

## Assignment 3

Generated by Doxygen 1.8.17



<b>1 Namespace Index</b>	<b>1</b>
1.1 Packages	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 Namespace Documentation</b>	<b>7</b>
4.1 Package src	7
4.1.1 Detailed Description	7
<b>5 Class Documentation</b>	<b>9</b>
5.1 src.DemT Class Reference	9
5.1.1 Detailed Description	10
5.1.2 Constructor & Destructor Documentation	10
5.1.2.1 DemT()	10
5.1.3 Member Function Documentation	10
5.1.3.1 ascendingRows()	10
5.1.3.2 max()	11
5.1.3.3 total()	11
5.2 src.LanduseMapT Class Reference	12
5.2.1 Detailed Description	12
5.2.2 Constructor & Destructor Documentation	12
5.2.2.1 LanduseMapT()	13
5.3 src.LuT Enum Reference	13
5.3.1 Detailed Description	13
5.4 src.PointT Class Reference	13
5.4.1 Detailed Description	14
5.4.2 Constructor & Destructor Documentation	14
5.4.2.1 PointT()	14
5.4.3 Member Function Documentation	14
5.4.3.1 col()	14
5.4.3.2 row()	15
5.4.3.3 translate()	15
5.5 src.Seq2D< T > Class Template Reference	15
5.5.1 Detailed Description	16
5.5.2 Constructor & Destructor Documentation	16
5.5.2.1 Seq2D()	16
5.5.3 Member Function Documentation	16
5.5.3.1 area()	16
5.5.3.2 count()	18
5.5.3.3 countRow()	18

5.5.3.4 <code>get()</code> . . . . .	19
5.5.3.5 <code>getNumCol()</code> . . . . .	19
5.5.3.6 <code>getNumRow()</code> . . . . .	19
5.5.3.7 <code>getScale()</code> . . . . .	19
5.5.3.8 <code>set()</code> . . . . .	19

<b>Index</b>	<b>21</b>
--------------	-----------

# Chapter 1

## Namespace Index

### 1.1 Packages

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

<a href="#">src</a> . . . . .	<a href="#">7</a>
-------------------------------	-------------------



## Chapter 2

# Hierarchical Index

### 2.1 Class Hierarchy

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

src.LuT . . . . .	13
src.PointT . . . . .	13
src.Seq2D< T > . . . . .	15
src.Seq2D< Integer > . . . . .	15
src.DemT . . . . .	9
src.Seq2D< LuT > . . . . .	15
src.LanduseMapT . . . . .	12





## Chapter 3

# Class Index

### 3.1 Class List

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

<a href="#">src.DemT</a>	An ADT that represents the grid ( <a href="#">Seq2D</a> ) with type Integer . . . . .	9
<a href="#">src.LanduseMapT</a>	An ADT that represents Landuse types on the grid ( <a href="#">Seq2D</a> ) . . . . .	12
<a href="#">src.LuT</a>	A enum class to represent Land use types . . . . .	13
<a href="#">src.PointT</a>	An ADT that represents a point . . . . .	13
<a href="#">src.Seq2D&lt; T &gt;</a>	An ADT that represents a matrix . . . . .	15



## Chapter 4

# Namespace Documentation

### 4.1 Package src

#### Classes

- class [DemT](#)  
*An ADT that represents the grid ([Seq2D](#)) with type Integer.*
- class [LanduseMapT](#)  
*An ADT that represents Landuse types on the grid ([Seq2D](#))*
- enum [LuT](#)  
*A enum class to represent Land use types.*
- class [PointT](#)  
*An ADT that represents a point.*
- class [Seq2D](#)  
*An ADT that represents a matrix.*

#### 4.1.1 Detailed Description

Author: Dhruv Bhavsar Revised: March 16 2020 Description: Enum class for Landuse Types

Author: Dhruv Bhavsar Revised: March 16 2020 Description: [PointT](#) ADT class

Author: Dhruv Bhavsar Revised: March 16 2020 Description: [Seq2D](#) ADT class

Author: Dhruv Bhavsar Revised: March 16 2020 Description: [LanduseMapT](#) ADT class

Author: Dhruv Bhavsar Revised: March 16 2020 Description: [DemT](#) ADT class



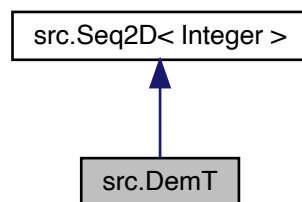
## Chapter 5

# Class Documentation

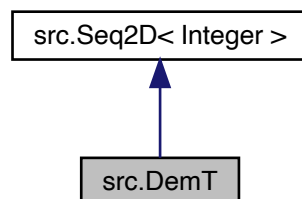
### 5.1 src.DemT Class Reference

An ADT that represents the grid ([Seq2D](#)) with type Integer.

Inheritance diagram for src.DemT:



Collaboration diagram for src.DemT:



## Public Member Functions

- [DemT](#) (ArrayList< ArrayList< Integer >> S, double scl)  
*Constructor that creates a new [DemT](#) object.*
- Integer [total](#) ()  
*Method that calculate the total of all the values in all of the cells.*
- Integer [max](#) ()  
*Method that finds the maximum value in the 2d grid of integers.*
- boolean [ascendingRows](#) ()  
*Method to find out if the total in each row is increasing as the row number increases.*

### 5.1.1 Detailed Description

An ADT that represents the grid ([Seq2D](#)) with type Integer.

### 5.1.2 Constructor & Destructor Documentation

#### 5.1.2.1 DemT()

```
src.DemT.DemT (
    ArrayList< ArrayList< Integer >> S,
    double scl )
```

Constructor that creates a new [DemT](#) object.

Same exceptions as [Seq2D](#) apply here, using super to call the parent constructor

#### Parameters

<i>S</i>	The ArrayList in a Arraylist of type Integer
<i>scl</i>	type double, scale - length of each side of each cell in the grid

### 5.1.3 Member Function Documentation

#### 5.1.3.1 ascendingRows()

```
boolean src.DemT.ascendingRows ( )
```

Method to find out if the total in each row is increasing as the row number increases.

If num of rows is 1, then its not ascending.

**Returns**

True if the rows are ascending else false

**5.1.3.2 max()**

```
Integer src.DemT.max ( )
```

Method that finds the maximum value in the 2d grid of integers.

Loop over the number of rows and number of columns, make a new pointT and use the get method to find value and compare it with the current max

**Returns**

Integer - maximum value in the grid

**5.1.3.3 total()**

```
Integer src.DemT.total ( )
```

Method that calculate the total of all the values in all of the cells.

Loop over the number of rows and number of columns, make a new pointT and use the get method to find value

**Returns**

Integer - total of all cells' values

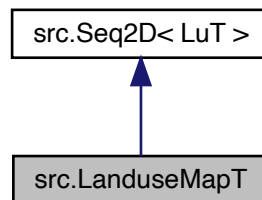
The documentation for this class was generated from the following file:

- src/DemT.java

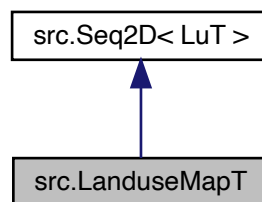
## 5.2 src.LanduseMapT Class Reference

An ADT that represents Landuse types on the grid ([Seq2D](#))

Inheritance diagram for src.LanduseMapT:



Collaboration diagram for src.LanduseMapT:



### Public Member Functions

- [LanduseMapT](#) (ArrayList< ArrayList< [LuT](#) >> S, double scl)  
*Constructor that creates a new [LanduseMapT](#).*

### 5.2.1 Detailed Description

An ADT that represents Landuse types on the grid ([Seq2D](#))

### 5.2.2 Constructor & Destructor Documentation



### 5.2.2.1 LanduseMapT()

```
src.LanduseMapT.LanduseMapT (
    ArrayList< ArrayList< LuT >> S,
    double scl )
```

Constructor that creates a new [LanduseMapT](#).

Same exceptions as [Seq2D](#) apply here, using super to call the parent constructor

#### Parameters

<i>S</i>	The ArrayList in a Arraylist of type <a href="#">LuT</a>
<i>scl</i>	type double, scale - length of each side of each cell in the grid

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

- src/LanduseMapT.java

## 5.3 src.LuT Enum Reference

A enum class to represent Land use types.

### Public Attributes

- **R**
- **T**
- **A**
- **C**

### 5.3.1 Detailed Description

A enum class to represent Land use types.

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

- src/LuT.java

## 5.4 src.PointT Class Reference

An ADT that represents a point.

## Public Member Functions

- [PointT](#) (int [row](#), int [col](#))  
*Constructor to create a [PointT](#) object.*
- int [row](#) ()  
*Getter for row variable.*
- int [col](#) ()  
*Getter for column variable.*
- [PointT translate](#) (int dr, int dc)  
*Method to translate the current point by given row and column parameters.*

### 5.4.1 Detailed Description

An ADT that represents a point.

### 5.4.2 Constructor & Destructor Documentation

#### 5.4.2.1 PointT()

```
src.PointT.PointT (  
    int row,  
    int col )
```

Constructor to create a [PointT](#) object.

##### Parameters

<i>row</i>	Takes the row number as an int
<i>col</i>	Takes the col number as an int

### 5.4.3 Member Function Documentation

#### 5.4.3.1 col()

```
int src.PointT.col ( )
```

Getter for column variable.

##### Returns

column variable (int)

### 5.4.3.2 row()

```
int src.PointT.row ( )
```

Getter for row variable.

#### Returns

row variable (int)

### 5.4.3.3 translate()

```
PointT src.PointT.translate (
    int dr,
    int dc )
```

Method to translate the current point by given row and column parameters.

#### Parameters

<i>dr</i>	Change in row position (int)
<i>dc</i>	Change in column position (int)

#### Returns

New [PointT](#) with the new coordinates after translate

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

- src/PointT.java

## 5.5 src.Seq2D< T > Class Template Reference

An ADT that represents a matrix.

### Public Member Functions

- [Seq2D](#) (ArrayList< ArrayList< T >> S, double scl)  
*Constructor to create a new [Seq2D](#), number of columns must be the same for each row.*
- void [set](#) ([PointT](#) p, T v)  
*Method to set a specific element in the sequence to a value given by the parameter,.*
- T [get](#) ([PointT](#) p)  
*Method to get the value at the specified index by a [PointT](#).*
- int [getNumRow](#) ()  
*Getter to get the number of rows in the sequence.*

- int `getNumCol ()`  
*Getter to get the number of columns in the sequence.*
- double `getScale ()`  
*Getter to get the scale.*
- int `count (T t)`  
*Method to find out how many times a value exists in the sequence.*
- int `countRow (T t, int i)`  
*Method to count how many times the value t appears in row i.*
- double `area (T t)`  
*Method to calculate the total area in the grid taken up by cell value t.*

### 5.5.1 Detailed Description

An ADT that represents a matrix.

#### Parameters

<code>&lt;T&gt;</code>	Type T for generality
------------------------	-----------------------

### 5.5.2 Constructor & Destructor Documentation

#### 5.5.2.1 Seq2D()

```
src.Seq2D< T >.Seq2D (
    ArrayList< ArrayList< T >> S,
    double scl )
```

Constructor to create a new `Seq2D`, number of columns must be the same for each row.

#### Parameters

<code>S</code>	Arraylist of Arraylist, cannot be empty else exception is thrown
<code>scl</code>	Scale, cannot be less than 0 else exception is thrown

### 5.5.3 Member Function Documentation

#### 5.5.3.1 area()

```
double src.Seq2D< T >.area (
    T t )
```

Method to calculate the total area in the grid taken up by cell value t.

The length of each side of each cell is the scale hence the scale squared

**Parameters**

<i>t</i>	cell value to look for, type T
----------	--------------------------------

**Returns**

area type double

**5.5.3.2 count()**

```
int src.Seq2D< T >.count (
    T t )
```

Method to find out how many times a value exists in the sequence.

Loop over each row and call the countRow method to find the total count in that row

**Parameters**

<i>t</i>	Type T, the value to find in the sequence
----------	---

**Returns**

count Type int indicating how many times it appeared

**5.5.3.3 countRow()**

```
int src.Seq2D< T >.countRow (
    T t,
    int i )
```

Method to count how many times the value t appears in row i.

Check if the inputted row number is valid, loop over all the elements in the ith arraylist

**Parameters**

<i>t</i>	Type T, value to check
<i>i</i>	int row number

**Returns**

count - number of occurrences

#### 5.5.3.4 get()

```
T src.Seq2D< T >.get (
    PointT p )
```

Method to get the value at the specified index by a [PointT](#).

##### Parameters

<i>p</i>	<a href="#">PointT</a> , the specific index
----------	---

##### Returns

value of whats in the index Type T

#### 5.5.3.5 getNumCol()

```
int src.Seq2D< T >.getNumCol ( )
```

Getter to get the number of columns in the sequence.

##### Returns

int value of columns

#### 5.5.3.6 getNumRow()

```
int src.Seq2D< T >.getNumRow ( )
```

Getter to get the number of rows in the sequence.

##### Returns

int value of rows

#### 5.5.3.7 getScale()

```
double src.Seq2D< T >.getScale ( )
```

Getter to get the scale.

##### Returns

double value of scale

#### 5.5.3.8 set()

```
void src.Seq2D< T >.set (
    PointT p,
    T v )
```

Method to set a specific element in the sequence to a value given by the parameter,.  
if point doesn't exist in the sequence exception thrown `IndexOutOfBounds`

**Parameters**

$p$	<a href="#">PointT</a> , the specific element in the sequence
$v$	Type T, Set the element to

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

- `src/Seq2D.java`



# Index

- area
  - src.Seq2D< T >, 16
- ascendingRows
  - src.DemT, 10
- col
  - src.PointT, 14
- count
  - src.Seq2D< T >, 18
- countRow
  - src.Seq2D< T >, 18
- DemT
  - src.DemT, 10
- get
  - src.Seq2D< T >, 18
- getNumCol
  - src.Seq2D< T >, 19
- getNumRow
  - src.Seq2D< T >, 19
- getScale
  - src.Seq2D< T >, 19
- LanduseMapT
  - src.LanduseMapT, 12
- max
  - src.DemT, 11
- PointT
  - src.PointT, 14
- row
  - src.PointT, 14
- Seq2D
  - src.Seq2D< T >, 16
- set
  - src.Seq2D< T >, 19
- src, 7
  - src.DemT, 9
    - ascendingRows, 10
    - DemT, 10
    - max, 11
    - total, 11
  - src.LanduseMapT, 12
    - LanduseMapT, 12
  - src.LuT, 13
  - src.PointT, 13
    - col, 14
    - PointT, 14
    - row, 14
    - translate, 15
  - src.Seq2D< T >, 15
    - area, 16
    - count, 18
    - countRow, 18
    - get, 18
    - getNumCol, 19
    - getNumRow, 19
    - getScale, 19
    - Seq2D, 16
    - set, 19
- total
  - src.DemT, 11
- translate
  - src.PointT, 15