

Programowanie komponentowe

**Sprawozdanie z pracy projektowej**

# **Sudoku**

Igor Oryński 203959

Dawid Urbaniak 204023

**Informatyka, sem. IV**

**2016/ 2017**

# 1. Najważniejsze klasy warstwy danych projektu

## 1.1 SudokulItem

sudokupack

### Class SudokulItem

java.lang.Object  
sudokupack.SudokulItem

**Direct Known Subclasses:**

SudokuBox, SudokuColumn, SudokuRow

```
public abstract class SudokulItem  
extends java.lang.Object
```

SudokulItem is an abstract class that represents items which sudoku contains

#### Field Summary

##### Fields

Modifier and Type	Field and Description
protected java.util.ArrayList<SudokuField>	<b>values</b> values which represents row in sudoku

#### Constructor Summary

##### Constructors

Constructor and Description
<b>SudokulItem</b> (java.util.ArrayList<SudokuField> values) Constructor of SudokulItem

## Method Summary

All Methods	Instance Methods	Concrete Methods
Modifier and Type	Method and Description	
boolean	<code>equals(java.lang.Object obj)</code>	
int	<code>hashCode()</code>	
boolean	<code>isSolved()</code> isSolved is a method that checks if values contains only unique values	
java.lang.String	<code>toString()</code>	
boolean	<code>verify()</code> verify is a method that checks if values contains only unique values	

### Methods inherited from class java.lang.Object

`clone, finalize, getClass, notify, notifyAll, wait, wait, wait`

## Field Detail

### values

`protected java.util.ArrayList<SudokuField> values`

values which represents row in sudoku

## Constructor Detail

### SudokuItem

`public SudokuItem(java.util.ArrayList<SudokuField> values)`

Constructor of SudokuItem

#### Parameters:

values - values is List of SudokuFields which represents row in Sudoku

## Method Detail

### verify

`public boolean verify()`

verify is a method that checks if values contains only unique values

#### Returns:

true if contains unique values, false if contains duplicated values (excluding zeroes)

### isSolved

`public boolean isSolved()`

isSolved is a method that checks if values contains only unique values

#### Returns:

true if contains unique values, false if contains duplicated values (including zeroes)

## 1.2 SudokuField

sudokupack

### Class SudokuField

java.lang.Object  
sudokupack.SudokuField

All Implemented Interfaces:

java.io.Serializable, java.lang.Cloneable, java.lang.Comparable

@Entity

```
public class SudokuField
extends java.lang.Object
implements java.io.Serializable, java.lang.Cloneable, java.lang.Comparable
```

Class SudokuField represents one field of the sudoku grid.

See Also:

Serialized Form

#### Constructor Summary

##### Constructors

##### Constructor and Description

**SudokuField()**

Parameterless constructor.

**SudokuField(int value)**

Constructor with 1 int parameter

#### Method Summary

##### All Methods

##### Instance Methods

##### Concrete Methods

##### Modifier and Type

##### Method and Description

protected java.lang.Object

**clone()**

int

**compareTo(java.lang.Object obj)**

boolean

**equals(java.lang.Object obj)**

java.lang.Boolean

**getChangeable()**

**SudokuBoard**

**getSudokuBoard()**

int

**getValue()**

int

**hashCode()**

void

**setChangeable(java.lang.Boolean changeable)**

void

**setSudokuBoard(SudokuBoard sudokuBoard)**

void

**setValue(int value)**

java.lang.String

**toString()**

##### Methods inherited from class java.lang.Object

finalize, getClass, notify, notifyAll, wait, wait, wait

## Constructor Detail

### SudokuField

```
public SudokuField()
```

Parameterless constructor. Sets value to 0 and isChangeable to false.

### SudokuField

```
public SudokuField(int value)
```

Constructor with 1 int parameter

#### Parameters:

value - value of SudokuField

## Method Detail

### getSudokuBoard

```
public SudokuBoard getSudokuBoard()
```

### setSudokuBoard

```
public void setSudokuBoard(SudokuBoard sudokuBoard)
```

### getValue

```
public int getValue()
```

### setValue

```
public void setValue(int value)
```

### getChangeable

```
public java.lang.Boolean getChangeable()
```

### setChangeable

```
public void setChangeable(java.lang.Boolean changeable)
```

### clone

```
protected java.lang.Object clone()  
    throws java.lang.CloneNotSupportedException
```

#### Overrides:

clone in class java.lang.Object

#### Throws:

java.lang.CloneNotSupportedException

# 1.3 SudokuBoard

sudokupack

## Class SudokuBoard

java.lang.Object  
sudokupack.SudokuBoard

All Implemented Interfaces:

java.io.Serializable, java.lang.Cloneable

```
@Entity
public class SudokuBoard
extends java.lang.Object
implements java.io.Serializable, java.lang.Cloneable
```

SudokuBoard is a class that represents Sudoku

See Also:

Serialized Form

### Constructor Summary

#### Constructors

##### Constructor and Description

###### SudokuBoard()

Parameterless constructor.

### Method Summary

All Methods	Instance Methods	Concrete Methods
Modifier and Type		Method and Description
void		<code>cleanRandomlyFields(int amount)</code> Sets 0 value in randomly choosen SudokuFields
protected java.lang.Object		<code>clone()</code>
void		<code>convert1dto2d()</code> convert1dto2d converts List of SudokuFields to List of List of SudokuFields and sets it in field variable
void		<code>convert2dto1d()</code> convert2dto1d converts List of List of SudokuFields to List of SudokuFields and sets it in linearBoard variable
boolean		<code>equals(java.lang.Object obj)</code>
java.util.ArrayList<java.util.ArrayList<SudokuField>>		<code>getAll()</code> field getter.
SudokuBox		<code>getBox(int row, int column)</code> getBox returns SudokuBox
SudokuColumn		<code>getColumn(int column)</code> getColumn returns SudokuColumn
SudokuField		<code>getField(int row, int col)</code> getField returns SudokuField
java.lang.String		<code>getId()</code>
SudokuRow		<code>getRow(int row)</code> setRow returns SudokuRow

int	<b>getValue</b> (int row, int col) getValue returns SudokuField value
int	<b>hashCode</b> ()
boolean	<b>isSolved</b> () isSolved checks if Sudoku has been solved correctly
java.lang.String	<b>print</b> ()
void	<b>setId</b> (java.lang.String id)
void	<b>setValue</b> (int row, int col, int value) setValue sets value in the SudokuField
java.lang.String	<b>toString</b> ()

#### Methods inherited from class java.lang.Object

finalize, getClass, notify, notifyAll, wait, wait, wait

### Method Detail

#### getAll

```
public java.util.ArrayList<java.util.ArrayList<SudokuField>> getAll()
```

field getter.

**Returns:**

field

#### getValue

```
public int getValue(int row,
                   int col)
```

getValue returns SudokuField value

**Parameters:**

row - row of the field

col - column of the field

**Returns:**

SudokuField value

#### getField

```
public SudokuField getField(int row,
                           int col)
```

getField returns SudokuField

**Parameters:**

row - row of the field

col - column of the field

**Returns:**

SudokuField

### setValue

```
public void setValue(int row,  
                    int col,  
                    int value)  
    throws java.lang.IndexOutOfBoundsException
```

setValue sets value in the SudokuField

**Parameters:**

row - row of the field

col - column of the field

value - given value

**Throws:**

java.lang.IndexOutOfBoundsException - when row or column lower than 0 or higher than 8

### getRow

```
public SudokuRow getRow(int row)  
    throws java.lang.IndexOutOfBoundsException
```

getRow returns SudokuRow

**Parameters:**

row - row of the field

**Returns:**

SudokuRow

**Throws:**

java.lang.IndexOutOfBoundsException - when row lower than 0 or higher than 8

### getColumn

```
public SudokuColumn getColumn(int column)  
    throws java.lang.IndexOutOfBoundsException
```

getColumn returns SudokuColumn

**Parameters:**

column - column of the field

**Returns:**

SudokuColumn

**Throws:**

java.lang.IndexOutOfBoundsException - when column lower than 0 or higher than 8

### getBox

```
public SudokuBox getBox(int row,  
                       int column)  
    throws java.lang.IndexOutOfBoundsException
```

getBox returns SudokuBox

**Parameters:**

row - row of the field

column - column of the field

**Returns:**

SudokuBox

**Throws:**

java.lang.IndexOutOfBoundsException - when row, column lower than 0 or higher than 8



#### **cleanRandomlyFields**

```
public void cleanRandomlyFields(int amount)
```

Sets 0 value in randomly choosen SudokuFields

##### **Parameters:**

amount - amount of fields to be cleared

#### **isSolved**

```
public boolean isSolved()
```

isSolved checks if Sudoku has been solved correctly

##### **Returns:**

true if solved, false if not

#### **convert2dto1d**

```
public void convert2dto1d()
```

convert2dto1d converts List of List of SudokuFields to List of SudokuFields and sets it in linearBoard variable

#### **convert1dto2d**

```
public void convert1dto2d()
```

convert1dto2d converts List of SudokuFields to List of List of SudokuFields and sets it in field variable

#### **getId**

```
public java.lang.String getId()
```

#### **setId**

```
public void setId(java.lang.String id)
```

## 1.4 JDBC SudokuBoardDao

sudokupack

### Class JDBC SudokuBoardDao

java.lang.Object  
sudokupack.JDBC SudokuBoardDao

All Implemented Interfaces:

Dao<SudokuBoard>

```
public class JDBC SudokuBoardDao
extends java.lang.Object
implements Dao<SudokuBoard>
```

JDBC SudokuBoardDao is an implementation of Dao class. It's dao that read and write Sudoku from/to database.

#### Constructor Summary

##### Constructors

##### Constructor and Description

**JDBC SudokuBoardDao**(java.lang.String unit)  
Constructotr of JDBC SudokuBoardDao

#### Method Summary

##### All Methods

##### Instance Methods

##### Concrete Methods

Modifier and Type	Method and Description
void	<b>finalize</b> () finalize close existing connection to database.
java.util.List<java.lang.String>	<b>getAllBoardsNames</b> () getAllBoardNames returns all names of SudokuBoards that database currently contains
SudokuBoard	<b>read</b> (java.lang.String name) read SudokuBoard of given name from database.
void	<b>write</b> (SudokuBoard obj, java.lang.String name) write SudokuBoard to database

#### Methods inherited from class java.lang.Object

clone, equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

#### Constructor Detail

##### JDBC SudokuBoardDao

```
public JDBC SudokuBoardDao(java.lang.String unit)
```

Constructotr of JDBC SudokuBoardDao

##### Parameters:

unit - unit is a name of the persistance-unit name.

## Method Detail

### read

```
public SudokuBoard read(java.lang.String name)
```

read SudokuBoard of given name from database.

**Specified by:**

read in interface Dao<SudokuBoard>

**Parameters:**

name - name of the Sudoku

**Returns:**

SudokuBoard

### write

```
public void write(SudokuBoard obj,  
                 java.lang.String name)
```

write SudokuBoard to database

**Specified by:**

write in interface Dao<SudokuBoard>

**Parameters:**

obj - SudokuBoard which will be written

name - name of the Sudoku which will be it's id in database

### getAllBoardsNames

```
public java.util.List<java.lang.String> getAllBoardsNames()
```

getAllBoardNames returns all names of SudokuBoards that database currently contains

**Returns:**

List of strings with SudokuBoard names.

### finalize

```
public void finalize()
```

finalize close existing connection to database.

**Overrides:**

finalize in class java.lang.Object

# 1.5 FileSudokuBoardDao

sudokupack

## Class FileSudokuBoardDao

java.lang.Object  
sudokupack.FileSudokuBoardDao

All Implemented Interfaces:

Dao<SudokuBoard>

```
public class FileSudokuBoardDao
extends java.lang.Object
implements Dao<SudokuBoard>
```

FileSudokuBoardDao is an implementation of Dao class. It's dao that read and write Sudoku from/to file.

### Constructor Summary

#### Constructors

##### Constructor and Description

**FileSudokuBoardDao()**  
Parameterless constructor

### Method Summary

#### All Methods

#### Instance Methods

#### Concrete Methods

##### Modifier and Type

##### Method and Description

void

**finalize()**  
Close opened streams

**SudokuBoard**

**read(java.lang.String name)**  
read Sudoku from given file

void

**write(SudokuBoard obj, java.lang.String name)**  
write SudokuBoard to file

#### Methods inherited from class java.lang.Object

clone, equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Constructor Detail

#### FileSudokuBoardDao

```
public FileSudokuBoardDao()
Parameterless constructor
```

## Method Detail

### read

```
public SudokuBoard read(java.lang.String name)
    throws SudokuDeserializeException
```

read Sudoku from given file

**Specified by:**

read in interface `Dao<SudokuBoard>`

**Parameters:**

name - name of the file

**Returns:**

SudokuBoard instance

**Throws:**

SudokuDeserializeException - is being thrown when problem appear during reading SudokuBoard from file.

### write

```
public void write(SudokuBoard obj,
    java.lang.String name)
    throws SudokuSerializeException
```

write SudokuBoard to file

**Specified by:**

write in interface `Dao<SudokuBoard>`

**Parameters:**

obj - SudokuBoard instance to be written

name - name of the file

**Throws:**

SudokuSerializeException - is being thrown when problem appear during writing SudokuBoard to file

### finalize

```
public void finalize()
    throws SudokuDaoException
```

Close opened streams

**Overrides:**

finalize in class `java.lang.Object`

**Throws:**

SudokuDaoException - is being thrown when problem appear during closing opened streams

## 1.6 BackTrackingSudokuSolver

sudokupack

### Class BackTrackingSudokuSolver

java.lang.Object

sudokupack.BackTrackingSudokuSolver

All Implemented Interfaces:

SudokuSolver

```
public class BackTrackingSudokuSolver
extends java.lang.Object
implements SudokuSolver
```

BackTrackingSudokuSolver is an implementation of SudokuSolver interface

#### Constructor Summary

##### Constructors

##### Constructor and Description

BackTrackingSudokuSolver()

#### Method Summary

##### All Methods

##### Instance Methods

##### Concrete Methods

##### Modifier and Type

##### Method and Description

boolean

**equals**(java.lang.Object obj)

int

**hashCode**()

boolean

**solve**(SudokuBoard sudokuBoard)  
solve try to solve SudokuBoard

java.lang.String

**toString**()

##### Methods inherited from class java.lang.Object

clone, finalize, getClass, notify, notifyAll, wait, wait, wait

#### Constructor Detail

##### BackTrackingSudokuSolver

```
public BackTrackingSudokuSolver()
```

## Method Detail

### solve

```
public boolean solve(SudokuBoard sudokuBoard)
```

solve try to solve SudokuBoard

**Specified by:**

solve in interface `SudokuSolver`

**Parameters:**

sudokuBoard - SudokuBoard instance which has to be solved

**Returns:**

true if succeeded, false if failed

# 2. Najważniejsze klasy warstwy logiki

## 2.1 MainSudokuWindowController

sample.controllers

### Class MainSudokuWindowController

java.lang.Object  
sample.controllers.MainSudokuWindowController

public class MainSudokuWindowController  
extends java.lang.Object

MainSudokuWindowController is a controller class for MainSudokuWindow View.

#### Constructor Summary

Constructors
Constructor and Description
MainSudokuWindowController()

#### Method Summary

All Methods	Instance Methods	Concrete Methods
Modifier and Type		Method and Description
void		initialize() initialize method create and initialize ui controls.
void		onNewGameButtonClicked() onNewGameButtonClicked close MainSudokuWindow and shows ChooseLevelWindow
void		onResetGameButtonClicked() onResetGameButtonClicked reset Sudoku to starting state
void		onSaveDBStateButtonClicked() onSaveDBStateButtonClicked save game to database
void		onSaveFileStateButtonClicked() onSaveFileStateButtonClicked save game to file

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

#### Constructor Detail

MainSudokuWindowController
public MainSudokuWindowController()



## Method Detail

### initialize

```
public void initialize()
```

initialize method create and initialize ui controls.

### onResetGameButtonClicked

```
public void onResetGameButtonClicked()
```

onResetGameButtonClicked reset Sudoku to starting state

### onNewGameButtonClicked

```
public void onNewGameButtonClicked()
```

onNewGameButtonClicked close MainSudokuWindow and shows ChooseLevelWindow

### onSaveFileStateButtonClicked

```
public void onSaveFileStateButtonClicked()
```

onSaveFileStateButtonClicked save game to file

### onSaveDBStateButtonClicked

```
public void onSaveDBStateButtonClicked()
```

onSaveDBStateButtonClicked save game to database

## 2.2 ChooseLevelWindowController

sample.controllers

### Class ChooseLevelWindowController

java.lang.Object

sample.controllers.ChooseLevelWindowController

```
public class ChooseLevelWindowController
```

```
extends java.lang.Object
```

ChooseLevelWindowController is a controller class for ChooseLevelWindow View.

#### Constructor Summary

##### Constructors

##### Constructor and Description

`ChooseLevelWindowController()`

#### Method Summary

##### All Methods

##### Instance Methods

##### Concrete Methods

Modifier and Type	Method and Description
void	<b>initialize()</b> initialize method creates and initialize ui controls.
void	<b>onCloseButtonClicked()</b> onCloseButtonClicked method is being invoked on closeButton (in toolbar) click.
void	<b>onEasyButtonClick()</b> onEasyButtonClick method is being invoked on easyButton click.
void	<b>onExitButtonClick()</b> onExitButtonClick method is being invoked on exitButton click.
void	<b>onHardButtonClick()</b> onHardButtonClick method is being invoked on easyButton click.
void	<b>onLoadDBGameButtonClicked()</b> onLoadDBGameButtonClicked loads saved game from database.
void	<b>onLoadFileGameButtonClicked()</b> onLoadFileGameButtonClicked loads saved game from file
void	<b>onMediumButtonClick()</b> onMediumButtonClick method is being invoked on easyButton click.
void	<b>onRadioToggleGroupToggled()</b> onRadioToggleGroupToggled is being invoked after changing language on ui.

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

#### Constructor Detail

##### ChooseLevelWindowController

```
public ChooseLevelWindowController()
```

## Method Detail

### onEasyButtonClick

```
public void onEasyButtonClick()
```

onEasyButtonClick method is being invoked on easyButton click. It runs game in easy mode.

### onMediumButtonClick

```
public void onMediumButtonClick()
```

onMediumButtonClick method is being invoked on easyButton click. It runs game in medium mode.

### onHardButtonClick

```
public void onHardButtonClick()
```

onHardButtonClick method is being invoked on easyButton click. It runs game in hard mode.

### initialize

```
public void initialize()
```

initialize method creates and initialize ui controls.

### onExitButtonClick

```
public void onExitButtonClick()
```

onExitButtonClick method is being invoked on exitButton click. It close application.

### onRadioToggleGroupToggled

```
public void onRadioToggleGroupToggled()  
        throws FXMLOpenFailedException
```

onRadioToggleGroupToggled is being invoked after changing language on ui. It's changing application locale and refresh UI.

#### Throws:

FXMLOpenFailedException - throws it when exception occurs while loading fxml file.

### onCloseButtonClicked

```
public void onCloseButtonClicked()
```

onCloseButtonClicked method is being invoked on closeButton (in toolbar) click. It close application.

### onLoadFileGameButtonClicked

```
public void onLoadFileGameButtonClicked()
```

onLoadFileGameButtonClicked loads saved game from file

### onLoadDBGGameButtonClicked

```
public void onLoadDBGGameButtonClicked()
```

onLoadDBGGameButtonClicked loads saved game from database.

## 3. Najważniejsze klasy warstwy interfejsu

### 3.1 Main

sample

#### Class Main

java.lang.Object  
  javafx.application.Application  
    sample.Main

```
public class Main  
extends javafx.application.Application
```

#### Nested Class Summary

##### Nested classes/interfaces inherited from class javafx.application.Application

javafx.application.Application.Parameters

#### Field Summary

##### Fields inherited from class javafx.application.Application

STYLESET\_CASPIAN, STYLESET\_MODENA

#### Constructor Summary

##### Constructors

##### Constructor and Description

Main()

#### Method Summary

All Methods	Static Methods	Instance Methods	Concrete Methods
-------------	----------------	------------------	------------------

Modifier and Type	Method and Description
void	init()
static void	main(java.lang.String[] args)
void	start(javafx.stage.Stage primaryStage)

##### Methods inherited from class javafx.application.Application

getHostServices, getParameters, getUserAgentStylesheet, launch, launch, notifyPreloader, setUserAgentStylesheet, stop

##### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

#### Constructor Detail

##### Main

```
public Main()
```

## Method Detail

### init

```
public void init()  
    throws WaitFailedException
```

**Overrides:**

init in class `javafx.application.Application`

**Throws:**

`WaitFailedException`

### start

```
public void start(javafx.stage.Stage primaryStage)  
    throws FXMLOpenFailedException
```

**Specified by:**

start in class `javafx.application.Application`

**Throws:**

`FXMLOpenFailedException`

### main

```
public static void main(java.lang.String[] args)
```

# 3.2 MainSudokuWindow

sample

## Class MainSudokuWindow

java.lang.Object  
sample.MainSudokuWindow

public class MainSudokuWindow  
extends java.lang.Object

### Constructor Summary

#### Constructors

##### Constructor and Description

MainSudokuWindow()

### Method Summary

#### All Methods

#### Instance Methods

#### Concrete Methods

##### Modifier and Type

##### Method and Description

void

start()

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Constructor Detail

#### MainSudokuWindow

public MainSudokuWindow()

### Method Detail

#### start

public void start()  
throws FXMLOpenFailedException

#### Throws:

FXMLOpenFailedException

## 3.3 SplashScreenLoader

sample

### Class SplashScreenLoader

```
java.lang.Object
  javafx.application.Application
    javafx.application.Preloader
      sample.SplashScreenLoader
```

```
public class SplashScreenLoader
  extends javafx.application.Preloader
```

#### Nested Class Summary

##### Nested classes/interfaces inherited from class javafx.application.Preloader

javafx.application.Preloader.ErrorNotification, javafx.application.Preloader.PreloaderNotification, javafx.application.Preloader.ProgressNotification, javafx.application.Preloader.StateChangeNotification

##### Nested classes/interfaces inherited from class javafx.application.Application

javafx.application.Application.Parameters

#### Field Summary

##### Fields inherited from class javafx.application.Application

STYLESHEET\_CASPIAN, STYLESHEET\_MODENA

#### Constructor Summary

##### Constructors

##### Constructor and Description

SplashScreenLoader()

#### Method Summary

##### All Methods

##### Instance Methods

##### Concrete Methods

Modifier and Type	Method and Description
javafx.scene.Scene	<b>createScene()</b>
void	<b>handleStateChangeNotification</b> (javafx.application.Preloader.StateChangeNotification notification)
void	<b>start</b> (javafx.stage.Stage stage)

##### Methods inherited from class javafx.application.Preloader

handleApplicationNotification, handleErrorNotification, handleProgressNotification

##### Methods inherited from class javafx.application.Application

getHostServices, getParameters, getUserAgentStylesheet, init, launch, launch, notifyPreloader, setUserAgentStylesheet, stop

##### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

#### Constructor Detail

##### SplashScreenLoader

```
public SplashScreenLoader()
```

## Method Detail

### start

```
public void start(javafx.stage.Stage stage)
    throws java.lang.Exception
```

**Specified by:**

start in class `javafx.application.Application`

**Throws:**

`java.lang.Exception`

### createScene

```
public javafx.scene.Scene createScene()
```

### handleStateChangeNotification

```
public void handleStateChangeNotification(javafx.application.Preloader.StateChangeNotification notification)
```

**Overrides:**

handleStateChangeNotification in class `javafx.application.Preloader`

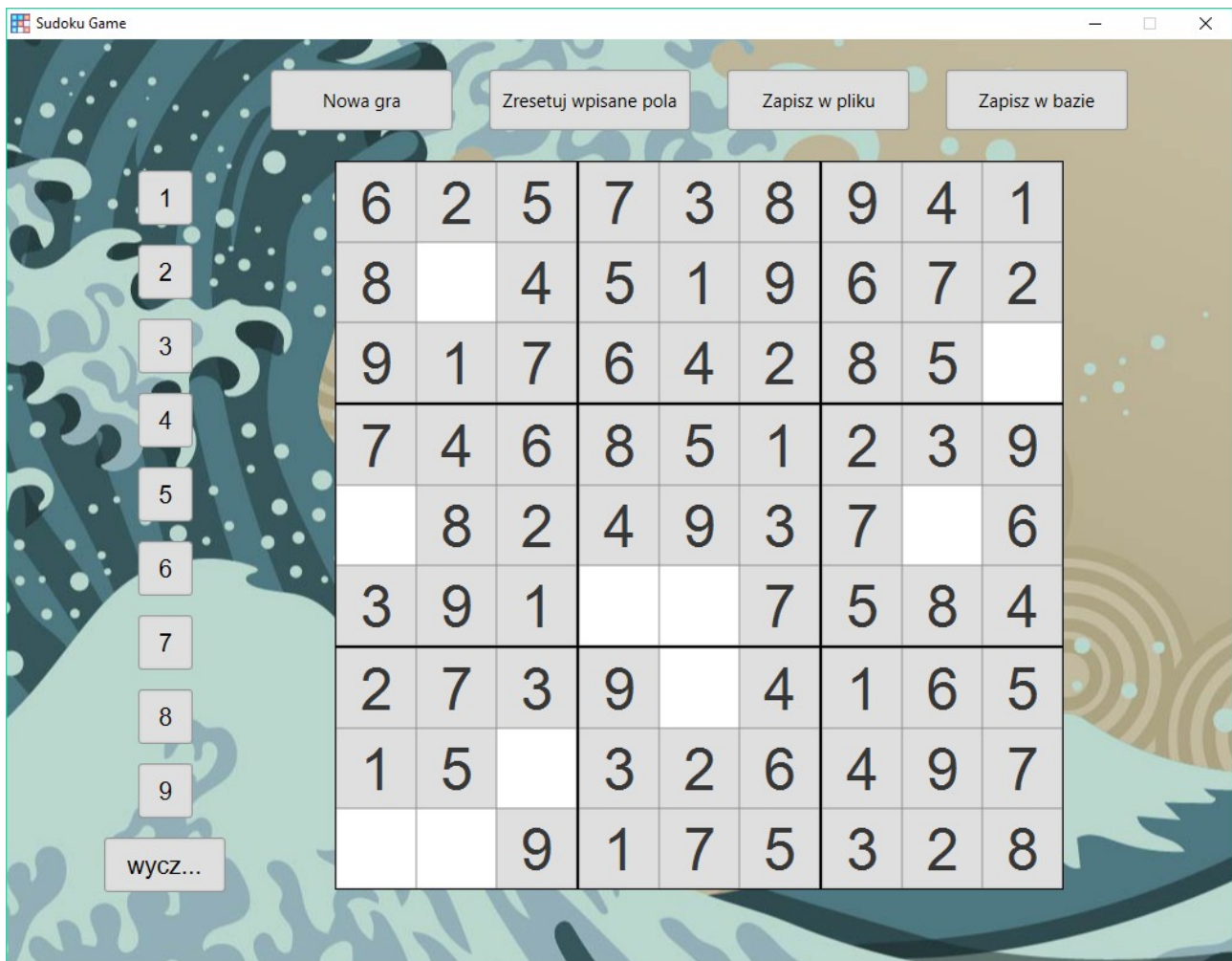


### 3.4 Zrzuty ekranu najważniejszych widoków działającej aplikacji

#### 3.4.1. ChooseLevelWindow



### 3.4.2. MainSudokuWindow



## 3.5 Opis przycisków aplikacji

### 3.5.1. ChooseLevelWindow

- Menu File:
  - Load from file – odczyt zapisu stanu gry z wybranego pliku,
  - Load from database – odczyt zapisu stanu gry z lokalnej bazy danych,
  - Close – zamknięcie aplikacji,
- Łatwy, Średni, Trudny – wybór poziomu trudności, po kliknięciu otworzy się okno gry,
- Wyjście – zamknięcie aplikacji,
- Flaga – zmiana języka aplikacji.

### 3.5.2. MainSudokuWindow

Wypełnienie danego pola liczbą polega na jednokrotnym kliknięciu w wybrane pole na planszy a następnie kliknięcie w przycisk w lewej części okna z liczbą odpowiadającą tej, którą chcemy wpisać.

Przyciski:

- 1..9 – wypełnienie zaznaczonego pola wybraną liczbą,
- Wyczyść – wyczyszczenie zaznaczone pole,
- Nowa gra – zakończenie obecnej gry (zamknięcie okna) i otworenie okna ChooseLevelWindow,
- Zresetuj wpisane pola – wyczyszczenie wszystkich wypełnionych pól,
- Zapisz w pliku – zapisanie aktualnego stanu gry do pliku,
- Zapisz w bazie – zapisanie aktualnego stanu gry do lokalnej bazy danych.

## 3.6 Podsumowanie

Napisana przez nas aplikacja spełnia wymagania dotyczące założeń projektu. Kolekcja danych przechowywana jest w odpowiedniej, napisanej przez nas klasie. Algorytm wypełnienia i rozwiązywania Sudoku może zostać zastąpiony każdym innym, o ile implementuje on interfejs SudokuSolver. Sprawdzanie poprawności elementów Sudoku jest przeprowadzane w zgodnych z założeniami klasach. Wszystkie wymagane klasy posiadają nadpisane metody toString(), equals() oraz hashCode(). Aplikacja pozwala na zapis stanu gry do pliku jak i do bazy. Posiada dwukierunkowe wiązanie instancji SudokuBoard i formularza z planszą Sudoku. Do kontroli błędów w naszej implementacji wykorzystujemy hierarchię własnych wyjątków.

W stworzonej przez nas aplikacji występuje pewien błąd – w trakcie jeden na kilka gier część wiązań między instancjami SudokuBoard a polami planszy interfejsu użytkownika nie zostaje utworzona. Żaden wyjątek nie zostaje rzucony podczas wystąpienia tego błędu. Mimo usilnych starań nie potrafiliśmy rozwiązać powyższego problemu.