

## My Project

Generated by Doxygen 1.9.5



<b>1 Namespace Index</b>	<b>1</b>
1.1 Package List . . . . .	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 Namespace Documentation</b>	<b>7</b>
4.1 Package fileHandling . . . . .	7
4.2 Package people . . . . .	7
4.3 Package relationships . . . . .	7
<b>5 Class Documentation</b>	<b>9</b>
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 . . . . .	10
5.2.1 Detailed Description . . . . .	10
5.2.2 Member Function Documentation . . . . .	10
5.2.2.1 main() . . . . .	10
5.2.2.2 menu() . . . . .	11
5.3 people.People Class Reference . . . . .	11
5.3.1 Constructor & Destructor Documentation . . . . .	11
5.3.1.1 People() . . . . .	11
5.3.2 Member Function Documentation . . . . .	11
5.3.2.1 addPersonFromString() . . . . .	11
5.3.2.2 findPersonById() . . . . .	12
5.3.2.3 getPeople() . . . . .	12
5.3.2.4 setPeople() . . . . .	12
5.3.2.5 writeToFile() . . . . .	13
5.3.3 Member Data Documentation . . . . .	13
5.3.3.1 FIRST_LINE_PEOPLE . . . . .	13
5.4 people.Person Class Reference . . . . .	13
5.4.1 Constructor & Destructor Documentation . . . . .	14
5.4.1.1 Person() . . . . .	14
5.4.2 Member Function Documentation . . . . .	15
5.4.2.1 equals() . . . . .	15
5.4.2.2 getBirthdate() . . . . .	15
5.4.2.3 getBirthplace() . . . . .	15
5.4.2.4 getFilms() . . . . .	16

5.4.2.5	getGender()	16
5.4.2.6	getGroupCode()	16
5.4.2.7	getHome()	16
5.4.2.8	getIdPerson()	17
5.4.2.9	getLastname()	17
5.4.2.10	getName()	17
5.4.2.11	getStudiedAt()	17
5.4.2.12	getWorkplaces()	18
5.4.2.13	hashCode()	18
5.4.2.14	setBirthdate()	18
5.4.2.15	setBirthplace()	18
5.4.2.16	setFilms()	19
5.4.2.17	setGender()	19
5.4.2.18	setGroupCode()	19
5.4.2.19	setHome()	19
5.4.2.20	setIdPerson()	20
5.4.2.21	setLastname()	20
5.4.2.22	setName()	20
5.4.2.23	setStudiedAt()	21
5.4.2.24	setWorkplaces()	21
5.4.2.25	toString()	21
5.5	relationships.Relationship Class Reference	21
5.5.1	Detailed Description	22
5.5.2	Constructor & Destructor Documentation	22
5.5.2.1	Relationship()	22
5.5.3	Member Function Documentation	22
5.5.3.1	equals()	22
5.5.3.2	getFriend1()	23
5.5.3.3	getFriend2()	23
5.5.3.4	hashCode()	23
5.5.3.5	setFriend1()	23
5.5.3.6	setFriend2()	24
5.5.3.7	toString()	24
5.6	relationships.Relationships Class Reference	24
5.6.1	Constructor & Destructor Documentation	24
5.6.1.1	Relationships()	25
5.6.2	Member Function Documentation	25
5.6.2.1	addRelationship()	25
5.6.2.2	getRelations()	25
5.6.2.3	toString()	25
5.6.2.4	writeToFile()	26
5.6.3	Member Data Documentation	27

---

5.6.3.1 FIRST_LINE_RELATIONSHIPS . . . . .	27
5.7 UnitTests Class Reference . . . . .	27
5.7.1 Member Function Documentation . . . . .	27
5.7.1.1 done() . . . . .	27
<b>6 File Documentation</b>	<b>29</b>
6.1 D:/Projekty/DSA_ehu_JavaProject/src/fileHandling/FileHandling.java File Reference . . . . .	29
6.2 D:/Projekty/DSA_ehu_JavaProject/src/Menu.java File Reference . . . . .	29
6.3 D:/Projekty/DSA_ehu_JavaProject/src/people/People.java File Reference . . . . .	29
6.4 D:/Projekty/DSA_ehu_JavaProject/src/people/Person.java File Reference . . . . .	30
6.5 D:/Projekty/DSA_ehu_JavaProject/src/relationships/Relationship.java File Reference . . . . .	30
6.6 D:/Projekty/DSA_ehu_JavaProject/src/relationships/Relationships.java File Reference . . . . .	30
6.7 D:/Projekty/DSA_ehu_JavaProject/src/tests/UnitTests.java File Reference . . . . .	30
<b>Index</b>	<b>31</b>



# Chapter 1

## Namespace Index

### 1.1 Package List

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

<a href="#">fileHandling</a>	.....	<a href="#">7</a>
<a href="#">people</a>	.....	<a href="#">7</a>
<a href="#">relationships</a>	.....	<a href="#">7</a>





## Chapter 2

# Class Index

### 2.1 Class List

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

<a href="#">fileHandling.FileHandling</a>	9
<a href="#">Menu</a>	10
<a href="#">people.People</a>	11
<a href="#">people.Person</a>	13
<a href="#">relationships.Relationship</a>	21
<a href="#">relationships.Relationships</a>	24
<a href="#">UnitTests</a>	27



## 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 . . . . .	29
D:/Projekty/Dsa_ehu_JavaProject/src/fileHandling/FileHandling.java . . . . .	29
D:/Projekty/Dsa_ehu_JavaProject/src/people/People.java . . . . .	29
D:/Projekty/Dsa_ehu_JavaProject/src/people/Person.java . . . . .	30
D:/Projekty/Dsa_ehu_JavaProject/src/relationships/Relationship.java . . . . .	30
D:/Projekty/Dsa_ehu_JavaProject/src/relationships/Relationships.java . . . . .	30
D:/Projekty/Dsa_ehu_JavaProject/src/tests/UnitTests.java . . . . .	30



## Chapter 4

# Namespace Documentation

### 4.1 Package fileHandling

#### Classes

- class [FileHandling](#)

### 4.2 Package people

#### Classes

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

### 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

### 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.
--------------------	--

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

<i>args</i>	
-------------	--

## 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

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 [addPersonFromString](#) (String data) throws ParseException
- [Person](#) [findPersonById](#) (String id)
- void [writeToFile](#) (String filename) throws IOException

### 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 <a href="#">Person</a> class.
-------------	--

**Exceptions**

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

**5.3.2.2 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

**5.3.2.3 getPeople()**

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

Getter for a people field.

**Returns**

list of [Person](#) type objects.

**5.3.2.4 setPeople()**

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

Setter for a people field.

## Parameters

<i>people</i>	is list of <a href="#">Person</a> type objects.
---------------	---

## 5.3.2.5 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
--------------------	--

## 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]
```

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

### 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.

#### 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.

#### 5.4.2.9 getLastName()

```
String people.Person.getLastName ( )
```

Getter for a lastname field.

##### Returns

String type object.

#### 5.4.2.10 getName()

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

Getter for a name field.

##### Returns

String type object.

#### 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

#### 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>studiedAt</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 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)

### 5.5.1 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

### 5.5.2 Constructor & Destructor Documentation

#### 5.5.2.1 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.

### 5.5.3 Member Function Documentation

#### 5.5.3.1 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

### 5.5.3.2 getFriend1()

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

Getter for a friend1 field.

#### Returns

object of Person type.

### 5.5.3.3 getFriend2()

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

Getter for a friend2 field.

#### Returns

object of Person type.

### 5.5.3.4 hashCode()

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

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

#### Returns

integer

### 5.5.3.5 setFriend1()

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

Setter for a friend1 field.

#### Parameters

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

### 5.5.3.6 setFriend2()

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

Setter for a friend2 field.

#### Parameters

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

### 5.5.3.7 toString()

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

This method 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/relationships/[Relationship.java](#)

## 5.6 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

### Static Public Attributes

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

### 5.6.1 Constructor & Destructor Documentation

### 5.6.1.1 Relationships()

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

Constructor for a [Relationships](#) class.

## 5.6.2 Member Function Documentation

### 5.6.2.1 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

### 5.6.2.2 getRelations()

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

Getter for a relations field.

#### Returns

list of [Relationship](#) objects

### 5.6.2.3 toString()

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

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

#### Returns

string

#### 5.6.2.4 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
--------------------	--

### 5.6.3 Member Data Documentation

#### 5.6.3.1 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](#)

## 5.7 UnitTests Class Reference

### Classes

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

### Public Member Functions

- void [done](#) ()

#### 5.7.1 Member Function Documentation

##### 5.7.1.1 done()

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

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

- D:/Projekty/DSA\_ehu\_JavaProject/src/tests/[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/relationships/↵ Relationship.java File Reference

### Classes

- class [relationships.Relationship](#)

### Packages

- package [relationships](#)

## 6.6 D:/Projekty/DSA\_ehu\_JavaProject/src/relationships/↵ Relationships.java File Reference

### Classes

- class [relationships.Relationships](#)

### Packages

- package [relationships](#)

## 6.7 D:/Projekty/DSA\_ehu\_JavaProject/src/tests/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, 11
- addRelationship
  - relationships.Relationships, 25
- D:/Projekty/DSA\_ehu\_JavaProject/src/fileHandling/FileHandling.java, 29
- D:/Projekty/DSA\_ehu\_JavaProject/src/Menu.java, 29
- D:/Projekty/DSA\_ehu\_JavaProject/src/people/People.java, 29
- D:/Projekty/DSA\_ehu\_JavaProject/src/people/Person.java, 30
- D:/Projekty/DSA\_ehu\_JavaProject/src/relationships/Relationship.java, 30
- D:/Projekty/DSA\_ehu\_JavaProject/src/relationships/Relationships.java, 30
- D:/Projekty/DSA\_ehu\_JavaProject/src/tests/UnitTests.java, 30
- done
  - UnitTests, 27
- equals
  - people.Person, 15
  - relationships.Relationship, 22
- fileHandling, 7
- fileHandling.FileHandling, 9
  - readFile, 9
  - writeToFile, 9
- findPersonById
  - people.People, 12
- FIRST\_LINE\_PEOPLE
  - people.People, 13
- FIRST\_LINE\_RELATIONSHIPS
  - relationships.Relationships, 27
- getBirthdate
  - people.Person, 15
- getBirthplace
  - people.Person, 15
- getFilms
  - people.Person, 15
- getFriend1
  - relationships.Relationship, 22
- getFriend2
  - relationships.Relationship, 23
- getGender
  - people.Person, 16
- getGroupCode
  - people.Person, 16
- getHome
  - people.Person, 16
- getIdPerson
  - people.Person, 16
- getLastName
  - people.Person, 17
- getName
  - people.Person, 17
- getPeople
  - people.People, 12
- getRelations
  - relationships.Relationships, 25
- getSurname
  - people.Person, 17
- getWorkplaces
  - people.Person, 17
- hashCode
  - people.Person, 18
  - relationships.Relationship, 23
- main
  - Menu, 10
- Menu, 10
  - main, 10
  - menu, 11
- menu
  - Menu, 11
- People
  - people.People, 11
- people, 7
  - people.People, 11
  - addPersonFromString, 11
  - findPersonById, 12
  - FIRST\_LINE\_PEOPLE, 13
  - getPeople, 12
  - People, 11
  - setPeople, 12
  - writeToFile, 13
- people.Person, 13
  - equals, 15
  - getBirthdate, 15
  - getBirthplace, 15
  - getFilms, 15
  - getGender, 16
  - getGroupCode, 16
  - getHome, 16
  - getIdPerson, 16
  - getLastName, 17

- getName, [17](#)
  - getStudiedAt, [17](#)
  - getWorkplaces, [17](#)
  - hashCode, [18](#)
  - Person, [14](#)
  - setBirthdate, [18](#)
  - setBirthplace, [18](#)
  - setFilms, [19](#)
  - setGender, [19](#)
  - setGroupCode, [19](#)
  - setHome, [19](#)
  - setIdPerson, [20](#)
  - setLastname, [20](#)
  - setName, [20](#)
  - setStudiedAt, [20](#)
  - setWorkplaces, [21](#)
  - toString, [21](#)
- Person
  - people.Person, [14](#)
- readFile
  - fileHandling.FileHandling, [9](#)
- Relationship
  - relationships.Relationship, [22](#)
- Relationships
  - relationships.Relationships, [24](#)
- relationships, [7](#)
- relationships.Relationship, [21](#)
  - equals, [22](#)
  - getFriend1, [22](#)
  - getFriend2, [23](#)
  - hashCode, [23](#)
  - Relationship, [22](#)
  - setFriend1, [23](#)
  - setFriend2, [23](#)
  - toString, [24](#)
- relationships.Relationships, [24](#)
  - addRelationship, [25](#)
  - FIRST\_LINE\_RELATIONSHIPS, [27](#)
  - getRelations, [25](#)
  - Relationships, [24](#)
  - toString, [25](#)
  - writeToFile, [25](#)
- setBirthdate
  - people.Person, [18](#)
- setBirthplace
  - people.Person, [18](#)
- setFilms
  - people.Person, [19](#)
- setFriend1
  - relationships.Relationship, [23](#)
- setFriend2
  - relationships.Relationship, [23](#)
- setGender
  - people.Person, [19](#)
- setGroupCode
  - people.Person, [19](#)
- setHome
  - people.Person, [19](#)
- setIdPerson
  - people.Person, [20](#)
- setLastname
  - people.Person, [20](#)
- setName
  - people.Person, [20](#)
- setPeople
  - people.People, [12](#)
- setStudiedAt
  - people.Person, [20](#)
- setWorkplaces
  - people.Person, [21](#)
- toString
  - people.Person, [21](#)
  - relationships.Relationship, [24](#)
  - relationships.Relationships, [25](#)
- UnitTests, [27](#)
  - done, [27](#)
- writeToFile
  - fileHandling.FileHandling, [9](#)
  - people.People, [13](#)
  - relationships.Relationships, [25](#)