
```

funkcja odczytuje z napisu w formacie csv dane ksiazki

#### Parameters

<i>text</i>	dane do odczytu
<i>delimiter</i>	odstep pomiedzy danymi

### 4.1.2.3 getId()

```
int Book::getId ( )
```

funkcja podaje id ksiazki

#### Returns

identyfikator ksiazki

### 4.1.2.4 getStatus()

```
int Book::getStatus ( )
```

funkcja podaje status ksiazki

#### Returns

status ksiazki

#### 4.1.2.5 rent()

```
void Book::rent ( )
```

funkcja obsługuje wypożyczenie książki

#### 4.1.2.6 requestData()

```
void Book::requestData ( )
```

funkcja wczytuje dane książki z klawiatury

#### 4.1.2.7 returnBack()

```
void Book::returnBack ( )
```

funkcja obsługuje zwrot książki

#### 4.1.2.8 search()

```
bool Book::search (
    string text )
```

funkcja sprawdza czy tytuł lub autor książki pasuje do podanej frazy wyszukiwania

##### Parameters

<i>text</i>	fraza wyszukiwania
-------------	--------------------

##### Returns

prawda gdy książka pasuje, fałsz gdy nie pasuje

#### 4.1.2.9 toCsv()

```
string Book::toCsv (
    string delimiter )
```

funkcja zamienia dane książki na napis w formacie csv

#### Parameters

<i>delimiter</i>	odstep pomiedzy danymi
------------------	------------------------

#### Returns

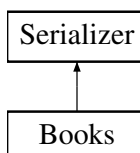
dane książki w formacie csv

## 4.2 Books Class Reference

klasa do obsługi wielu książek

```
#include <Books.h>
```

Inheritance diagram for Books:



### Public Member Functions

- [Books](#) ()  
*konstruktor*
- void [add](#) ()  
*funkcja obsługuje dodanie nowej książki*
- bool [exists](#) (int id)  
*funkcja zwraca informacje czy książka o podanym id istnieje*
- bool [isAvailable](#) (int id)  
*funkcja zwraca informacje czy książka o podanym id jest dostępna do wypożyczenia*
- [Book](#) [get](#) (int id)  
*funkcja zwraca dane książki o podanym id*
- void [rent](#) (int id)  
*funkcja obsługuje wypożyczenie książki o podanym id*
- void [returnBack](#) (int id)  
*funkcja obsługuje zwrot książki o podanym id*
- void [display](#) ()  
*funkcja wyświetla wszystkie książki*
- void [search](#) ()  
*funkcja szuka książek według frazy wyszukiwania*
- void [remove](#) ()  
*funkcja usuwa książkę o podanym id*
- void [serialize](#) ()  
*funkcja zapisuje wszystkie książki do pliku (książki reprezentowane są w formacie csv)*
- void [deserialize](#) ()  
*funkcja odczytuje książki z pliku csv*



## Additional Inherited Members

### 4.2.1 Detailed Description

klasa do obsługi wielu książek

### 4.2.2 Constructor & Destructor Documentation

#### 4.2.2.1 Books()

```
Books::Books ( ) [inline]
```

konstruktor

### 4.2.3 Member Function Documentation

#### 4.2.3.1 add()

```
void Books::add ( )
```

funkcja obsługuje dodanie nowej książki

#### 4.2.3.2 deserialize()

```
void Books::deserialize ( ) [virtual]
```

funkcja odczytuje książki z pliku csv

Reimplemented from [Serializer](#).

#### 4.2.3.3 display()

```
void Books::display ( )
```

funkcja wyświetla wszystkie książki

#### 4.2.3.4 exists()

```
bool Books::exists (
    int id )
```

funkcja zwraca informacje czy książka o podanym id istnieje

**Parameters**

<i>id</i>	identyfikator książki
-----------	-----------------------

**Returns**

prawda jeżeli książka istnieje, fałsz jeżeli nie istnieje

**4.2.3.5 get()**

```
Book Books::get (
    int id )
```

funkcja zwraca dane książki o podanym id

**Parameters**

<i>id</i>	identyfikator książki
-----------	-----------------------

**Returns**

obiekt książki

**4.2.3.6 isAvailable()**

```
bool Books::isAvailable (
    int id )
```

funkcja zwraca informacje czy książka o podanym id jest dostępna do wypożyczenia

**Parameters**

<i>id</i>	identyfikator książki
-----------	-----------------------

**Returns**

prawda jeżeli książka jest dostępna, fałszy jeżeli jest niedostępna

**4.2.3.7 remove()**

```
void Books::remove ( )
```

funkcja usuwa ksiazke o podanym id

#### 4.2.3.8 rent()

```
void Books::rent (
    int id )
```

funkcja obsluguje wypozyczenie ksiazki o podanym id

##### Parameters

<i>id</i>	identyfikator ksiazki
-----------	-----------------------

#### 4.2.3.9 returnBack()

```
void Books::returnBack (
    int id )
```

funkcja obsluguje zwrot ksiazki o podanym id

##### Parameters

<i>id</i>	identyfikator ksiazki
-----------	-----------------------

#### 4.2.3.10 search()

```
void Books::search ( )
```

funkcja szuka ksiazek wedlug frazy wyszukiwania

#### 4.2.3.11 serialize()

```
void Books::serialize ( ) [virtual]
```

funkcja zapisuje wszystkie ksiazki do pliku (ksiazki reprezentowane sa w formacie csv)

Reimplemented from [Serializer](#).

## 4.3 Rental Class Reference

klasa wypozyczenie

```
#include <Rental.h>
```

### Public Member Functions

- void [requestData](#) ()  
*funkcja wczytuje dane wypozyczenia z klawiatury*
- int [getId](#) ()  
*funkcja podaje id wypozyczenia*
- void [setId](#) (int id)  
*funkcja ustawia id wypozyczenia*
- int [getStudentId](#) ()  
*funkcja podaje id ucznia*
- int [getBookId](#) ()  
*funkcja podaje id książki*
- bool [search](#) (string text)  
*funkcja sprawdza czy id wypozyczenie, książki lub ucznia pasuje do podanej frazy wyszukiwania*
- void [display](#) ()  
*funkcja wyświetla informacje o wypozyczeniu*
- string [toCsv](#) (string delimiter)  
*funkcja zamienia dane wypozyczenie na napis w formacie csv*
- void [fromCsv](#) (string text, string delimiter)  
*funkcja odczytuje z napisu w formacie csv dane wypozyczenia*

### 4.3.1 Detailed Description

klasa wypozyczenie

### 4.3.2 Member Function Documentation

#### 4.3.2.1 display()

```
void Rental::display ( )
```

funkcja wyświetla informacje o wypozyczeniu

#### 4.3.2.2 fromCsv()

```
void Rental::fromCsv (
    string text,
    string delimiter )
```

funkcja odczytuje z napisu w formacie csv dane wypozyczenia

## Parameters

<i>text</i>	dane do odczytu
<i>delimiter</i>	odstep pomiedzy danymi

**4.3.2.3 getBookId()**

```
int Rental::getBookId ( )
```

funkcja podaje id ksiazki

## Returns

identyfikator ksiazki

**4.3.2.4 getId()**

```
int Rental::getId ( )
```

funkcja podaje id wypozyczenia

## Returns

identyfikator wypozyczenia

**4.3.2.5 getStudentId()**

```
int Rental::getStudentId ( )
```

funkcja podaje id ucznia

## Returns

identyfikator ucznia

**4.3.2.6 requestData()**

```
void Rental::requestData ( )
```

funkcja wczytuje dane wypozyczenia z klawiatury

**4.3.2.7 search()**

```
bool Rental::search (
    string text )
```

funkcja sprawdza czy id wypozyczenie, ksiazki lub ucznia pasuje do podanej frazy wyszukiwania

**Parameters**

<i>text</i>	fraza wyszukiwania
-------------	--------------------

**Returns**

prawda gdy wypożyczenie pasuje, fałsz gdy nie pasuje

**4.3.2.8 setId()**

```
void Rental::setId (
    int id )
```

funkcja ustawia id wypożyczenia

**Parameters**

<i>id</i>	identyfikator wypożyczenia
-----------	----------------------------

**4.3.2.9 toCsv()**

```
string Rental::toCsv (
    string delimiter )
```

funkcja zamienia dane wypożyczenie na napis w formacie csv

**Parameters**

<i>delimiter</i>	odstęp pomiędzy danymi
------------------	------------------------

**Returns**

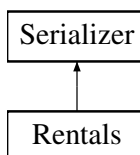
dane wypożyczenia w formacie csv

**4.4 Rentals Class Reference**

klasa do obsługi wypożyczeń

```
#include <Rentals.h>
```

Inheritance diagram for Rentals:



## Public Member Functions

- [Rentals](#) ([Books](#) \*books, [Students](#) \*students)  
*konstruktor*
- void [add](#) ()  
*funkcja obsługuje dodanie nowego wypożyczenia*
- [Rental](#) [get](#) (int id)  
*funkcja zwraca dane wypożyczenia o podanym id*
- void [display](#) ()  
*funkcja wyświetla wszystkie wypożyczenia*
- void [search](#) ()  
*funkcja szuka wypożyczenia według frazy wyszukiwania*
- void [remove](#) ()  
*funkcja usuwa wypożyczenie o podanym id*
- void [serialize](#) ()  
*funkcja zapisuje wszystkie wypożyczenia do pliku (wypożyczenia reprezentowane są w formacie csv)*
- void [deserialize](#) ()  
*funkcja odczytuje wypożyczenia z pliku csv*

## Additional Inherited Members

### 4.4.1 Detailed Description

klasa do obsługi wypożyczeń

### 4.4.2 Constructor & Destructor Documentation

#### 4.4.2.1 Rentals()

```

Rentals::Rentals (
    Books * books,
    Students * students ) [inline]
  
```

konstruktor

#### Parameters

<i>books</i>	wskaznik obiektu książek
<i>students</i>	wskaznik obiektu uczniów

### 4.4.3 Member Function Documentation

#### 4.4.3.1 add()

```
void Rentals::add ( )
```

funkcja obsługuje dodanie nowego wypożyczenia

#### 4.4.3.2 deserialize()

```
void Rentals::deserialize ( ) [virtual]
```

funkcja odczytuje wypożyczenia z pliku csv

Reimplemented from [Serializer](#).

#### 4.4.3.3 display()

```
void Rentals::display ( )
```

funkcja wyświetla wszystkie wypożyczenia

#### 4.4.3.4 get()

```
Rental Rentals::get (
    int id )
```

funkcja zwraca dane wypożyczenia o podanym id

##### Parameters

<i>id</i>	identyfikator wypożyczenia
-----------	----------------------------

##### Returns

obiekt wypożyczenia



#### 4.4.3.5 remove()

```
void Rentals::remove ( )
```

funkcja usuwa wypożyczenie o podanym id

#### 4.4.3.6 search()

```
void Rentals::search ( )
```

funkcja szuka wypożyczenia według frazy wyszukiwania

#### 4.4.3.7 serialize()

```
void Rentals::serialize ( ) [virtual]
```

funkcja zapisuje wszystkie wypożyczenia do pliku (wypożyczenia reprezentowane są w formacie csv)

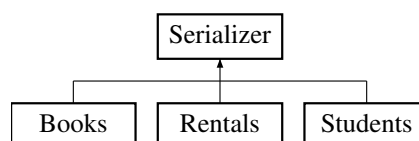
Reimplemented from [Serializer](#).

## 4.5 Serializer Class Reference

klasa bazowa zapewniająca możliwość zapisu i odczytu z pliku w formacie csv

```
#include <Serializer.h>
```

Inheritance diagram for Serializer:



### Public Member Functions

- [Serializer](#) (string [fileName](#))  
*konstruktor*
- string [getDelimiter](#) ()  
*funkcja podaje odstęp pomiędzy danymi*
- virtual void [serialize](#) ()  
*funkcja wirtualna obsługująca zapis do pliku*
- virtual void [deserialize](#) ()  
*funkcja wirtualna obsługująca odczyt z pliku*

## Protected Attributes

- string [fileName](#)
- string [delimiter](#) = ";

### 4.5.1 Detailed Description

klasa bazowa zapewniajaca mozliwosc zapisu i odczytu z pliku w formacie csv

### 4.5.2 Constructor & Destructor Documentation

#### 4.5.2.1 Serializer()

```
Serializer::Serializer (  
    string fileName ) [inline]
```

konstruktor

Parameters

<i>fileName</i>	nazwa pliku
-----------------	-------------

### 4.5.3 Member Function Documentation

#### 4.5.3.1 deserialize()

```
virtual void Serializer::deserialize ( ) [inline], [virtual]
```

funkcja wirtualna obslugujaca odczyt z pliku

Reimplemented in [Books](#), [Rentals](#), and [Students](#).

#### 4.5.3.2 getDelimiter()

```
string Serializer::getDelimiter ( ) [inline]
```

funkcja podaje odstep pomiedzy danymi

Returns

odstep pomiedzy danymi

#### 4.5.3.3 serialize()

```
virtual void Serializer::serialize ( ) [inline], [virtual]
```

funkcja wirtualna obslugujaca zapis do pliku

Reimplemented in [Books](#), [Rentals](#), and [Students](#).

### 4.5.4 Member Data Documentation

#### 4.5.4.1 delimiter

```
string Serializer::delimiter = ";" [protected]
```

#### 4.5.4.2 fileName

```
string Serializer::fileName [protected]
```

## 4.6 Student Class Reference

klasa uczen

```
#include <Student.h>
```

### Public Member Functions

- void [requestData](#) ()  
*funkcja wczytuje dane ucznia z klawiatury*
- int [getId](#) ()  
*funkcja podaje id ucznia*
- bool [search](#) (string text)  
*funkcja sprawdza czy imie lub nazwisko ucznia pasuje do podanej frazy wyszukiwania*
- void [display](#) ()  
*funkcja wyswietla informacje o uczniu*
- string [toCsv](#) (string delimiter)  
*funkcja zamienia dane ucznia na napis w formacie csv*
- void [fromCsv](#) (string text, string delimiter)  
*funkcja odczytuje z napisu w formacie csv dane ucznia*

#### 4.6.1 Detailed Description

klasa uczen

## 4.6.2 Member Function Documentation

### 4.6.2.1 display()

```
void Student::display ( )
```

funkcja wyswietla informacje o uczniu

### 4.6.2.2 fromCsv()

```
void Student::fromCsv (
    string text,
    string delimiter )
```

funkcja odczytuje z napisu w formacie csv dane ucznia

#### Parameters

<i>text</i>	dane do odczytu
<i>delimiter</i>	odstep pomiedzy danymi

### 4.6.2.3 getId()

```
int Student::getId ( )
```

funkcja podaje id ucznia

#### Returns

identyfikator ucznia

### 4.6.2.4 requestData()

```
void Student::requestData ( )
```

funkcja wczytuje dane ucznia z klawiatury

### 4.6.2.5 search()

```
bool Student::search (
    string text )
```

funkcja sprawdza czy imie lub nazwisko ucznia pasuje do podanej frazy wyszukiwania

## Parameters

<i>text</i>	fraza wyszukiwania
-------------	--------------------

## Returns

prawda gdy uczen pasuje, falsz gdy nie pasuje

## 4.6.2.6 toCsv()

```
string Student::toCsv (
    string delimiter )
```

funkcja zamienia dane ucznia na napis w formacie csv

## Parameters

<i>delimiter</i>	odstep pomiedzy danymi
------------------	------------------------

## Returns

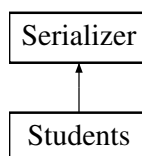
dane ucznia w formacie csv

## 4.7 Students Class Reference

klasa do obsługi wielu uczniow

```
#include <Students.h>
```

Inheritance diagram for Students:



### Public Member Functions

- [Students](#) ()  
*konstruktor*
- void [add](#) ()  
*funkcja obsługi dodanie nowego ucznia*
- bool [exists](#) (int id)  
*funkcja sprawdza czy uczen o podanym id istnieje*

- `Student get` (int id)  
*funkcja zwraca dane ucznia o podanym id*
- void `display` ()  
*funkcja wyswietla wszystkich uczniow*
- void `search` ()  
*funkcja wyszukuje ucznia wedlug frazy wyszukiwania*
- void `remove` ()  
*funkcja usuwa ucznia o podanym id*
- void `serialize` ()  
*funkcja zapisuje wszystkich uczniow do pliku (uczen reprezentowane sa w formacie csv)*
- void `deserialize` ()  
*funkcja odczytuje uczniow z pliku csv*

## Additional Inherited Members

### 4.7.1 Detailed Description

klasa do obslugi wielu uczniow

### 4.7.2 Constructor & Destructor Documentation

#### 4.7.2.1 Students()

```
Students::Students ( ) [inline]
```

konstruktor

### 4.7.3 Member Function Documentation

#### 4.7.3.1 add()

```
void Students::add ( )
```

funkcja obluguje dodanie nowego ucznia

#### 4.7.3.2 deserialize()

```
void Students::deserialize ( ) [virtual]
```

funkcja odczytuje uczniow z pliku csv

Reimplemented from [Serializer](#).

#### 4.7.3.3 display()

```
void Students::display ( )
```

funkcja wyswietla wszystkich uczniow

#### 4.7.3.4 exists()

```
bool Students::exists (
    int id )
```

funkcja sprawdza czy uczen o podanym id istnieje

##### Parameters

<i>id</i>	identyfikator ucznia
-----------	----------------------

##### Returns

prawda jezeli uczen istnieje, falsz jezeli nie istnieje

#### 4.7.3.5 get()

```
Student Students::get (
    int id )
```

funkcja zwraca dane ucznia o podanym id

##### Parameters

<i>id</i>	identyfikator ucznia
-----------	----------------------

**Returns**

obiekt ucznia

**4.7.3.6 remove()**

```
void Students::remove ( )
```

funkcja usuwa ucznia o podanym id

**4.7.3.7 search()**

```
void Students::search ( )
```

funkcja wyszukuje ucznia według frazy wyszukiwania

**4.7.3.8 serialize()**

```
void Students::serialize ( ) [virtual]
```

funkcja zapisuje wszystkich uczniów do pliku (uczen reprezentowane sa w formacie csv)

Reimplemented from [Serializer](#).



## Chapter 5

# File Documentation

### 5.1 C:/Users/Kuba/Tajne/Book.cpp File Reference

```
#include <iostream>
#include <string>
#include <sstream>
#include "Book.h"
```

### 5.2 C:/Users/Kuba/Tajne/Book.h File Reference

```
#include <string>
```

#### Classes

- class [Book](#)  
*klasa ksiazka*

### 5.3 Book.h

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

```
1 #pragma once
2 #include<string>
3
4 using namespace std;
5
6 class Book
7 {
8     private:
9         int book_id;
10        string book_name;
11        string author;
12        float price;
13        int status = 0;
14
15     public:
16        void requestData();
17
18
19
```

```
22     int getId();
23
26     int getStatus();
27
29     void rent();
30
32     void returnBack();
33
37     bool search(string text);
38
40     void display();
41
45     string toCsv(string delimiter);
46
50     void fromCsv(string text, string delimiter);
51 };
```

## 5.4 C:/Users/Kuba/Tajne/Books.cpp File Reference

```
#include <iostream>
#include <string>
#include <fstream>
#include <map>
#include "Book.h"
#include "Books.h"
```

## 5.5 C:/Users/Kuba/Tajne/Books.h File Reference

```
#include <map>
#include "Serializer.h"
#include "Book.h"
```

### Classes

- class [Books](#)

*klasa do obsługi wielu książek*

## 5.6 Books.h

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

```
1 #pragma once
2 #include <map>
3 #include "Serializer.h"
4 #include "Book.h"
5
6 using namespace std;
7
9 class Books : public Serializer
10 {
11     private:
12         map<int, Book> books;
13
14     public:
16         Books() : Serializer("books") {}
17
19         void add();
20
24         bool exists(int id);
```

```

25
29     bool isAvailable(int id);
30
34     Book get(int id);
35
38     void rent(int id);
39
42     void returnBack(int id);
43
45     void display();
46
48     void search();
49
51     void remove();
52
54     void serialize();
55
57     void deserialize();
58 };

```

## 5.7 C:/Users/Kuba/Tajne/main.cpp File Reference

```

#include <iostream>
#include <string>
#include <memory>
#include <vector>
#include <conio.h>
#include "Serializer.h"
#include "Books.h"
#include "Students.h"
#include "Rentals.h"

```

### Functions

- void `deserialize` (vector< shared\_ptr< `Serializer` > > serializers)
- void `displayMenu` ()  
*funkcja do wyswietlanie menu*
- void `operationConfirmation` ()  
*funkcja czeka na wybor polecenia*
- int `main` ()  
*glowna funkcja obsugujaca polecenia w programie*

### 5.7.1 Function Documentation

#### 5.7.1.1 deserialize()

```

void deserialize (
    vector< shared_ptr< Serializer > > serializers )

```

#### 5.7.1.2 displayMenu()

```
void displayMenu ( )
```

funkcja do wyswietlanie menu

#### 5.7.1.3 main()

```
int main ( )
```

glowna funkcja obsugujaca polecenia w programie

#### 5.7.1.4 operationConfirmation()

```
void operationConfirmation ( )
```

funkcja czeka na wybor polecenia

## 5.8 C:/Users/Kuba/Tajne/Rental.cpp File Reference

```
#include <iostream>
#include <string>
#include <sstream>
#include <chrono>
#include "Rental.h"
#include "Books.h"
```

### Functions

- string [currentDateTime](#) ()

#### 5.8.1 Function Documentation

##### 5.8.1.1 currentDateTime()

```
string currentDateTime ( )
```

## 5.9 C:/Users/Kuba/Tajne/Rental.h File Reference

```
#include <string>
```

### Classes

- class [Rental](#)  
*klasa wypozyczenie*

## 5.10 Rental.h

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

```
1 #pragma once
2 #include<string>
3
4 using namespace std;
5
6
7 class Rental
8 {
9     private:
10         int rental_id;
11         int student_id;
12         int book_id;
13         string rental_date;
14     public:
15         void requestData();
16
17         int getId();
18
19         void setId(int id);
20
21         int getStudentId();
22
23         int getBookId();
24
25         bool search(string text);
26
27         void display();
28
29         string toCsv(string delimiter);
30
31         void fromCsv(string text, string delimiter);
32 };
33
```

## 5.11 C:/Users/Kuba/Tajne/Rentals.cpp File Reference

```
#include <iostream>
#include <string>
#include <fstream>
#include <map>
#include "Rental.h"
#include "Rentals.h"
```

## 5.12 C:/Users/Kuba/Tajne/Rentals.h File Reference

```
#include <map>
#include "Serializer.h"
#include "Rental.h"
#include "Books.h"
#include "Students.h"
```

## Classes

- class [Rentals](#)

*klasa do obsługi wypożyczeń*

### 5.13 Rentals.h

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

```
1 #pragma once
2 #include <map>
3 #include "Serializer.h"
4 #include "Rental.h"
5 #include "Books.h"
6 #include "Students.h"
7
8 using namespace std;
9
11 class Rentals : public Serializer
12 {
13     private:
14         map<int, Rental> rentals;
15         Books* books;
16         Students* students;
17
18     public:
22         Rentals(Books* books, Students* students) : Serializer("rentals")
23         {
24             this->books = books;
25             this->students = students;
26         }
27
29         void add();
30
34         Rental get(int id);
35
37         void display();
38
40         void search();
41
43         void remove();
44
46         void serialize();
47
49         void deserialize();
50 };
```

### 5.14 C:/Users/Kuba/Tajne/Serializer.h File Reference

```
#include <string>
```

## Classes

- class [Serializer](#)

*klasa bazowa zapewniająca możliwość zapisu i odczytu z pliku w formacie csv*

## 5.15 Serializer.h

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

```
1 #pragma once
2 #include<string>
3
4 using namespace std;
5
6
7 class Serializer
8 {
9     protected:
10         string fileName;
11         string delimiter = ";";
12
13     public:
14         Serializer(string fileName)
15         {
16             this -> fileName = fileName;
17         };
18
19         string getDelimiter()
20         {
21             return delimiter;
22         }
23
24         virtual void serialize()
25         {
26             cout << "Serialization not implemented";
27         };
28
29         virtual void deserialize()
30         {
31             cout << "Deserialization not implemented";
32         };
33 };
34
```

## 5.16 C:/Users/Kuba/Tajne/Student.cpp File Reference

```
#include <iostream>
#include <string>
#include <sstream>
#include "Student.h"
```

## 5.17 C:/Users/Kuba/Tajne/Student.h File Reference

```
#include <string>
```

### Classes

- class [Student](#)  
*klasa uczen*

## 5.18 Student.h

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

```
1 #pragma once
2 #include<string>
3
4 using namespace std;
5
6
7 class Student
8 {
9     private:
10         int student_id;
11         string name;
12         string surname;
13
14     public:
15         void requestData();
16
17         int getId();
18
19         bool search(string text);
20
21         void display();
22
23         string toCsv(string delimiter);
24
25         void fromCsv(string text, string delimiter);
26 };
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
```

## 5.19 C:/Users/Kuba/Tajne/Students.cpp File Reference

```
#include <iostream>
#include <string>
#include <fstream>
#include <map>
#include "Student.h"
#include "Students.h"
```

## 5.20 C:/Users/Kuba/Tajne/Students.h File Reference

```
#include <iostream>
#include <map>
#include "Serializer.h"
#include "Student.h"
```

### Classes

- class [Students](#)

*klasa do obsługi wielu uczniów*



## 5.21 Students.h

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

```
1 #pragma once
2 #include<iostream>
3 #include <map>
4 #include "Serializer.h"
5 #include "Student.h"
6
7 using namespace std;
8
10 class Students : public Serializer
11 {
12     private:
13         map<int, Student> students;
14
15     public:
17         Students() : Serializer("students") {}
18
20         void add();
21
25         bool exists(int id);
26
30         Student get(int id);
31
33         void display();
34
36         void search();
37
39         void remove();
40
42         void serialize();
43
45         void deserialize();
46 };
```



# Index

- add
  - Books, 11
  - Rentals, 18
  - Students, 24
- Book, 7
  - display, 8
  - fromCsv, 8
  - getId, 8
  - getStatus, 8
  - rent, 8
  - requestData, 9
  - returnBack, 9
  - search, 9
  - toCsv, 9
- Books, 10
  - add, 11
  - Books, 11
  - deserialize, 11
  - display, 11
  - exists, 11
  - get, 12
  - isAvailable, 12
  - remove, 12
  - rent, 13
  - returnBack, 13
  - search, 13
  - serialize, 13
- C:/Users/Kuba/Tajne/Book.cpp, 27
- C:/Users/Kuba/Tajne/Book.h, 27
- C:/Users/Kuba/Tajne/Books.cpp, 28
- C:/Users/Kuba/Tajne/Books.h, 28
- C:/Users/Kuba/Tajne/main.cpp, 29
- C:/Users/Kuba/Tajne/Rental.cpp, 30
- C:/Users/Kuba/Tajne/Rental.h, 31
- C:/Users/Kuba/Tajne/Rentals.cpp, 31
- C:/Users/Kuba/Tajne/Rentals.h, 31, 32
- C:/Users/Kuba/Tajne/Serializer.h, 32, 33
- C:/Users/Kuba/Tajne/Student.cpp, 33
- C:/Users/Kuba/Tajne/Student.h, 33, 34
- C:/Users/Kuba/Tajne/Students.cpp, 34
- C:/Users/Kuba/Tajne/Students.h, 34, 35
- currentDateTime
  - Rental.cpp, 30
- delimiter
  - Serializer, 21
- deserialize
  - Books, 11
  - main.cpp, 29
  - Rentals, 18
  - Serializer, 20
  - Students, 24
- display
  - Book, 8
  - Books, 11
  - Rental, 14
  - Rentals, 18
  - Student, 22
  - Students, 25
- displayMenu
  - main.cpp, 29
- exists
  - Books, 11
  - Students, 25
- fileName
  - Serializer, 21
- fromCsv
  - Book, 8
  - Rental, 14
  - Student, 22
- get
  - Books, 12
  - Rentals, 18
  - Students, 25
- getBookId
  - Rental, 15
- getDelimiter
  - Serializer, 20
- getId
  - Book, 8
  - Rental, 15
  - Student, 22
- getStatus
  - Book, 8
- getStudentId
  - Rental, 15
- isAvailable
  - Books, 12
- main
  - main.cpp, 30
- main.cpp
  - deserialize, 29
  - displayMenu, 29
  - main, 30

- operationConfirmation, 30
- operationConfirmation
  - main.cpp, 30
- remove
  - Books, 12
  - Rentals, 18
  - Students, 26
- rent
  - Book, 8
  - Books, 13
- Rental, 14
  - display, 14
  - fromCsv, 14
  - getBookId, 15
  - getId, 15
  - getStudentId, 15
  - requestData, 15
  - search, 15
  - setId, 16
  - toCsv, 16
- Rental.cpp
  - currentDateTime, 30
- Rentals, 16
  - add, 18
  - deserialize, 18
  - display, 18
  - get, 18
  - remove, 18
  - Rentals, 17
  - search, 19
  - serialize, 19
- requestData
  - Book, 9
  - Rental, 15
  - Student, 22
- returnBack
  - Book, 9
  - Books, 13
- search
  - Book, 9
  - Books, 13
  - Rental, 15
  - Rentals, 19
  - Student, 22
  - Students, 26
- serialize
  - Books, 13
  - Rentals, 19
  - Serializer, 20
  - Students, 26
- Serializer, 19
  - delimiter, 21
  - deserialize, 20
  - fileName, 21
  - getDelimiter, 20
  - serialize, 20
  - Serializer, 20
- setId
  - Rental, 16
- Student, 21
  - display, 22
  - fromCsv, 22
  - getId, 22
  - requestData, 22
  - search, 22
  - toCsv, 23
- Students, 23
  - add, 24
  - deserialize, 24
  - display, 25
  - exists, 25
  - get, 25
  - remove, 26
  - search, 26
  - serialize, 26
  - Students, 24
- toCsv
  - Book, 9
  - Rental, 16
  - Student, 23