

Scrabble

2.0

Generated by Doxygen 1.8.17



<b>1 The-Good-The-Bad-and-The-Ugly</b>	<b>1</b>
1.0.1 <strong>Scrabble Game</strong> . . . . .	1
<b>2 Hierarchical Index</b>	<b>3</b>
2.1 Class Hierarchy . . . . .	3
<b>3 Class Index</b>	<b>5</b>
3.1 Class List . . . . .	5
<b>4 Class Documentation</b>	<b>7</b>
4.1 scrabble.Board Class Reference . . . . .	7
4.1.1 Detailed Description . . . . .	7
4.1.2 Constructor & Destructor Documentation . . . . .	7
4.1.2.1 Board() . . . . .	8
4.1.3 Member Function Documentation . . . . .	8
4.1.3.1 checkPlayerHasTiles() . . . . .	8
4.1.3.2 checkPositionContainsTile() . . . . .	8
4.1.3.3 checkPositionLine() . . . . .	8
4.1.3.4 checkValidMove() . . . . .	9
4.1.3.5 checkValidPosition() . . . . .	9
4.1.3.6 checkWordConnects() . . . . .	9
4.1.3.7 checkWordLength() . . . . .	10
4.1.3.8 getBoardSquares() . . . . .	10
4.1.3.9 getSquare() . . . . .	10
4.1.3.10 placeTile() . . . . .	11
4.1.3.11 placeTiles() . . . . .	11
4.1.3.12 resetBoard() . . . . .	11
4.1.3.13 toString() . . . . .	11
4.1.4 Member Data Documentation . . . . .	12
4.1.4.1 BOARD_SIZE . . . . .	12
4.2 scrabble.BoardTest Class Reference . . . . .	12
4.3 scrabble.Frame Class Reference . . . . .	12
4.3.1 Detailed Description . . . . .	13
4.3.2 Constructor & Destructor Documentation . . . . .	13
4.3.2.1 Frame() . . . . .	13
4.3.3 Member Function Documentation . . . . .	13
4.3.3.1 addTile() . . . . .	13
4.3.3.2 checkTiles() [1/2] . . . . .	13
4.3.3.3 checkTiles() [2/2] . . . . .	15
4.3.3.4 fillFrame() . . . . .	15
4.3.3.5 getTile() [1/2] . . . . .	15
4.3.3.6 getTile() [2/2] . . . . .	16
4.3.3.7 getTiles() . . . . .	16

4.3.3.8 isEmpty()	16
4.3.3.9 removeTile() [1/3]	17
4.3.3.10 removeTile() [2/3]	18
4.3.3.11 removeTile() [3/3]	18
4.3.3.12 removeTiles() [1/2]	18
4.3.3.13 removeTiles() [2/2]	19
4.3.3.14 returnFrame()	19
4.3.3.15 swapTiles()	19
4.3.3.16 toString()	20
4.4 scrabble.exceptions.InvalidBoardException Class Reference	20
4.5 scrabble.exceptions.InvalidFrameException Class Reference	20
4.6 scrabble.exceptions.InvalidPlayerNameException Class Reference	21
4.7 scrabble.exceptions.InvalidPlayerScoreException Class Reference	21
4.8 scrabble.exceptions.InvalidPoolException Class Reference	22
4.9 scrabble.exceptions.InvalidSquareException Class Reference	22
4.10 scrabble.exceptions.InvalidTileException Class Reference	22
4.11 scrabble.Main Class Reference	23
4.12 scrabble.Player Class Reference	23
4.12.1 Detailed Description	23
4.12.2 Constructor & Destructor Documentation	23
4.12.2.1 Player()	23
4.12.3 Member Function Documentation	24
4.12.3.1 decreaseScore()	24
4.12.3.2 getName()	24
4.12.3.3 getPlayerFrame()	25
4.12.3.4 getScore()	25
4.12.3.5 increaseScore()	25
4.12.3.6 playerReset()	25
4.12.3.7 setName()	26
4.12.3.8 toString()	26
4.13 scrabble.Pool Class Reference	26
4.13.1 Detailed Description	27
4.13.2 Constructor & Destructor Documentation	27
4.13.2.1 Pool()	27
4.13.3 Member Function Documentation	27
4.13.3.1 isEmpty()	27
4.13.3.2 poolFill()	27
4.13.3.3 receiveTile()	27
4.13.3.4 removeTile()	28
4.13.3.5 tilesInPool()	28
4.13.3.6 toString()	28
4.14 scrabble.PoolTest Class Reference	29

4.15 scrabble.Square Class Reference . . . . .	29
4.15.1 Detailed Description . . . . .	29
4.15.2 Constructor & Destructor Documentation . . . . .	29
4.15.2.1 Square() . . . . .	29
4.15.3 Member Function Documentation . . . . .	30
4.15.3.1 getTile() . . . . .	30
4.15.3.2 getType() . . . . .	30
4.15.3.3 isEmpty() . . . . .	30
4.15.3.4 setNormal() . . . . .	30
4.15.3.5 setTile() . . . . .	30
4.15.3.6 toString() . . . . .	31
4.16 scrabble.SquareTest Class Reference . . . . .	31
4.17 scrabble.Tile Class Reference . . . . .	31
4.17.1 Detailed Description . . . . .	32
4.17.2 Constructor & Destructor Documentation . . . . .	32
4.17.2.1 Tile() . . . . .	32
4.17.3 Member Function Documentation . . . . .	32
4.17.3.1 compareTo() . . . . .	32
4.17.3.2 equals() . . . . .	33
4.17.3.3 getCharacter() . . . . .	33
4.17.3.4 getValue() . . . . .	33
4.17.3.5 main() . . . . .	34
4.17.3.6 setCharacter() . . . . .	34
4.17.3.7 setNull() . . . . .	34
4.17.3.8 toString() . . . . .	34
<b>Index</b>	<b>37</b>



# Chapter 1

## The-Good-The-Bad-and-The-Ugly

**Authors:** Group 9: The Good, The Bad and The Ugly

John O'Donnell - 18368983  
Éanna Curran - 18311676  
Killian Callaghan - 18332783

### 1.0.1 <strong>Scrabble Game</strong>

#### Classes

Player - Used to create objects that represents the players in Scrabble  
Frame - Used to create objects that represents the players' character frames in Scrabble  
Pool - Used to create an object that represents the pool bag of tiles in Scrabble  
Tile - Used to create objects that represents the character tiles in Scrabble  
Square - Used to create objects that represents the squares on the board in Scrabble  
Board - Used to create object that represents the board in Scrabble  
Main - Is the class containing the main method executed by the jar and run example of the first few turns

#### Custom Exceptions

InvalidFrameException - Custom Exception for Frame Class  
InvalidPlayerNameException - Custom Exception for name in Player Class  
InvalidPlayerScoreException - Custom Exception for Score in Player Class  
InvalidPoolException - Custom Exception for Pool Class  
InvalidTileException - Custom Exception for Tile Class  
InvalidSquareException - Custom Exception for Square Class  
InvalidBoardException - Custom Exception for Board Class

#### JUnit Tests

PlayerUnitTest - Tests Player Class  
FrameTest - Tests Frame Class  
PoolTest - Tests Pool Class  
TileTest - Tests Tile Class  
SquareTest - Tests Square Class  
BoardTest - Tests Board Class

**Project Notes:**

This project is a gradle project and uses gradle and groovy as its build tool. To run the junit test create a gradle project by running `./gradlew build` in command line in the project folder to generate a gradle project. Then navigate in command line to the project folder and run `"gradlew test"`. The output of the unit test is in a html file at `"ProjectFolder"/build/reports/tests/test/index.html` (Where ProjectFolder is the folder where the project is stored).

The executable jar file run the Main Class that creates sample objects from Scrabble and prints out there initialized states.

The documentation is in the Doxygen folder as a pdf.

HTML JavaDoc is in `build\docs\javadoc\index`

Note: This project was made made on Windows 10 and was tested on Windows 10, other operating systems have not been test. Therefore there may be compatibility issues with other operating systems.



## Chapter 2

# Hierarchical Index

### 2.1 Class Hierarchy

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

scrabble.Board . . . . .	7
scrabble.BoardTest . . . . .	12
Comparable	
scrabble.Tile . . . . .	31
scrabble.Frame . . . . .	12
IllegalArgumentException	
scrabble.exceptions.InvalidBoardException . . . . .	20
scrabble.exceptions.InvalidFrameException . . . . .	20
scrabble.exceptions.InvalidPlayerNameException . . . . .	21
scrabble.exceptions.InvalidPlayerScoreException . . . . .	21
scrabble.exceptions.InvalidSquareException . . . . .	22
scrabble.exceptions.InvalidTileException . . . . .	22
scrabble.Main . . . . .	23
scrabble.Player . . . . .	23
scrabble.Pool . . . . .	26
scrabble.PoolTest . . . . .	29
scrabble.Square . . . . .	29
scrabble.SquareTest . . . . .	31
NoSuchElementException	
scrabble.exceptions.InvalidPoolException . . . . .	22



## Chapter 3

# Class Index

### 3.1 Class List

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

<a href="#">scrabble.Board</a>	7
<a href="#">scrabble.BoardTest</a>	12
<a href="#">scrabble.Frame</a>	12
<a href="#">scrabble.exceptions.InvalidBoardException</a>	20
<a href="#">scrabble.exceptions.InvalidFrameException</a>	20
<a href="#">scrabble.exceptions.InvalidPlayerNameException</a>	21
<a href="#">scrabble.exceptions.InvalidPlayerScoreException</a>	21
<a href="#">scrabble.exceptions.InvalidPoolException</a>	22
<a href="#">scrabble.exceptions.InvalidSquareException</a>	22
<a href="#">scrabble.exceptions.InvalidTileException</a>	22
<a href="#">scrabble.Main</a>	23
<a href="#">scrabble.Player</a>	23
<a href="#">scrabble.Pool</a>	26
<a href="#">scrabble.PoolTest</a>	29
<a href="#">scrabble.Square</a>	29
<a href="#">scrabble.SquareTest</a>	31
<a href="#">scrabble.Tile</a>	31



## Chapter 4

# Class Documentation

### 4.1 scrabble.Board Class Reference

#### Public Member Functions

- [Board](#) ()
- [Square](#)[][] [getBoardSquares](#) ()
- void [resetBoard](#) ()
- [Square](#) [getSquare](#) (int i, int j)
- String [toString](#) ()
- void [placeTiles](#) ([Player](#) player, char[] word, int[][] positions)

#### Static Public Attributes

- static final int [BOARD\\_SIZE](#) = 15

#### Protected Member Functions

- void [checkValidMove](#) ([Player](#) player, char[] word, int[][] positions)
- void [placeTile](#) ([Tile](#) tile, int position\_i, int position\_j)
- void [checkValidPosition](#) (int[][] positions)
- void [checkPlayerHasTiles](#) ([Player](#) player, char[] word)
- void [checkWordLength](#) (char[] word)
- void [checkPositionContainsTile](#) (int[][] position)
- void [checkPositionLine](#) (int[][] position)
- void [checkWordConnects](#) (int[][] position)

#### 4.1.1 Detailed Description

The [Board](#) Class represents the [Board](#) for the Scrabble Game as an object

#### 4.1.2 Constructor & Destructor Documentation

#### 4.1.2.1 Board()

```
scrabble.Board.Board ( )
```

[Board](#) Constructor

### 4.1.3 Member Function Documentation

#### 4.1.3.1 checkPlayerHasTiles()

```
void scrabble.Board.checkPlayerHasTiles (
    Player player,
    char[] word ) [protected]
```

Method to validate if the [Player](#) placing a list of Tiles has each [Tile](#) in his [Frame](#)

Parameters

<i>player</i>	<a href="#">Player</a> to check if they have the necessary Tiles
<i>word</i>	List of Tiles to check

#### 4.1.3.2 checkPositionContainsTile()

```
void scrabble.Board.checkPositionContainsTile (
    int position[][] ) [protected]
```

Method to check if a position on the [Board](#) already has a [Tile](#) in it

Parameters

<i>position</i>	Co-ordinates to check if a <a href="#">Tile</a> in already in it
-----------------	--

#### 4.1.3.3 checkPositionLine()

```
void scrabble.Board.checkPositionLine (
    int position[][] ) [protected]
```

Method to check that a list of positions to place a word is in a line

## Parameters

<i>position</i>	List of co-ordinates to check they are part of a line of Tiles
-----------------	--

**4.1.3.4 checkValidMove()**

```
void scrabble.Board.checkValidMove (
    Player player,
    char[] word,
    int positions[][] ) [protected]
```

Method to check that a move from the player is valid

## Parameters

<i>player</i>	Player making the move
<i>word</i>	List of tiles requested to make the move
<i>positions</i>	Positions entered to place each Tile

**4.1.3.5 checkValidPosition()**

```
void scrabble.Board.checkValidPosition (
    int positions[][] ) [protected]
```

Method to validate that a position passed in is on the board

## Parameters

<i>positions</i>	Position to check if its on the board
------------------	---------------------------------------

**4.1.3.6 checkWordConnects()**

```
void scrabble.Board.checkWordConnects (
    int position[][] ) [protected]
```

Method to check if a list of positions connect with a tile already on the board

## Parameters

<i>position</i>	List of positions to check if any of them would connect with a tile on the board
-----------------	--

#### 4.1.3.7 checkWordLength()

```
void scrabble.Board.checkWordLength (
    char[] word ) [protected]
```

Method to check that words to be placed on the board are valid lengths

##### Parameters

<i>word</i>	List of Tiles to check
-------------	------------------------

#### 4.1.3.8 getBoardSquares()

```
Square [][] scrabble.Board.getBoardSquares ( )
```

Method to return the current [Board](#)

##### Returns

the current [Board](#)

#### 4.1.3.9 getSquare()

```
Square scrabble.Board.getSquare (
    int i,
    int j )
```

Method that returns a specific board square

##### Parameters

<i>i</i>	row coordinate
<i>j</i>	column coordinate

##### Returns

The [Square](#) at i and j

##### Exceptions

<i>InvalidBoardException</i>	Coordinates are not inside the <a href="#">Board</a>
------------------------------	--



#### 4.1.3.10 placeTile()

```
void scrabble.Board.placeTile (
    Tile tile,
    int position_i,
    int position_j ) [protected]
```

Method to place a [Tile](#) on the board

##### Parameters

<i>tile</i>	<a href="#">Tile</a> to be placed on the <a href="#">Board</a>
<i>position↔ _j</i>	I position on the <a href="#">Board</a> to place the <a href="#">Tile</a>
<i>position↔ _j</i>	J position on the <a href="#">Board</a> to place the <a href="#">Tile</a>

#### 4.1.3.11 placeTiles()

```
void scrabble.Board.placeTiles (
    Player player,
    char[] word,
    int positions[][] )
```

Method for a [Player](#) to place a list of Tiles on the [Board](#)

##### Parameters

<i>player</i>	Person to place Tiles
<i>word</i>	Characters to place on the <a href="#">Board</a>
<i>positions</i>	List of positions to place Tiles on the <a href="#">Board</a>

#### 4.1.3.12 resetBoard()

```
void scrabble.Board.resetBoard ( )
```

Method to reset the [Board](#)

#### 4.1.3.13 toString()

```
String scrabble.Board.toString ( )
```

toString method that prints the [Board](#)

#### Returns

Returns the board as a string

### 4.1.4 Member Data Documentation

#### 4.1.4.1 BOARD\_SIZE

```
final int scrabble.Board.BOARD_SIZE = 15 [static]
```

Constant value for [Board](#) size

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

- [src/main/java/scrabble/Board.java](#)

## 4.2 scrabble.BoardTest Class Reference

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

- [src/test/java/scrabble/BoardTest.java](#)

## 4.3 scrabble.Frame Class Reference

### Public Member Functions

- [Frame](#) ([Pool](#) pool)
- void [fillFrame](#) ()
- [ArrayList](#)< [Tile](#) > [returnFrame](#) ()
- boolean [isEmpty](#) ()
- void [removeTile](#) (int i)
- void [removeTile](#) (char c)
- void [removeTile](#) ([Tile](#) t)
- void [removeTiles](#) (char[] word)
- void [removeTiles](#) ([ArrayList](#)< [Tile](#) > tiles)
- void [addTile](#) ([Tile](#) tile)
- boolean [checkTiles](#) ([ArrayList](#)< [Tile](#) > tiles)
- boolean [checkTiles](#) (char[] word)
- [Tile](#) [getTile](#) (char c)
- [Tile](#) [getTile](#) (int i)
- [ArrayList](#)< [Tile](#) > [getTiles](#) (char[] word)
- void [swapTiles](#) ([ArrayList](#)< [Tile](#) > tiles)
- String [toString](#) ()

## Static Public Member Functions

- static void **main** (String[] args)

### 4.3.1 Detailed Description

This class represents the [Player's Frame](#) in Scrabble

### 4.3.2 Constructor & Destructor Documentation

#### 4.3.2.1 Frame()

```
scrabble.Frame.Frame (  
    Pool pool )
```

Constructor for [Frame](#)

##### Parameters

<i>pool</i>	The reference to the <a href="#">Pool</a> to access Tiles from
-------------	--

### 4.3.3 Member Function Documentation

#### 4.3.3.1 addTile()

```
void scrabble.Frame.addTile (  
    Tile tile )
```

Method to add a single [Tile](#) to the frame

##### Parameters

<i>tile</i>	To be added to the playerFrame
-------------	--------------------------------

#### 4.3.3.2 checkTiles() [1/2]

```
boolean scrabble.Frame.checkTiles (  
    ArrayList< Tile > tiles )
```

Method which checks if a series of Tiles are currently in the [Frame](#)

## Parameters

<i>tiles</i>	List of tiles to be checked
--------------	-----------------------------

## Returns

boolean: Result for if the [Frame](#) contains all the Tiles

**4.3.3.3 checkTiles()** [2/2]

```
boolean scrabble.Frame.checkTiles (
    char[] word )
```

Method to check if a list of characters are in the [Frame](#)

## Parameters

<i>word</i>	List of characters to check
-------------	-----------------------------

## Returns

Boolean answer

**4.3.3.4 fillFrame()**

```
void scrabble.Frame.fillFrame ( )
```

Method to fill the [Frame](#) up to the maximum number of Tiles

**4.3.3.5 getTile()** [1/2]

```
Tile scrabble.Frame.getTile (
    char c )
```

Method that retrieves a [Tile](#) with a given character from the [Frame](#)

## Parameters

<i>c</i>	Character of <a href="#">Tile</a> wanted
----------	--

## Returns

[Tile](#) with given character

#### 4.3.3.6 `getTile()` [2/2]

```
Tile scrabble.Frame.getTile (
    int i )
```

Accessor method for [Tile](#) from [Frame](#)

##### Parameters

<i>i</i>	index of the <a href="#">Tile</a> in the <a href="#">Frame</a>
----------	--

##### Returns

The [Tile](#) at index *i* in the [Frame](#)

#### 4.3.3.7 `getTiles()`

```
ArrayList<Tile> scrabble.Frame.getTiles (
    char[] word )
```

Method to retrieve a list of [Tile](#) with given characters from [Frame](#)

##### Parameters

<i>word</i>	list of wanted character Tiles
-------------	--------------------------------

##### Returns

List of Tiles

#### 4.3.3.8 `isEmpty()`

```
boolean scrabble.Frame.isEmpty ( )
```

Method to check if the `playerFrame` has Tiles in it

##### Returns

boolean: Answer for if the `playerFrame` is empty

#### 4.3.3.9 removeTile() [1/3]

```
void scrabble.Frame.removeTile (  
    char c )
```

Method to remove a single [Tile](#) from the [Frame](#)

## Parameters

<i>c</i>	Character of the <a href="#">Tile</a> to be removed
----------	---

## Exceptions

<i>InvalidTileException</i>	If Character is not in the <a href="#">Frame</a>
-----------------------------	--

**4.3.3.10 removeTile()** [2/3]

```
void scrabble.Frame.removeTile (  
    int i )
```

Method to remove a single [Tile](#) from the [Frame](#)

## Parameters

<i>i</i>	<a href="#">Tile</a> index to be removed
----------	--

## Exceptions

<i>InvalidTileException</i>	If index is not in the <a href="#">Frame</a>
-----------------------------	--

**4.3.3.11 removeTile()** [3/3]

```
void scrabble.Frame.removeTile (  
    Tile t )
```

Method to remove a single [Tile](#) from the [Frame](#)

## Parameters

<i>t</i>	<a href="#">Tile</a> to be removed
----------	------------------------------------

**4.3.3.12 removeTiles()** [1/2]

```
void scrabble.Frame.removeTiles (  
    ArrayList< Tile > tiles )
```

Method to remove a list of Tiles from the [Frame](#)



## Parameters

<i>tiles</i>	List of Tiles to be removed
--------------	-----------------------------

## Exceptions

<i>InvalidFrameException</i>	If the Tiles to be removed are in the <a href="#">Frame</a>
------------------------------	---

**4.3.3.13 removeTiles()** [2/2]

```
void scrabble.Frame.removeTiles (
    char[] word )
```

Method to remove a multiple [Tile](#) from the [Frame](#)

## Parameters

<i>word</i>	Array of <a href="#">Tile</a> Characters to be removed
-------------	--

## Exceptions

<i>InvalidTileException</i>	If <a href="#">Tile</a> is not in the <a href="#">Frame</a>
-----------------------------	---

**4.3.3.14 returnFrame()**

```
ArrayList<Tile> scrabble.Frame.returnFrame ( )
```

Accessor method for playerFrame

## Returns

playerFrame: The ArrayList of Tiles in playerFrame

**4.3.3.15 swapTiles()**

```
void scrabble.Frame.swapTiles (
    ArrayList< Tile > tiles )
```

Method used to swap a number of Tiles from the [Frame](#) for new ones in the [Pool](#)

**Parameters**

<i>tiles</i>	List of Tiles to be removed
--------------	-----------------------------

**4.3.3.16 toString()**

```
String scrabble.Frame.toString ( )
```

Method overriding the toString method

**Returns**

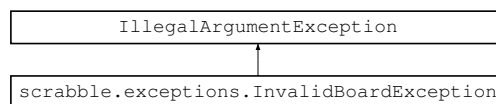
String: Formatted string of the Tiles in [Frame](#)

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

- src/main/java/scrabble/Frame.java

**4.4 scrabble.exceptions.InvalidBoardException Class Reference**

Inheritance diagram for scrabble.exceptions.InvalidBoardException:

**Public Member Functions**

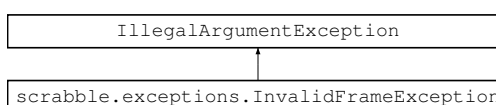
- **InvalidBoardException** (String s)

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

- src/main/java/scrabble/exceptions/InvalidBoardException.java

**4.5 scrabble.exceptions.InvalidFrameException Class Reference**

Inheritance diagram for scrabble.exceptions.InvalidFrameException:



## Public Member Functions

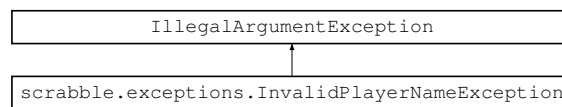
- **InvalidFrameException** (String s)

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

- src/main/java/scrabble/exceptions/InvalidFrameException.java

## 4.6 scrabble.exceptions.InvalidPlayerNameException Class Reference

Inheritance diagram for scrabble.exceptions.InvalidPlayerNameException:



## Public Member Functions

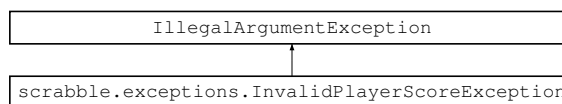
- **InvalidPlayerNameException** (String s)

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

- src/main/java/scrabble/exceptions/InvalidPlayerNameException.java

## 4.7 scrabble.exceptions.InvalidPlayerScoreException Class Reference

Inheritance diagram for scrabble.exceptions.InvalidPlayerScoreException:



## Public Member Functions

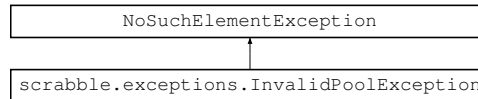
- **InvalidPlayerScoreException** (String s)

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

- src/main/java/scrabble/exceptions/InvalidPlayerScoreException.java

## 4.8 scrabble.exceptions.InvalidPoolException Class Reference

Inheritance diagram for `scrabble.exceptions.InvalidPoolException`:



### Public Member Functions

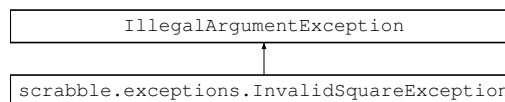
- **InvalidPoolException** (String s)

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

- `src/main/java/scrabble/exceptions/InvalidPoolException.java`

## 4.9 scrabble.exceptions.InvalidSquareException Class Reference

Inheritance diagram for `scrabble.exceptions.InvalidSquareException`:



### Public Member Functions

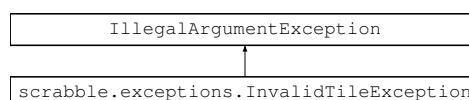
- **InvalidSquareException** (String s)

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

- `src/main/java/scrabble/exceptions/InvalidSquareException.java`

## 4.10 scrabble.exceptions.InvalidTileException Class Reference

Inheritance diagram for `scrabble.exceptions.InvalidTileException`:



## Public Member Functions

- [InvalidTileException](#) (String s)

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

- src/main/java/scrabble/exceptions/InvalidTileException.java

## 4.11 scrabble.Main Class Reference

### Static Public Member Functions

- static void **main** (String[] args)

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

- src/main/java/scrabble/Main.java

## 4.12 scrabble.Player Class Reference

### Public Member Functions

- [Player](#) (String namePlayer, [Pool](#) pool)
- void [playerReset](#) (String newName)
- String [getName](#) ()
- void [setName](#) (String name)
- int [getScore](#) ()
- void [increaseScore](#) (int scoreIncrease)
- void [decreaseScore](#) (int scoreDecrease)
- [Frame](#) [getPlayerFrame](#) ()
- boolean **charUserInputCheck** (char letter)
- String [toString](#) ()

### 4.12.1 Detailed Description

Class that represents a player. Contains the information of each players

### 4.12.2 Constructor & Destructor Documentation

#### 4.12.2.1 Player()

```
scrabble.Player.Player (  
    String namePlayer,  
    Pool pool )
```

[Player](#) Constructor

## Parameters

<i>namePlayer</i>	String for the name of the <a href="#">Player</a>
<i>pool</i>	Reference to the <a href="#">Pool</a> of the game

## Exceptions

<i>InvalidPlayerNameException</i>	If inputted name is invalid
-----------------------------------	-----------------------------

## 4.12.3 Member Function Documentation

### 4.12.3.1 decreaseScore()

```
void scrabble.Player.decreaseScore (
    int scoreDecrease )
```

Mutator method for score to decrease the players score

## Parameters

<i>scoreDecrease</i>	The value for the score to be increased by
----------------------	--

## Exceptions

<i>InvalidPlayerScoreException</i>	If a negative value is passed into Decrease_Score
------------------------------------	---

### 4.12.3.2 getName()

```
String scrabble.Player.getName ( )
```

Accessor Method for Name

## Returns

The name of the player

#### 4.12.3.3 getPlayerFrame()

```
Frame scrabble.Player.getPlayerFrame ( )
```

Accessing Method player's frame

##### Returns

The value of playerFrame

#### 4.12.3.4 getScore()

```
int scrabble.Player.getScore ( )
```

Accessor Method for [Player](#) Score

##### Returns

The [Player](#)'s Score

#### 4.12.3.5 increaseScore()

```
void scrabble.Player.increaseScore (
    int scoreIncrease )
```

Mutator method for score to increase the players score

##### Parameters

<i>scoreIncrease</i>	The value for the score to be increased by
----------------------	--

##### Exceptions

<i>InvalidPlayerScoreException</i>	If player's score is increased by a negative value
------------------------------------	--

#### 4.12.3.6 playerReset()

```
void scrabble.Player.playerReset (
    String newName )
```

Resetting the score and name of the player

**Parameters**

<i>newName</i>	The new player name after the player is reset
----------------	---

**Exceptions**

<i>InvalidPlayerNameException</i>	If inputted name is invalid
-----------------------------------	-----------------------------

**4.12.3.7 setName()**

```
void scrabble.Player.setName (
    String name )
```

Mutator Method for name

**Parameters**

<i>name</i>	The new Name of the <a href="#">Player</a>
-------------	--

**Exceptions**

<i>InvalidPlayerNameException</i>	If inputted name is invalid
-----------------------------------	-----------------------------

**4.12.3.8 toString()**

```
String scrabble.Player.toString ( )
```

A toString method to print the [Player](#) class variables

**Returns**

The [Player](#) class variables

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

- src/main/java/scrabble/Player.java

**4.13 scrabble.Pool Class Reference****Public Member Functions**

- void [poolFill](#) () throws InvalidTileException
- String [toString](#) ()
- [Tile removeTile](#) () throws InvalidPoolException
- void [receiveTile](#) ([Tile](#) tileAdded)
- int [tilesInPool](#) ()
- boolean [isEmpty](#) ()
- [Pool](#) ()



## Static Public Member Functions

- static void **main** (String[] args)

### 4.13.1 Detailed Description

Class that represent a Character [Tile](#) for the Game of Scrabble

### 4.13.2 Constructor & Destructor Documentation

#### 4.13.2.1 Pool()

```
scrabble.Pool.Pool ( )
```

[Pool](#) constructor

### 4.13.3 Member Function Documentation

#### 4.13.3.1 isEmpty()

```
boolean scrabble.Pool.isEmpty ( )
```

Method to Check if the pool is empty

##### Returns

True if pool is empty

#### 4.13.3.2 poolFill()

```
void scrabble.Pool.poolFill ( ) throws InvalidTileException
```

Function to fill the array with the set amount of each [Tile](#) in the standard English rules

#### 4.13.3.3 receiveTile()

```
void scrabble.Pool.receiveTile (
    Tile tileAdded )
```

Method to take in a tile and add it to a pool

## Parameters

<i>tileAdded</i>	The tile to be added to the pool
------------------	----------------------------------

**4.13.3.4 removeTile()**

`Tile` `scrabble.Pool.removeTile ( )` throws `InvalidPoolException`

Method to remove a random tile from the pool and return the tile that was removed

## Returns

The tile which was randomly removed from the pool

## Exceptions

<i>InvalidPoolException</i>	If the pool is empty
-----------------------------	----------------------

**4.13.3.5 tilesInPool()**

`int` `scrabble.Pool.tilesInPool ( )`

Method to find the number of tiles in pool

## Returns

The number of tiles in pool

**4.13.3.6 toString()**

`String` `scrabble.Pool.toString ( )`

`Pool` toString Method

## Returns

`Pool` information in the form of a String

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

- `src/main/java/scrabble/Pool.java`

## 4.14 scrabble.PoolTest Class Reference

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

- `src/test/java/scrabble/PoolTest.java`

## 4.15 scrabble.Square Class Reference

### Classes

- enum **SquareType**

### Public Member Functions

- [Square](#) (SquareType type)
- SquareType [getType](#) ()
- [Tile](#) [getTile](#) ()
- Boolean [isEmpty](#) ()
- SquareType [setNormal](#) ()
- void [setTile](#) ([Tile](#) tile)
- String [toString](#) ()

### 4.15.1 Detailed Description

The [Square](#) Class represents the square on the Scrabble [Board](#) as objects

### 4.15.2 Constructor & Destructor Documentation

#### 4.15.2.1 Square()

```
scrabble.Square.Square (
    SquareType type )
```

[Square](#) Constructor

#### Parameters

<i>type</i>	The SquareType of the <a href="#">Square</a>
-------------	--

### 4.15.3 Member Function Documentation

#### 4.15.3.1 `getTile()`

```
Tile scrabble.Square.getTile ( )
```

Accessor Method for the [Tile](#) on the [Square](#)

**Returns**

The [Tile](#) on the [Square](#)

#### 4.15.3.2 `getType()`

```
SquareType scrabble.Square.getType ( )
```

Accessor Method for the `SquareType` of the [Square](#)

**Returns**

The `SquareType` of the [Square](#)

#### 4.15.3.3 `isEmpty()`

```
Boolean scrabble.Square.isEmpty ( )
```

Method to find if the [Square](#) has a [Tile](#)

**Returns**

True if the [Square](#) has no [Tile](#) on it

#### 4.15.3.4 `setNormal()`

```
SquareType scrabble.Square.setNormal ( )
```

Method to set the `SquareType` of a [Square](#) to NORMAL

**Returns**

The `SquareType` of the [Square](#) before setting to normal

#### 4.15.3.5 `setTile()`

```
void scrabble.Square.setTile (
    Tile tile )
```

Mutator Method for [Tile](#) on [Square](#)

## Parameters

<i>tile</i>	Tile to be placed on the Square
-------------	---------------------------------

## Exceptions

<i>InvalidSquareException</i>	The Square has a Tile on it already or the Tile is a null Tile
-------------------------------	--

#### 4.15.3.6 toString()

```
String scrabble.Square.toString ( )
```

toString Method for Square

## Returns

the String form of Square

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

- src/main/java/scrabble/Square.java

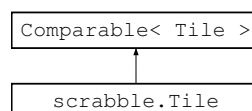
## 4.16 scrabble.SquareTest Class Reference

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

- src/test/java/scrabble/SquareTest.java

## 4.17 scrabble.Tile Class Reference

Inheritance diagram for scrabble.Tile:



## Public Member Functions

- int [compareTo](#) ([Tile](#) t)
- boolean [equals](#) (Object obj)
- short [getValue](#) ()
- char [getCharacter](#) ()
- void [setCharacter](#) (char character) throws [InvalidTileException](#)
- void [setNull](#) ()
- [Tile](#) (char c) throws [InvalidTileException](#)
- String [toString](#) ()

## Static Public Member Functions

- static void [main](#) (String[] args)

### 4.17.1 Detailed Description

Class that represent a Character [Tile](#) for the Game of Scrabble

### 4.17.2 Constructor & Destructor Documentation

#### 4.17.2.1 [Tile](#)()

```
scrabble.Tile.Tile (
    char c ) throws InvalidTileException
```

Constructor for [Tile](#)

Parameters

<a href="#">c</a>	Character for the <a href="#">Tile</a>
-------------------	--

Exceptions

<a href="#">InvalidTileException</a>	If an invalid character is inputted
--------------------------------------	-------------------------------------

### 4.17.3 Member Function Documentation

#### 4.17.3.1 [compareTo](#)()

```
int scrabble.Tile.compareTo (
    Tile t )
```

Method to compare [Tile](#) objects by Character then Value

#### Parameters

<i>t</i>	<a href="#">Tile</a> for this tile to be compared to
----------	--

#### Returns

Returns 0 if equal. Returns 1 if this Character is greater or Characters are equals and this Value is greater else returns -1

### 4.17.3.2 equals()

```
boolean scrabble.Tile.equals (
    Object obj )
```

Method to see if this [Tile](#) equals another object

#### Parameters

<i>obj</i>	Object to be compared to
------------	--------------------------

#### Returns

Returns True if the objects are equal

### 4.17.3.3 getCharacter()

```
char scrabble.Tile.getCharacter ( )
```

Accessor method for character of the [Tile](#)

#### Returns

Char of the [Tile](#)

### 4.17.3.4 getValue()

```
short scrabble.Tile.getValue ( )
```

Accessor method for value of the tile

#### Returns

Short value of the tile

#### 4.17.3.5 main()

```
static void scrabble.Tile.main (
    String[] args ) [static]
```

##### Main Function

##### Parameters

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

#### 4.17.3.6 setCharacter()

```
void scrabble.Tile.setCharacter (
    char character ) throws InvalidTileException
```

Method to change the character of a blank [Tile](#)

##### Parameters

<i>character</i>	New char for the <a href="#">Tile</a>
------------------	---------------------------------------

##### Exceptions

<i>InvalidTileException</i>	If an invalid char inputted or the <a href="#">Tile</a> is not blank
-----------------------------	--

#### 4.17.3.7 setNull()

```
void scrabble.Tile.setNull ( )
```

Method to set Blank Tiles Character back to null

##### Exceptions

<i>InvalidTileException</i>	If the tile is not a blank tile
-----------------------------	---------------------------------

#### 4.17.3.8 toString()

```
String scrabble.Tile.toString ( )
```

toString Method for [Tile](#)



#### Returns

The string of [Tile](#)

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

- `src/main/java/scrabble/Tile.java`



# Index

- addTile
  - scrabble.Frame, 13
- Board
  - scrabble.Board, 7
- BOARD\_SIZE
  - scrabble.Board, 12
- checkPlayerHasTiles
  - scrabble.Board, 8
- checkPositionContainsTile
  - scrabble.Board, 8
- checkPositionLine
  - scrabble.Board, 8
- checkTiles
  - scrabble.Frame, 13, 15
- checkValidMove
  - scrabble.Board, 9
- checkValidPosition
  - scrabble.Board, 9
- checkWordConnects
  - scrabble.Board, 9
- checkWordLength
  - scrabble.Board, 10
- compareTo
  - scrabble.Tile, 32
- decreaseScore
  - scrabble.Player, 24
- equals
  - scrabble.Tile, 33
- fillFrame
  - scrabble.Frame, 15
- Frame
  - scrabble.Frame, 13
- getBoardSquares
  - scrabble.Board, 10
- getCharacter
  - scrabble.Tile, 33
- getName
  - scrabble.Player, 24
- getPlayerFrame
  - scrabble.Player, 24
- getScore
  - scrabble.Player, 25
- getSquare
  - scrabble.Board, 10
- getTile
  - scrabble.Frame, 15, 16
  - scrabble.Square, 30
- getTiles
  - scrabble.Frame, 16
- getType
  - scrabble.Square, 30
- getValue
  - scrabble.Tile, 33
- increaseScore
  - scrabble.Player, 25
- isEmpty
  - scrabble.Frame, 16
  - scrabble.Pool, 27
  - scrabble.Square, 30
- main
  - scrabble.Tile, 33
- placeTile
  - scrabble.Board, 11
- placeTiles
  - scrabble.Board, 11
- Player
  - scrabble.Player, 23
- playerReset
  - scrabble.Player, 25
- Pool
  - scrabble.Pool, 27
- poolFill
  - scrabble.Pool, 27
- receiveTile
  - scrabble.Pool, 27
- removeTile
  - scrabble.Frame, 16, 18
  - scrabble.Pool, 28
- removeTiles
  - scrabble.Frame, 18, 19
- resetBoard
  - scrabble.Board, 11
- returnFrame
  - scrabble.Frame, 19
- scrabble.Board, 7
  - Board, 7
  - BOARD\_SIZE, 12
  - checkPlayerHasTiles, 8
  - checkPositionContainsTile, 8
  - checkPositionLine, 8
  - checkValidMove, 9

- checkValidPosition, 9
- checkWordConnects, 9
- checkWordLength, 10
- getBoardSquares, 10
- getSquare, 10
- placeTile, 11
- placeTiles, 11
- resetBoard, 11
- toString, 11
- scrabble.BoardTest, 12
- scrabble.exceptions.InvalidBoardException, 20
- scrabble.exceptions.InvalidFrameException, 20
- scrabble.exceptions.InvalidPlayerNameException, 21
- scrabble.exceptions.InvalidPlayerScoreException, 21
- scrabble.exceptions.InvalidPoolException, 22
- scrabble.exceptions.InvalidSquareException, 22
- scrabble.exceptions.InvalidTileException, 22
- scrabble.Frame, 12
  - addTile, 13
  - checkTiles, 13, 15
  - fillFrame, 15
  - Frame, 13
  - getTile, 15, 16
  - getTiles, 16
  - isEmpty, 16
  - removeTile, 16, 18
  - removeTiles, 18, 19
  - returnFrame, 19
  - swapTiles, 19
  - toString, 20
- scrabble.Main, 23
- scrabble.Player, 23
  - decreaseScore, 24
  - getName, 24
  - getPlayerFrame, 24
  - getScore, 25
  - increaseScore, 25
  - Player, 23
  - playerReset, 25
  - setName, 26
  - toString, 26
- scrabble.Pool, 26
  - isEmpty, 27
  - Pool, 27
  - poolFill, 27
  - receiveTile, 27
  - removeTile, 28
  - tilesInPool, 28
  - toString, 28
- scrabble.PoolTest, 29
- scrabble.Square, 29
  - getTile, 30
  - getType, 30
  - isEmpty, 30
  - setNormal, 30
  - setTile, 30
  - Square, 29
  - toString, 31
- scrabble.SquareTest, 31
- scrabble.Tile, 31
  - compareTo, 32
  - equals, 33
  - getCharacter, 33
  - getValue, 33
  - main, 33
  - setCharacter, 34
  - setNull, 34
  - Tile, 32
  - toString, 34
- setCharacter
  - scrabble.Tile, 34
- setName
  - scrabble.Player, 26
- setNormal
  - scrabble.Square, 30
- setNull
  - scrabble.Tile, 34
- setTile
  - scrabble.Square, 30
- Square
  - scrabble.Square, 29
- swapTiles
  - scrabble.Frame, 19
- Tile
  - scrabble.Tile, 32
- tilesInPool
  - scrabble.Pool, 28
- toString
  - scrabble.Board, 11
  - scrabble.Frame, 20
  - scrabble.Player, 26
  - scrabble.Pool, 28
  - scrabble.Square, 31
  - scrabble.Tile, 34