

DSA_EHU_Project

Generated by Doxygen 1.9.5

1 Namespace Index	1
1.1 Package List	1
2 Class Index	3
2.1 Class List	3
3 File Index	5
3.1 File List	5
4 Namespace Documentation	7
4.1 Package fileHandling	7
4.2 Package people	7
4.3 Package relationships	7
5 Class Documentation	9
5.1 fileHandling.FileHandling Class Reference	9
5.1.1 Detailed Description	9
5.1.2 Member Function Documentation	9
5.1.2.1 readFile()	9
5.1.2.2 writeToFile()	10
5.2 Menu Class Reference	11
5.2.1 Detailed Description	11
5.2.2 Member Function Documentation	11
5.2.2.1 main()	11
5.2.2.2 menu()	12
5.3 people.People Class Reference	13
5.3.1 Constructor & Destructor Documentation	14
5.3.1.1 People()	14
5.3.2 Member Function Documentation	14
5.3.2.1 addPersonFromString()	14
5.3.2.2 anySymbolLineFormatter()	15
5.3.2.3 dashLineFormatter()	15
5.3.2.4 fillTheClasses()	16
5.3.2.5 findMatchesByFile()	16
5.3.2.6 findPeopleBornBetween()	17
5.3.2.7 findPeopleByBirthplace()	18
5.3.2.8 findPeopleByHome()	18
5.3.2.9 findPersonById()	19
5.3.2.10 findPersonBySurname()	20
5.3.2.11 getPeople()	20
5.3.2.12 printAllClasses()	21
5.3.2.13 printAllPeople()	21
5.3.2.14 printPeopleBornBetween()	21
5.3.2.15 printPeopleByBirthplace()	22

5.3.2.16 printPeopleByHome()	23
5.3.2.17 printRelationshipsByLastname()	24
5.3.2.18 quicksortByBirthplaceSurnameAndName()	25
5.3.2.19 removePersonById()	25
5.3.2.20 setPeople()	26
5.3.2.21 sortByBirthplaceSurnameAndName()	26
5.3.2.22 sortBySurnameAndName()	27
5.3.2.23 sortClasses()	28
5.3.2.24 sortPeopleByMovies()	28
5.3.2.25 writeToFile()	29
5.3.3 Member Data Documentation	30
5.3.3.1 FIRST_LINE_PEOPLE	30
5.3.3.2 profiles	30
5.4 people.Person Class Reference	30
5.4.1 Constructor & Destructor Documentation	31
5.4.1.1 Person()	31
5.4.2 Member Function Documentation	31
5.4.2.1 equals()	31
5.4.2.2 getBirthdate()	32
5.4.2.3 getBirthplace()	33
5.4.2.4 getFilms()	33
5.4.2.5 getGender()	33
5.4.2.6 getGroupCode()	34
5.4.2.7 getHome()	34
5.4.2.8 getIdPerson()	34
5.4.2.9 getLastName()	35
5.4.2.10 getName()	35
5.4.2.11 getStudiedAt()	36
5.4.2.12 getWorkplaces()	36
5.4.2.13 hashCode()	36
5.4.2.14 setBirthdate()	36
5.4.2.15 setBirthplace()	37
5.4.2.16 setFilms()	37
5.4.2.17 setGender()	37
5.4.2.18 setGroupCode()	38
5.4.2.19 setHome()	38
5.4.2.20 setIdPerson()	38
5.4.2.21 setLastName()	38
5.4.2.22 setName()	39
5.4.2.23 setStudiedAt()	39
5.4.2.24 setWorkplaces()	39
5.4.2.25 toString()	40

5.5 people.Quicksort Class Reference	40
5.5.1 Member Function Documentation	40
5.5.1.1 sort()	40
5.6 relationships.Relationship Class Reference	40
5.6.1 Detailed Description	41
5.6.2 Constructor & Destructor Documentation	41
5.6.2.1 Relationship()	41
5.6.3 Member Function Documentation	41
5.6.3.1 equals()	41
5.6.3.2 getFriend1()	42
5.6.3.3 getFriend2()	43
5.6.3.4 hashCode()	43
5.6.3.5 setFriend1()	43
5.6.3.6 setFriend2()	44
5.6.3.7 toString()	44
5.7 relationships.Relationships Class Reference	45
5.7.1 Constructor & Destructor Documentation	45
5.7.1.1 Relationships()	45
5.7.2 Member Function Documentation	45
5.7.2.1 addRelationship()	45
5.7.2.2 deleteRelationship()	46
5.7.2.3 findRelationshipsById()	47
5.7.2.4 findRelationshipsByIds()	47
5.7.2.5 getRelations()	48
5.7.2.6 printAllRelationships()	48
5.7.2.7 toString()	49
5.7.2.8 writeToFile()	49
5.7.3 Member Data Documentation	50
5.7.3.1 FIRST_LINE_RELATIONSHIPS	50
5.8 UnitTests Class Reference	50
5.8.1 Member Function Documentation	51
5.8.1.1 done()	51
6 File Documentation	53
6.1 D:/Projekty/DSA_ehu_JavaProject/src/fileHandling/FileHandling.java File Reference	53
6.2 D:/Projekty/DSA_ehu_JavaProject/src/Menu.java File Reference	53
6.3 D:/Projekty/DSA_ehu_JavaProject/src/people/People.java File Reference	53
6.4 D:/Projekty/DSA_ehu_JavaProject/src/people/Person.java File Reference	54
6.5 D:/Projekty/DSA_ehu_JavaProject/src/people/Quicksort.java File Reference	54
6.6 D:/Projekty/DSA_ehu_JavaProject/src/relationships/Relationship.java File Reference	54
6.7 D:/Projekty/DSA_ehu_JavaProject/src/relationships/Relationships.java File Reference	54
6.8 D:/Projekty/DSA_ehu_JavaProject/src/UnitTests.java File Reference	55

Chapter 1

Namespace Index

1.1 Package List

Here are the packages with brief descriptions (if available):

fileHandling	7
people	7
relationships	7

Chapter 2

Class Index

2.1 Class List

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

fileHandling.FileHandling	9
Menu	11
people.People	13
people.Person	30
people.Quicksort	40
relationships.Relationship	40
relationships.Relationships	45
UnitTests	50

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

D:/Projekty/DSA_ehu_JavaProject/src/ Menu.java	53
D:/Projekty/DSA_ehu_JavaProject/src/ UnitTests.java	55
D:/Projekty/DSA_ehu_JavaProject/src/fileHandling/ FileHandling.java	53
D:/Projekty/DSA_ehu_JavaProject/src/people/ People.java	53
D:/Projekty/DSA_ehu_JavaProject/src/people/ Person.java	54
D:/Projekty/DSA_ehu_JavaProject/src/people/ Quicksort.java	54
D:/Projekty/DSA_ehu_JavaProject/src/relationships/ Relationship.java	54
D:/Projekty/DSA_ehu_JavaProject/src/relationships/ Relationships.java	54

Chapter 4

Namespace Documentation

4.1 Package fileHandling

Classes

- class [FileHandling](#)

4.2 Package people

Classes

- class [People](#)
- class [Person](#)
- class [Quicksort](#)

4.3 Package relationships

Classes

- class [Relationship](#)
- class [Relationships](#)

Chapter 5

Class Documentation

5.1 fileHandling.FileHandling Class Reference

Static Public Member Functions

- static List< String > [readFile](#) (String fileName) throws FileNotFoundException
- static void [writeToFile](#) (String output, String fileName) throws IOException

5.1.1 Detailed Description

This class responsible for handling operations related to files It has classes which writes and rides text from/to files

5.1.2 Member Function Documentation

5.1.2.1 [readFile\(\)](#)

```
static List< String > fileHandling.FileHandling.readFile (  
    String fileName ) throws FileNotFoundException [static]
```

This method read lines from the whole file.

Parameters

<i>fileName</i>	is String parameter with a file name
-----------------	--------------------------------------

Returns

list of Strings

Here is the caller graph for this function:

**5.1.2.2 writeToFile()**

```
static void fileHandling.FileHandling.writeToFile (
    String output,
    String fileName ) throws IOException [static]
```

This method write String to the file.

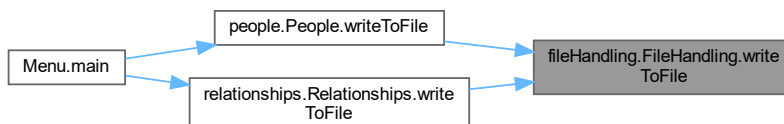
Parameters

<i>output</i>	String which will be written to file
<i>fileName</i>	String contains name of file where output will be written

Exceptions

<i>IOException</i>	This throws could occur when an I/O exception of some sort has occurred.
--------------------	--

Here is the caller graph for this function:



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

- D:/Projekty/DSA_ehu_JavaProject/src/fileHandling/[FileHandling.java](#)

5.2 Menu Class Reference

Static Public Member Functions

- static void `main` (String[] args)
- static int `menu` ()

5.2.1 Detailed Description

This class is currently the main class of the projects. It creates the menu that allows user to insert People and Relationships and view them.

5.2.2 Member Function Documentation

5.2.2.1 `main()`

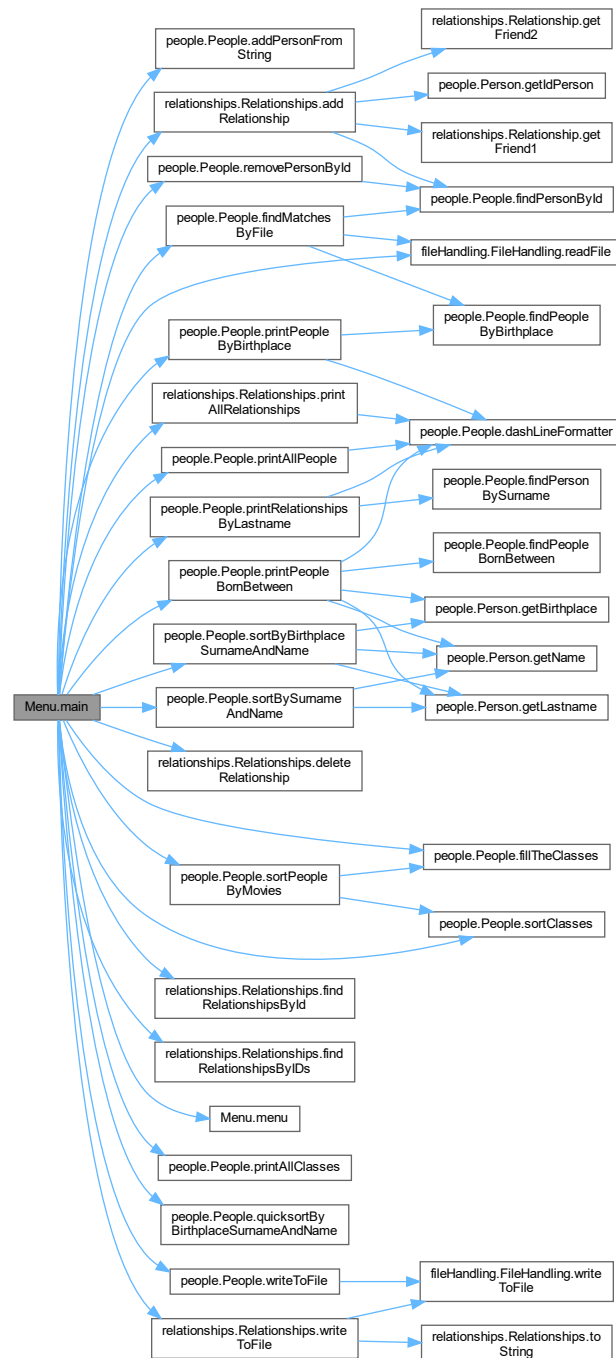
```
static void Menu.main (  
    String[] args ) [static]
```

This method consist of calls to all functions related to the options presented in menu.

Parameters

<code>args</code>	
-------------------	--

Here is the call graph for this function:



5.2.2.2 menu()

```
static int Menu.menu ( ) [static]
```

This method prints menu options of our programming project and reads input from the user.

Returns

integer

Here is the caller graph for this function:



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

- D:/Projekty/DSA_ehu_JavaProject/src/[Menu.java](#)

5.3 people.People Class Reference

Public Member Functions

- [People](#) ()
- List< [Person](#) > [getPeople](#) ()
- void [setPeople](#) (List< [Person](#) > people)
- void [printAllPeople](#) ()
- void [addPersonFromString](#) (String data) throws ParseException
- [Person](#) [findPersonById](#) (String id)
- List< [Person](#) > [findPersonBySurname](#) (String surname)
- void [printRelationshipsByLastname](#) (String lastname, [Relationships](#) relationships)
- List< [Person](#) > [findPeopleByHome](#) (String home)
- void [printPeopleByHome](#) (String home)
- List< [Person](#) > [findPeopleByBirthplace](#) (String birthplace)
- void [printPeopleByBirthplace](#) (String birthplace)
- List< [Person](#) > [findPeopleBornBetween](#) (int yearMin, int yearMax)
- void [printPeopleBornBetween](#) (int yearMin, int yearMax)
- void [writeToFile](#) (String filename) throws IOException
- void [removePersonById](#) (String id)
- void [findMatchesByFile](#) (String filename) throws IOException
- void [sortBySurnameAndName](#) ()
- void [sortByBirthplaceSurnameAndName](#) ()
- void [quicksortByBirthplaceSurnameAndName](#) ()
- void [fillTheClasses](#) ()
- void [printAllClasses](#) ()
- void [sortClasses](#) ()
- void [sortPeopleByMovies](#) ()

Static Public Member Functions

- static String [dashLineFormatter](#) (String line)
- static String [anySymbolLineFormatter](#) (String line, String symbol)

Public Attributes

- `HashMap< List< String >, List< Person > > profiles = new HashMap<>()`

Static Public Attributes

- `static final String FIRST_LINE_PEOPLE = "idperson,name,lastname,birthdate,gender,birthplace,home,studiedat,workplaces,fil`

5.3.1 Constructor & Destructor Documentation

5.3.1.1 [People](#)()

```
people.People.People ( )
```

Constructor for a [People](#) class.

5.3.2 Member Function Documentation

5.3.2.1 `addPersonFromString()`

```
void people.People.addPersonFromString (
    String data ) throws ParseException
```

This method check if the user ID is already in list of people, if not - add person from string to list of people

Parameters

<i>data</i>	is string containing data for creating a new object of Person class.
-------------	--

Exceptions

<i>ParseException</i>	Signals that an error has been reached unexpectedly while parsing.
-----------------------	--

Here is the caller graph for this function:



5.3.2.2 anySymbolLineFormatter()

```
static String people.People.anySymbolLineFormatter (  
    String line,  
    String symbol ) [static]
```

A little function that takes a line containing column names and returns a line of dashes adjusted to the length of passed line

Parameters

<i>line</i>	String with column names
<i>symbol</i>	String to be used as a delimiter line

Returns

String of passed symbol

5.3.2.3 dashLineFormatter()

```
static String people.People.dashLineFormatter (  
    String line ) [static]
```

A little function that takes a line containing column names and returns a line of dashes adjusted to the length of passed line

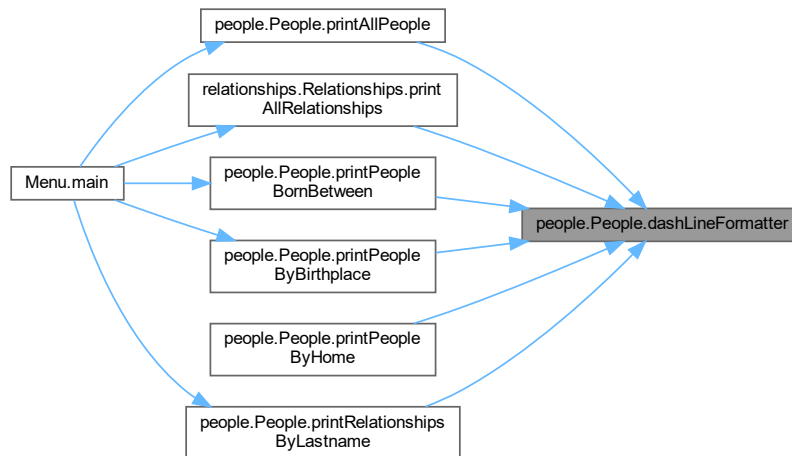
Parameters

<i>line</i>	String with column names
-------------	--------------------------

Returns

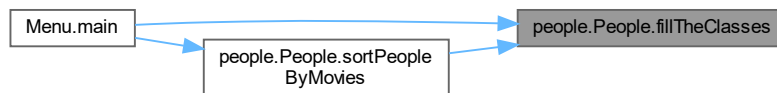
String of dashes

Here is the caller graph for this function:

**5.3.2.4 fillTheClasses()**

```
void people.People.fillTheClasses ( )
```

This function builds a classes of movies containing peoples who likes the same sets of movies Here is the caller graph for this function:

**5.3.2.5 findMatchesByFile()**

```
void people.People.findMatchesByFile (
    String filename ) throws IOException
```

The function takes the name of the .txt file, reads the contents of the file being the ID's of people and prints the lists of people whose birthplace is equal to hometown of people from file

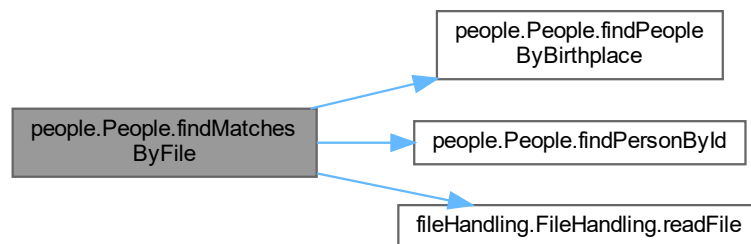
Parameters

<i>filename</i>	String - name of the file without extension
-----------------	---

Exceptions

<i>IOException</i>	in case the file causes issues
--------------------	--------------------------------

Here is the call graph for this function:



Here is the caller graph for this function:



5.3.2.6 findPeopleBornBetween()

```
List< Person > people.People.findPeopleBornBetween (
    int yearMin,
    int yearMax )
```

This method finds a list of people born between passed years

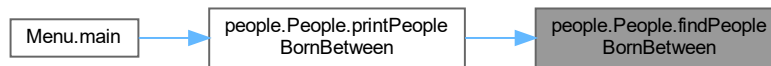
Parameters

<i>yearMin</i>	int
<i>yearMax</i>	int

Returns

List<Person> list of found people

Here is the caller graph for this function:

**5.3.2.7 findPeopleByBirthplace()**

```
List< Person > people.People.findPeopleByBirthplace (
    String birthplace )
```

This method finds a list of people with the passed birthplace

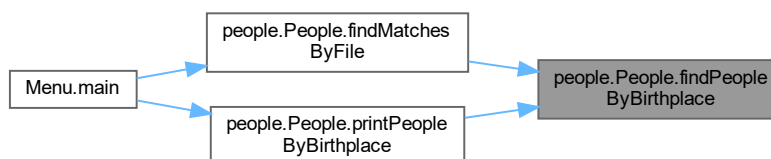
Parameters

<i>birthplace</i>	String
-------------------	--------

Returns

List<Person> list of found people

Here is the caller graph for this function:

**5.3.2.8 findPeopleByHome()**

```
List< Person > people.People.findPeopleByHome (
    String home )
```

This method finds a list of people with the passed hometown

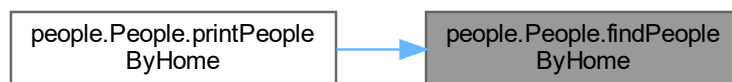
Parameters

<i>home</i>	String
-------------	--------

Returns

List<Person> list of found people

Here is the caller graph for this function:



5.3.2.9 findPersonById()

```
Person people.People.findPersonById (  
    String id )
```

This method finds `Person` object with corresponding ID in people list.

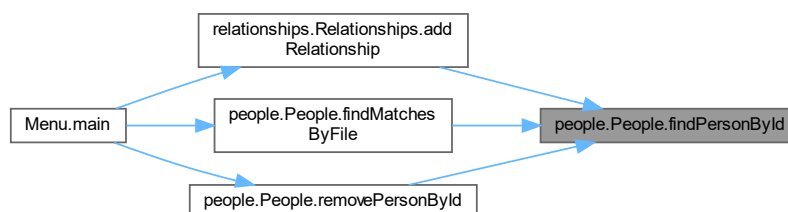
Parameters

<i>id</i>	string
-----------	--------

Returns

`Person` object

Here is the caller graph for this function:



5.3.2.10 findPersonBySurname()

```
List< Person > people.People.findPersonBySurname (
    String surname )
```

This method finds List of [Person](#) objects with corresponding surname in people list.

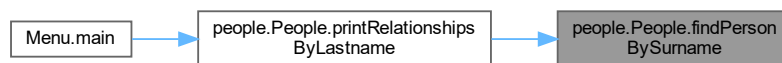
Parameters

<i>surname</i>	string
----------------	--------

Returns

List<Person> object

Here is the caller graph for this function:



5.3.2.11 getPeople()

```
List< Person > people.People.getPeople ( )
```

Getter for a people field.

Returns

list of [Person](#) type objects.

5.3.2.12 printAllClasses()

```
void people.People.printAllClasses ( )
```

This function prints all movies classes and all the people who like them Here is the caller graph for this function:



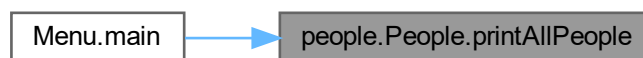
5.3.2.13 printAllPeople()

```
void people.People.printAllPeople ( )
```

This method prints out the list of all the people loaded to the program. Here is the call graph for this function:



Here is the caller graph for this function:



5.3.2.14 printPeopleBornBetween()

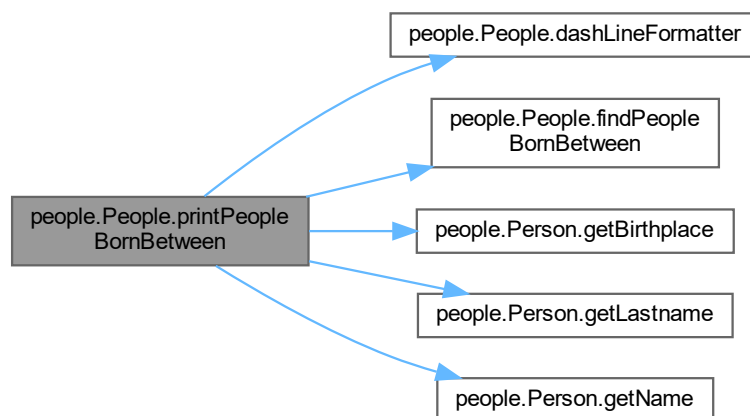
```
void people.People.printPeopleBornBetween (
    int yearMin,
    int yearMax )
```

This method prints people born between passed years

Parameters

<i>yearMin</i>	int
<i>yearMax</i>	int

Here is the call graph for this function:



Here is the caller graph for this function:



5.3.2.15 printPeopleByBirthplace()

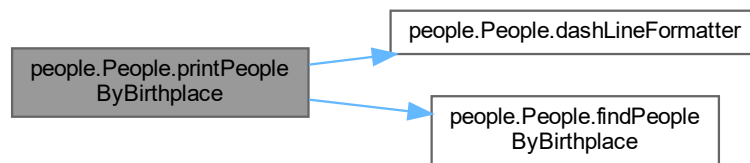
```
void people.People.printPeopleByBirthplace (
    String birthplace )
```

This method prints people with a given birthplace

Parameters

<i>birthplace</i>	String
-------------------	--------

Here is the call graph for this function:



Here is the caller graph for this function:



5.3.2.16 printPeopleByHome()

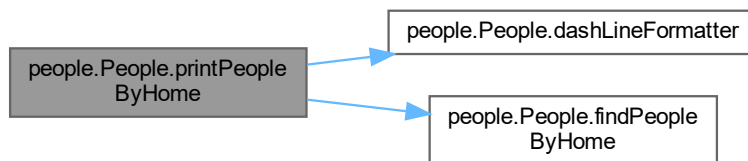
```
void people.People.printPeopleByHome (  
    String home )
```

This method prints people with a given hometown

Parameters

<i>home</i>	String
-------------	--------

Here is the call graph for this function:



5.3.2.17 printRelationshipsByLastname()

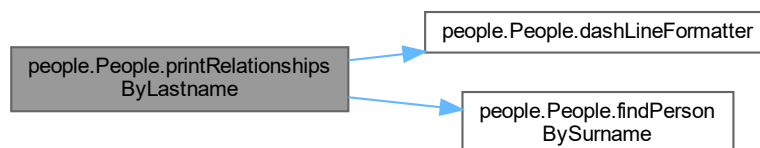
```
void people.People.printRelationshipsByLastname (
    String lastname,
    Relationships relationships )
```

This method prints relationships of people with a given lastname

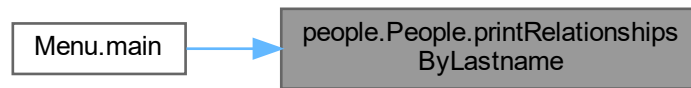
Parameters

<i>lastname</i>	String
<i>relationships</i>	Relationships

Here is the call graph for this function:



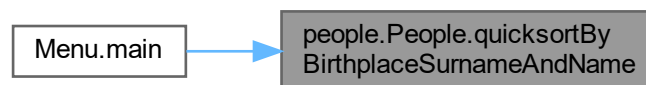
Here is the caller graph for this function:



5.3.2.18 quicksortByBirthplaceSurnameAndName()

```
void people.People.quicksortByBirthplaceSurnameAndName ( )
```

This function sorts people by birthplace surname and name with QuickSort Here is the caller graph for this function:



5.3.2.19 removePersonById()

```
void people.People.removePersonById (
    String id )
```

This method removes person with a certain id from the list of people If person with that id does not exist, prints out a message

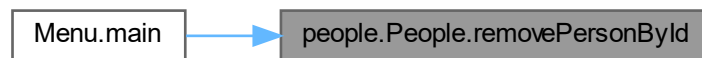
Parameters

<i>id</i>	String
-----------	--------

Here is the call graph for this function:



Here is the caller graph for this function:



5.3.2.20 setPeople()

```
void people.People.setPeople (
    List< Person > people )
```

Setter for a people field.

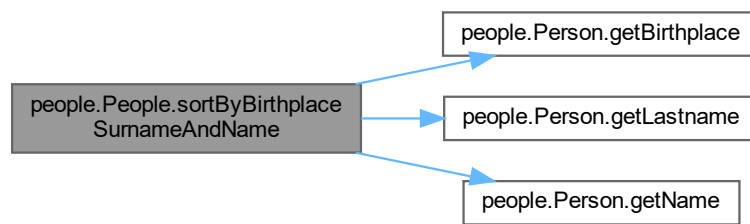
Parameters

<i>people</i>	is list of <code>Person</code> type objects.
---------------	--

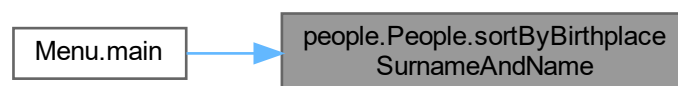
5.3.2.21 sortByBirthplaceSurnameAndName()

```
void people.People.sortByBirthplaceSurnameAndName ( )
```

This function sorts people by birthplace surname and name with build-in sort Here is the call graph for this function:



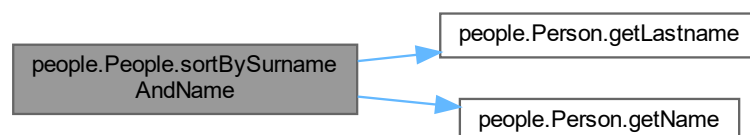
Here is the caller graph for this function:



5.3.2.22 sortBySurnameAndName()

```
void people.People.sortBySurnameAndName ( )
```

This function sorts people by surname and name with build-in sort Here is the call graph for this function:



Here is the caller graph for this function:



5.3.2.23 sortClasses()

```
void people.People.sortClasses ( )
```

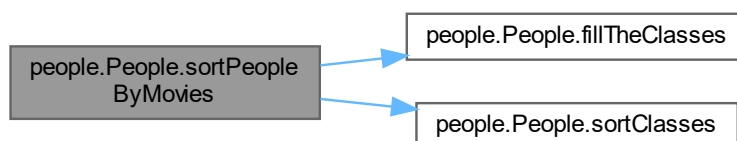
This function sorts the classes of movies by number of people who likes them Here is the caller graph for this function:



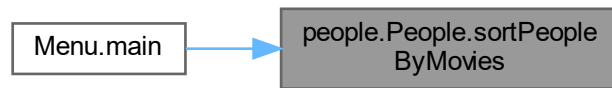
5.3.2.24 sortPeopleByMovies()

```
void people.People.sortPeopleByMovies ( )
```

This function sorts people by most common movies sets it creates the classes, sort them and then get sorted people from hashmap Here is the call graph for this function:



Here is the caller graph for this function:



5.3.2.25 writeToFile()

```
void people.People.writeToFile (
    String filename ) throws IOException
```

This method writes string to given file name.

Parameters

<i>filename</i>	string
-----------------	--------

Exceptions

<i>IOException</i>	This throws could occur when writing to a file
--------------------	--

Here is the call graph for this function:



Here is the caller graph for this function:



5.3.3 Member Data Documentation

5.3.3.1 FIRST_LINE_PEOPLE

```
final String people.People.FIRST_LINE_PEOPLE = "idperson,name,lastname,birthdate,gender,birthplace,home,studiedAt,workplaces,films,groupCode"
[static]
```

5.3.3.2 profiles

```
HashMap<List<String>, List<Person> > people.People.profiles = new HashMap<>()
```

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

- D:/Projekty/DSA_ehu_JavaProject/src/people/[People.java](#)

5.4 people.Person Class Reference

Public Member Functions

- boolean [equals](#) (Object o)
- int [hashCode](#) ()
- [Person](#) (String idPerson, String name, String lastname, Date birthdate, char gender, String birthplace, String home, List< String > studiedAt, List< String > workplaces, List< String > films, String groupCode)
- String [getIdPerson](#) ()
- void [setIdPerson](#) (String idPerson)
- String [getName](#) ()
- void [setName](#) (String name)
- String [getLastname](#) ()
- void [setLastname](#) (String lastname)
- Date [getBirthdate](#) ()
- void [setBirthdate](#) (Date birthdate)
- char [getGender](#) ()
- void [setGender](#) (char gender)
- String [getBirthplace](#) ()
- void [setBirthplace](#) (String birthplace)
- String [getHome](#) ()
- void [setHome](#) (String home)
- List< String > [getStudiedAt](#) ()
- void [setStudiedAt](#) (List< String > studiedAt)
- List< String > [getWorkplaces](#) ()
- void [setWorkplaces](#) (List< String > workplaces)
- List< String > [getFilms](#) ()
- void [setFilms](#) (List< String > films)
- String [getGroupCode](#) ()
- void [setGroupCode](#) (String groupCode)
- String [toString](#) ()

5.4.1 Constructor & Destructor Documentation

5.4.1.1 Person()

```
people.Person.Person (
    String idPerson,
    String name,
    String lastname,
    Date birthdate,
    char gender,
    String birthplace,
    String home,
    List< String > studiedAt,
    List< String > workplaces,
    List< String > films,
    String groupCode )
```

All-argument constructor for a [Person](#) class.

Parameters

<i>idPerson</i>	String contains user ID
<i>name</i>	String contains username
<i>lastname</i>	String contains lastname
<i>birthdate</i>	Date contains birthdate
<i>gender</i>	char contains 'M' as 'male' / 'F' as 'female'
<i>birthplace</i>	String contains birthplace
<i>home</i>	String contains name of hometown
<i>studiedAt</i>	list of Strings contains name of university
<i>workplaces</i>	list of Strings contains workplaces
<i>films</i>	list of Strings contains favourites movie titles
<i>groupCode</i>	String contains group code

5.4.2 Member Function Documentation

5.4.2.1 equals()

```
boolean people.Person.equals (
    Object o )
```

This method compares this object with object given in parameter. It overrides default method and allows us to use '==' operator.

Parameters

<i>o</i>	is an object of undefined type
----------	--------------------------------

Returns

boolean

Here is the call graph for this function:



Here is the caller graph for this function:

**5.4.2.2 getBirthdate()**

```
Date people.Person.getBirthdate ( )
```

Getter for a birthdate field.

Returns

Date type object.

5.4.2.3 getBirthplace()

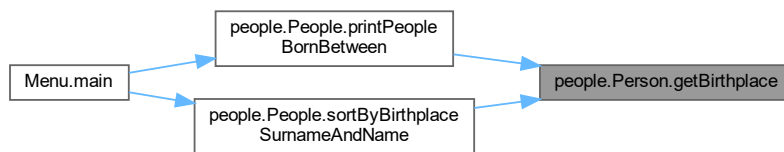
```
String people.Person.getBirthplace ( )
```

Getter for a birthplace field.

Returns

String type object.

Here is the caller graph for this function:



5.4.2.4 getFilms()

```
List< String > people.Person.getFilms ( )
```

Getter for a films field.

Returns

List of String type objects.

5.4.2.5 getGender()

```
char people.Person.getGender ( )
```

Getter for a gender field.

Returns

char type object.

5.4.2.6 getGroupCode()

```
String people.Person.getGroupCode ( )
```

Getter for a groupCode field.

Returns

String type object.

5.4.2.7 getHome()

```
String people.Person.getHome ( )
```

Getter for a getHome field.

Returns

String type object.

5.4.2.8 getIdPerson()

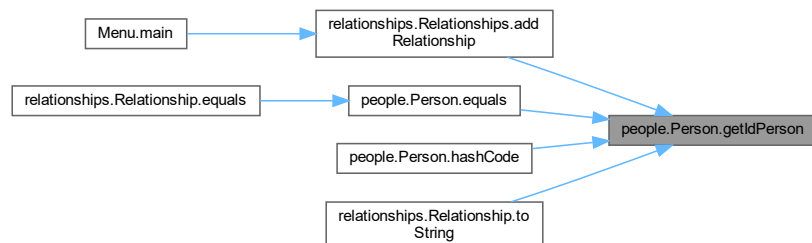
```
String people.Person.getIdPerson ( )
```

Getter for a idPerson field.

Returns

String type object.

Here is the caller graph for this function:



5.4.2.9 getLastname()

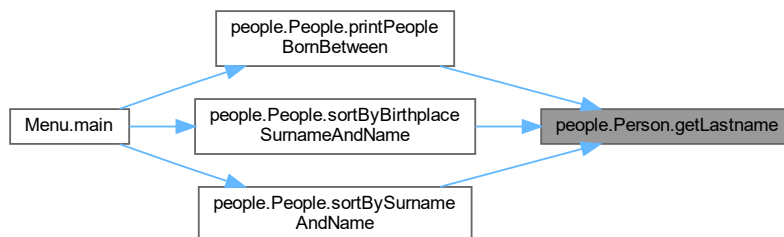
```
String people.Person.getLastname ( )
```

Getter for a lastname field.

Returns

String type object.

Here is the caller graph for this function:



5.4.2.10 getName()

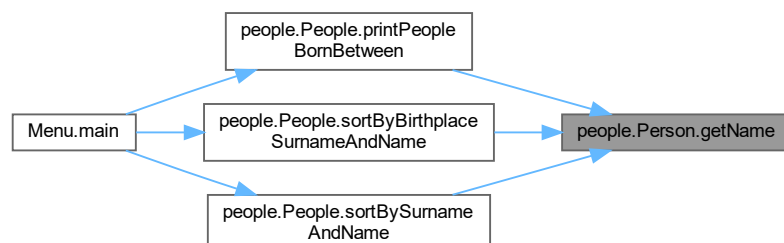
```
String people.Person.getName ( )
```

Getter for a name field.

Returns

String type object.

Here is the caller graph for this function:



5.4.2.11 `getStudiedAt()`

```
List< String > people.Person.getStudiedAt ( )
```

Getter for a studiedAt field.

Returns

List of String type objects.

5.4.2.12 `getWorkplaces()`

```
List< String > people.Person.getWorkplaces ( )
```

Getter for a workplaces field.

Returns

List of String type objects.

5.4.2.13 `hashCode()`

```
int people.Person.hashCode ( )
```

This method overrides default method and returns the integer hash code value of the object.

Returns

integer

Here is the call graph for this function:



5.4.2.14 `setBirthdate()`

```
void people.Person.setBirthdate (
    Date birthdate )
```

Setter for a birthdate field.

Parameters

<i>birthdate</i>	is a Date type object.
------------------	------------------------

5.4.2.15 setBirthplace()

```
void people.Person.setBirthplace (
    String birthplace )
```

Setter for a birthplace field.

Parameters

<i>birthplace</i>	is a String type object.
-------------------	--------------------------

5.4.2.16 setFilms()

```
void people.Person.setFilms (
    List< String > films )
```

Setter for a films field.

Parameters

<i>films</i>	is a list of String type object.
--------------	----------------------------------

5.4.2.17 setGender()

```
void people.Person.setGender (
    char gender )
```

Setter for a gender field.

Parameters

<i>gender</i>	is a char type object.
---------------	------------------------

5.4.2.18 setGroupCode()

```
void people.Person.setGroupCode (
    String groupCode )
```

Setter for a groupCode field.

Parameters

<i>groupCode</i>	is a String type object.
------------------	--------------------------

5.4.2.19 setHome()

```
void people.Person.setHome (
    String home )
```

Setter for a home field.

Parameters

<i>home</i>	is a String type object.
-------------	--------------------------

5.4.2.20 setIdPerson()

```
void people.Person.setIdPerson (
    String idPerson )
```

Setter for a idPerson field.

Parameters

<i>idPerson</i>	is a String type object.
-----------------	--------------------------

5.4.2.21 setLastname()

```
void people.Person.setLastname (
    String lastname )
```

Setter for a lastname field.

Parameters

<i>lastname</i>	is a String type object.
-----------------	--------------------------

5.4.2.22 setName()

```
void people.Person.setName (  
    String name )
```

Setter for a name field.

Parameters

<i>name</i>	is a String type object.
-------------	--------------------------

5.4.2.23 setStudiedAt()

```
void people.Person.setStudiedAt (  
    List< String > studiedAt )
```

Setter for a studiedAt field.

Parameters

<i>studied↔ At</i>	is a list of String type object.
------------------------	----------------------------------

5.4.2.24 setWorkplaces()

```
void people.Person.setWorkplaces (  
    List< String > workplaces )
```

Setter for a workplaces field.

Parameters

<i>workplaces</i>	is a list of String type object.
-------------------	----------------------------------

5.4.2.25 toString()

```
String people.Person.toString ( )
```

This method overrides default method and returns a string representation of this object.

Returns

string

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

- D:/Projekty/DSA_ehu_JavaProject/src/people/[Person.java](#)

5.5 people.Quicksort Class Reference

Static Public Member Functions

- static< T extends Comparable< T > void [sort](#) (List< [Person](#) > list)

5.5.1 Member Function Documentation

5.5.1.1 sort()

```
static< T extends Comparable< T > void people.Quicksort.sort (
    List< Person > list ) [static]
```

first method of sorting

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

- D:/Projekty/DSA_ehu_JavaProject/src/people/[Quicksort.java](#)

5.6 relationships.Relationship Class Reference

Public Member Functions

- String [toString](#) ()
- boolean [equals](#) (Object o)
- int [hashCode](#) ()
- [Relationship](#) ([Person](#) friend1, [Person](#) friend2)
- [Person](#) [getFriend1](#) ()
- void [setFriend1](#) ([Person](#) friend1)
- [Person](#) [getFriend2](#) ()
- void [setFriend2](#) ([Person](#) friend2)

5.6.1 Detailed Description

This class represents a single relationship between two people. Consists of two objects of type Person. @field friend1 an absolute URL giving the base location of the image @field friend2 the location of the image, relative to the url argument

5.6.2 Constructor & Destructor Documentation

5.6.2.1 Relationship()

```
relationships.Relationship.Relationship (
    Person friend1,
    Person friend2 )
```

Constructor for a [Relationships](#) class.

Parameters

<i>friend1</i>	is parameter of Person type.
<i>friend2</i>	is parameter of Person type.

5.6.3 Member Function Documentation

5.6.3.1 equals()

```
boolean relationships.Relationship.equals (
    Object o )
```

This method compares this object with object given in parameter. It overrides default method and allows us to use '==' operator.

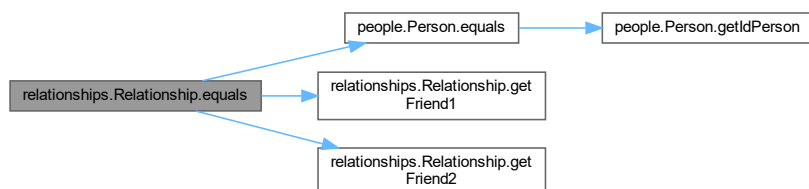
Parameters

<i>o</i>	is an object of undefined type
----------	--------------------------------

Returns

boolean

Here is the call graph for this function:

**5.6.3.2 getFriend1()**

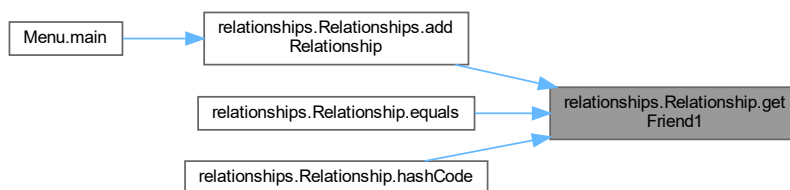
```
Person relationships.Relationship.getFriend1 ( )
```

Getter for a friend1 field.

Returns

object of Person type.

Here is the caller graph for this function:



5.6.3.3 getFriend2()

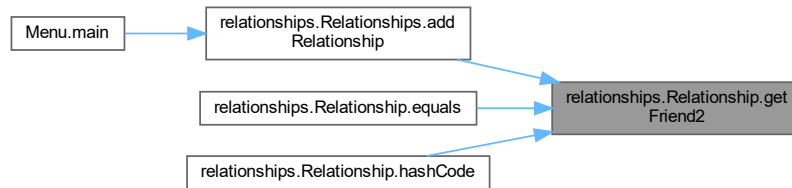
```
Person relationships.Relationship.getFriend2 ( )
```

Getter for a friend2 field.

Returns

object of Person type.

Here is the caller graph for this function:



5.6.3.4 hashCode()

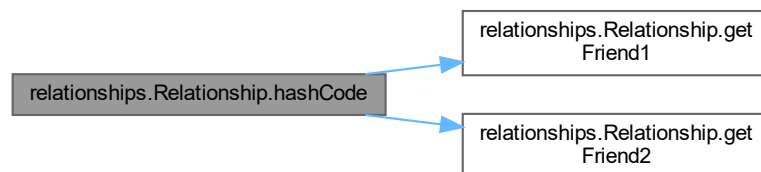
```
int relationships.Relationship.hashCode ( )
```

This method overrides default method and returns the integer hash code value of the object.

Returns

integer

Here is the call graph for this function:



5.6.3.5 setFriend1()

```
void relationships.Relationship.setFriend1 (
    Person friend1 )
```

Setter for a friend1 field.

Parameters

<i>friend1</i>	is parameter of Person type.
----------------	------------------------------

5.6.3.6 setFriend2()

```
void relationships.Relationship.setFriend2 (
    Person friend2 )
```

Setter for a friend2 field.

Parameters

<i>friend2</i>	is parameter of Person type.
----------------	------------------------------

5.6.3.7 toString()

```
String relationships.Relationship.toString ( )
```

This method returns a string representation of this object.

Returns

string

Here is the call graph for this function:



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

- [D:/Projekty/DSA_ehu_JavaProject/src/relationships/Relationship.java](#)

5.7 relationships.Relationships Class Reference

Public Member Functions

- [Relationships](#) ()
- List< [Relationship](#) > [getRelations](#) ()
- void [addRelationship](#) (String list, [People](#) ppl)
- String [toString](#) ()
- void [writeToFile](#) (String filename) throws IOException
- List< [Relationship](#) > [findRelationshipsById](#) (String id)
- void [deleteRelationship](#) ([Relationship](#) r)
- [Relationship](#) [findRelationshipsByIds](#) (String id1, String id2)
- void [printAllRelationships](#) ()

Static Public Attributes

- static final String [FIRST_LINE_RELATIONSHIPS](#) = "friend1,friend2\n"

5.7.1 Constructor & Destructor Documentation

5.7.1.1 Relationships()

```
relationships.Relationships.Relationships ( )
```

Constructor for a [Relationships](#) class.

5.7.2 Member Function Documentation

5.7.2.1 addRelationship()

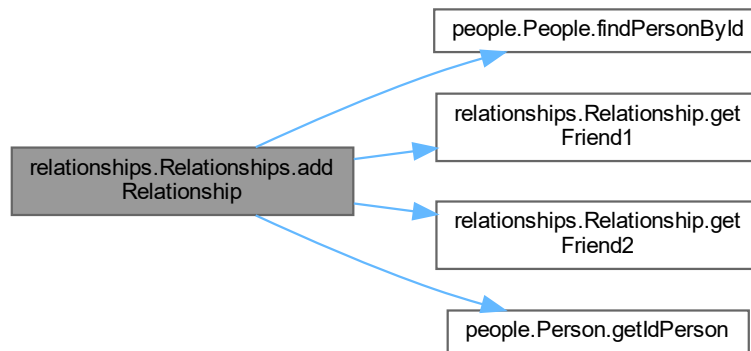
```
void relationships.Relationships.addRelationship (
    String list,
    People ppl )
```

This method check if relationship for two people exists. If not - create relationship between two people and add this relationship to list of relationships.

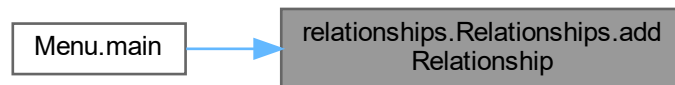
Parameters

<i>list</i>	This parameter is string list which consists two user IDs.
<i>ppl</i>	This parameter is People type object which have list of all users

Here is the call graph for this function:



Here is the caller graph for this function:



5.7.2.2 deleteRelationship()

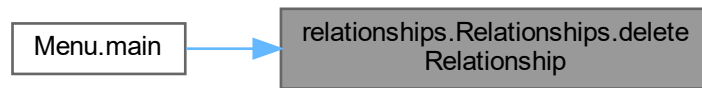
```
void relationships.Relationships.deleteRelationship (
    Relationship r )
```

Removes passed relationship from the list of relationships

Parameters

<i>r</i>	Relation
----------	----------

Here is the caller graph for this function:



5.7.2.3 findRelationshipsById()

```
List< Relationship > relationships.Relationships.findRelationshipsById (
    String id )
```

This method finds every relationship containing passed id

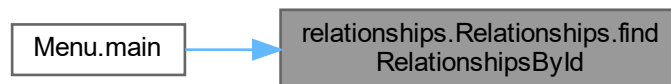
Parameters

<i>id</i>	String
-----------	--------

Returns

List<Relationship> List of relationships containing that id

Here is the caller graph for this function:



5.7.2.4 findRelationshipsByIds()

```
Relationship relationships.Relationships.findRelationshipsByIds (
    String id1,
    String id2 )
```

This method finds relationship containing passed IDs

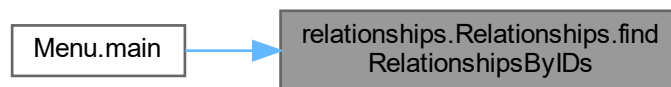
Parameters

<i>id1</i>	String
<i>id2</i>	String

Returns

[Relationship](#) containing passed IDs

Here is the caller graph for this function:

**5.7.2.5 getRelations()**

```
List< Relationship > relationships.Relationships.getRelations ( )
```

Getter for a relations field.

Returns

list of [Relationship](#) objects

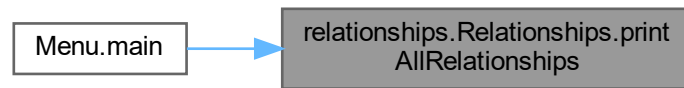
5.7.2.6 printAllRelationships()

```
void relationships.Relationships.printAllRelationships ( )
```

This method prints out every relationship stored in the program Here is the call graph for this function:



Here is the caller graph for this function:



5.7.2.7 toString()

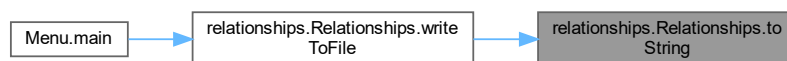
```
String relationships.Relationships.toString ( )
```

This method overrides default method and returns a string representation of this object.

Returns

string

Here is the caller graph for this function:



5.7.2.8 writeToFile()

```
void relationships.Relationships.writeToFile (
    String filename ) throws IOException
```

This method writes string to given file name.

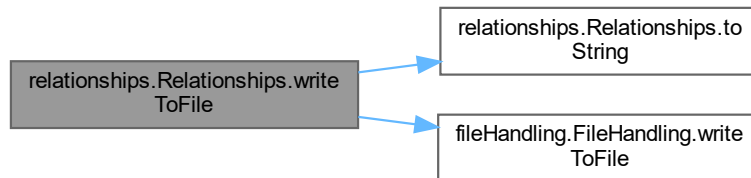
Parameters

<i>filename</i>	string
-----------------	--------

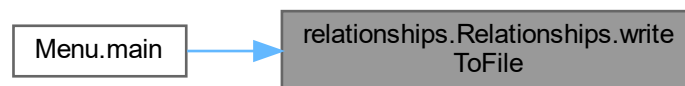
Exceptions

<i>IOException</i>	This throws could occur when writing to a file
--------------------	--

Here is the call graph for this function:



Here is the caller graph for this function:

**5.7.3 Member Data Documentation****5.7.3.1 FIRST_LINE_RELATIONSHIPS**

```
final String relationships.Relationships.FIRST_LINE_RELATIONSHIPS = "friend1,friend2\n" [static]
```

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

- D:/Projekty/DSA_ehu_JavaProject/src/relationships/[Relationships.java](#)

5.8 UnitTests Class Reference**Classes**

- class **FileHandlingTest**
- class **MenuTests**
- class **PeopleTest**
- class **PersonTest**
- class **RelationshipsTest**

Public Member Functions

- void [done](#) ()

5.8.1 Member Function Documentation

5.8.1.1 done()

```
void UnitTests.done ( )
```

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

- D:/Projekty/DSA_ehu_JavaProject/src/[UnitTests.java](#)

Chapter 6

File Documentation

6.1 D:/Projekty/DSA_ehu_JavaProject/src/fileHandling/FileHandling.java File Reference

Classes

- class [fileHandling.FileHandling](#)

Packages

- package [fileHandling](#)

6.2 D:/Projekty/DSA_ehu_JavaProject/src/Menu.java File Reference

Classes

- class [Menu](#)

6.3 D:/Projekty/DSA_ehu_JavaProject/src/people/People.java File Reference

Classes

- class [people.People](#)

Packages

- package [people](#)

6.4 D:/Projekty/DSA_ehu_JavaProject/src/people/Person.java File Reference

Classes

- class [people.Person](#)

Packages

- package [people](#)

6.5 D:/Projekty/DSA_ehu_JavaProject/src/people/Quicksort.java File Reference

Classes

- class [people.Quicksort](#)

Packages

- package [people](#)

6.6 D:/Projekty/DSA_ehu_JavaProject/src/relationships/↔ Relationship.java File Reference

Classes

- class [relationships.Relationship](#)

Packages

- package [relationships](#)

6.7 D:/Projekty/DSA_ehu_JavaProject/src/relationships/↔ Relationships.java File Reference

Classes

- class [relationships.Relationships](#)

Packages

- package [relationships](#)

6.8 D:/Projekty/DSA_ehu_JavaProject/src/UnitTests.java File Reference

Classes

- class [UnitTests](#)
- class **UnitTests.MenuTests**
- class **UnitTests.FileHandlingTest**
- class **UnitTests.PeopleTest**
- class **UnitTests.PersonTest**
- class **UnitTests.RelationshipsTest**
- class **UnitTests.RelationshipsTest.RelationTest**

Index

addPersonFromString
 people.People, 14

addRelationship
 relationships.Relationships, 45

anySymbolLineFormatter
 people.People, 15

D:/Projekty/Dsa_ehu_JavaProject/src/fileHandling/FileHandling.java, 53

D:/Projekty/Dsa_ehu_JavaProject/src/Menu.java, 53

D:/Projekty/Dsa_ehu_JavaProject/src/people/People.java, getBirthdate
53

D:/Projekty/Dsa_ehu_JavaProject/src/people/Person.java, getBirthplace
54

D:/Projekty/Dsa_ehu_JavaProject/src/people/Quicksort.java, getFilms
54

D:/Projekty/Dsa_ehu_JavaProject/src/relationships/Relationships.java, getFriends
54

D:/Projekty/Dsa_ehu_JavaProject/src/relationships/Relationships.java, getFriends
54

D:/Projekty/Dsa_ehu_JavaProject/src/UnitTests.java, 55

dashLineFormatter
 people.People, 15

deleteRelationship
 relationships.Relationships, 46

done
 UnitTests, 51

equals
 people.Person, 31
 relationships.Relationship, 41

fileHandling, 7

fileHandling.FileHandling, 9
 readFile, 9
 writeToFile, 10

fillTheClasses
 people.People, 16

findMatchesByFile
 people.People, 16

findPeopleBornBetween
 people.People, 17

findPeopleByBirthplace
 people.People, 18

findPeopleByHome
 people.People, 18

findPersonById
 people.People, 19

findPersonBySurname
 people.People, 20

findRelationshipsById
 relationships.Relationships, 47

findRelationshipsByIds
 relationships.Relationships, 47

FIRST_LINE_PEOPLE
 people.People, 30

firstLineRelationships
 relationships.Relationships, 50

getGender
 people.Person, 32

getGroupCode
 people.Person, 32

getHome
 people.Person, 33

getIdPerson
 people.Person, 33

getLastName
 people.Person, 34

getName
 people.Person, 35

getPeople
 people.People, 20

getRelations
 relationships.Relationships, 48

getStudiedAt
 people.Person, 35

getWorkplaces
 people.Person, 36

hashCode
 people.Person, 36
 relationships.Relationship, 43

main
 Menu, 11

Menu, 11
 main, 11
 menu, 12

- menu
 - Menu, [12](#)
- People
 - people.People, [14](#)
- people, [7](#)
- people.People, [13](#)
 - addPersonFromString, [14](#)
 - anySymbolLineFormatter, [15](#)
 - dashLineFormatter, [15](#)
 - fillTheClasses, [16](#)
 - findMatchesByFile, [16](#)
 - findPeopleBornBetween, [17](#)
 - findPeopleByBirthplace, [18](#)
 - findPeopleByHome, [18](#)
 - findPersonById, [19](#)
 - findPersonBySurname, [20](#)
 - FIRST_LINE_PEOPLE, [30](#)
 - getPeople, [20](#)
 - People, [14](#)
 - printAllClasses, [20](#)
 - printAllPeople, [21](#)
 - printPeopleBornBetween, [21](#)
 - printPeopleByBirthplace, [22](#)
 - printPeopleByHome, [23](#)
 - printRelationshipsByLastname, [24](#)
 - profiles, [30](#)
 - quicksortByBirthplaceSurnameAndName, [25](#)
 - removePersonById, [25](#)
 - setPeople, [26](#)
 - sortByBirthplaceSurnameAndName, [26](#)
 - sortBySurnameAndName, [27](#)
 - sortClasses, [28](#)
 - sortPeopleByMovies, [28](#)
 - writeToFile, [29](#)
- people.Person, [30](#)
 - equals, [31](#)
 - getBirthdate, [32](#)
 - getBirthplace, [32](#)
 - getFilms, [33](#)
 - getGender, [33](#)
 - getGroupCode, [33](#)
 - getHome, [34](#)
 - getIdPerson, [34](#)
 - getLastname, [34](#)
 - getName, [35](#)
 - getStudiedAt, [35](#)
 - getWorkplaces, [36](#)
 - hashCode, [36](#)
 - Person, [31](#)
 - setBirthdate, [36](#)
 - setBirthplace, [37](#)
 - setFilms, [37](#)
 - setGender, [37](#)
 - setGroupCode, [37](#)
 - setHome, [38](#)
 - setIdPerson, [38](#)
 - setLastname, [38](#)
 - setName, [39](#)
 - setStudiedAt, [39](#)
 - setWorkplaces, [39](#)
 - toString, [39](#)
- people.Quicksort, [40](#)
 - sort, [40](#)
- Person
 - people.Person, [31](#)
- printAllClasses
 - people.People, [20](#)
- printAllPeople
 - people.People, [21](#)
- printAllRelationships
 - relationships.Relationships, [48](#)
- printPeopleBornBetween
 - people.People, [21](#)
- printPeopleByBirthplace
 - people.People, [22](#)
- printPeopleByHome
 - people.People, [23](#)
- printRelationshipsByLastname
 - people.People, [24](#)
- profiles
 - people.People, [30](#)
- quicksortByBirthplaceSurnameAndName
 - people.People, [25](#)
- readFile
 - fileHandling.FileHandling, [9](#)
- Relationship
 - relationships.Relationship, [41](#)
- Relationships
 - relationships.Relationships, [45](#)
- relationships, [7](#)
- relationships.Relationship, [40](#)
 - equals, [41](#)
 - getFriend1, [42](#)
 - getFriend2, [42](#)
 - hashCode, [43](#)
 - Relationship, [41](#)
 - setFriend1, [43](#)
 - setFriend2, [44](#)
 - toString, [44](#)
- relationships.Relationships, [45](#)
 - addRelationship, [45](#)
 - deleteRelationship, [46](#)
 - findRelationshipsById, [47](#)
 - findRelationshipsByIds, [47](#)
 - FIRST_LINE_RELATIONSHIPS, [50](#)
 - getRelations, [48](#)
 - printAllRelationships, [48](#)
 - Relationships, [45](#)
 - toString, [49](#)
 - writeToFile, [49](#)
- removePersonById
 - people.People, [25](#)
- setBirthdate
 - people.Person, [36](#)

- setBirthplace
 - people.Person, [37](#)
- setFilms
 - people.Person, [37](#)
- setFriend1
 - relationships.Relationship, [43](#)
- setFriend2
 - relationships.Relationship, [44](#)
- setGender
 - people.Person, [37](#)
- setGroupCode
 - people.Person, [37](#)
- setHome
 - people.Person, [38](#)
- setIdPerson
 - people.Person, [38](#)
- setLastname
 - people.Person, [38](#)
- setName
 - people.Person, [39](#)
- setPeople
 - people.People, [26](#)
- setStudiedAt
 - people.Person, [39](#)
- setWorkplaces
 - people.Person, [39](#)
- sort
 - people.Quicksort, [40](#)
- sortByBirthplaceSurnameAndName
 - people.People, [26](#)
- sortBySurnameAndName
 - people.People, [27](#)
- sortClasses
 - people.People, [28](#)
- sortPeopleByMovies
 - people.People, [28](#)
- toString
 - people.Person, [39](#)
 - relationships.Relationship, [44](#)
 - relationships.Relationships, [49](#)
- UnitTests, [50](#)
 - done, [51](#)
- writeToFile
 - fileHandling.FileHandling, [10](#)
 - people.People, [29](#)
 - relationships.Relationships, [49](#)