

TrainVagas

0.1

Generated by Doxygen 1.8.13

Contents

- 1 Namespace Index** **1**
 - 1.1 Packages 1

- 2 Hierarchical Index** **3**
 - 2.1 Class Hierarchy 3

- 3 Class Index** **5**
 - 3.1 Class List 5

- 4 File Index** **7**
 - 4.1 File List 7

- 5 Namespace Documentation** **9**
 - 5.1 Package controller 9
 - 5.2 Package domain 9
 - 5.3 Package frontend 9
 - 5.4 Package software 9
 - 5.5 Package test 10
 - 5.6 Package translate 10
 - 5.7 Package utilities 10

6 Class Documentation	11
6.1 domain.CalculateDistance Class Reference	11
6.1.1 Member Function Documentation	12
6.1.1.1 calculate()	12
6.1.1.2 calculateAll()	13
6.1.2 Member Data Documentation	13
6.1.2.1 graph	13
6.2 test.CalculateDistanceTest Class Reference	14
6.2.1 Member Function Documentation	14
6.2.1.1 setUp()	14
6.2.1.2 testCalculate()	15
6.2.1.3 testCalculateAll()	15
6.2.2 Member Data Documentation	15
6.2.2.1 cd	15
6.3 utilities.DataFile Class Reference	16
6.3.1 Member Function Documentation	17
6.3.1.1 checkFileExist()	17
6.3.1.2 close()	17
6.3.1.3 getFileName()	18
6.3.1.4 open()	18
6.3.1.5 readCondition()	19
6.3.1.6 readDouble()	19
6.3.1.7 readInteger()	20
6.3.1.8 readList() [1/2]	21
6.3.1.9 readList() [2/2]	22
6.3.1.10 readLiteralOperand()	23
6.3.1.11 setFileName()	24
6.3.2 Member Data Documentation	24
6.3.2.1 DISTANCE_CONDITION	25
6.3.2.2 file_name	25

6.3.2.3	FILE_TESTS_CASES	25
6.3.2.4	input	25
6.3.2.5	PATH_FILE	25
6.3.2.6	PATTERN_CONDITION	25
6.3.2.7	PATTERN_DOUBLE	25
6.3.2.8	PATTERN_INTEGER	25
6.3.2.9	PATTERN_LIST	26
6.3.2.10	PATTERN_OPERAND	26
6.3.2.11	prop	26
6.3.2.12	SEPARATOR	26
6.3.2.13	STOP1_CONDITION	26
6.3.2.14	STOP2_CONDITION	26
6.4	test.DataFileTest Class Reference	26
6.4.1	Member Function Documentation	27
6.4.1.1	setUp()	27
6.4.1.2	testOpen()	27
6.4.1.3	testReadConditionStringInt()	27
6.4.1.4	testReadDoubleStringInt()	28
6.4.1.5	testReadIntegerStringInt()	28
6.4.1.6	testReadListString()	28
6.4.1.7	testReadListStringInt()	29
6.4.1.8	testReadLiteralOperandStringInt()	29
6.5	frontend.Display Class Reference	29
6.5.1	Member Function Documentation	30
6.5.1.1	printHead()	30
6.5.1.2	printLine()	30
6.5.1.3	printResultCalculateDistance()	31
6.5.1.4	printResultFilter()	31
6.5.1.5	printResultShortest()	32
6.5.1.6	showAllResults()	33

6.5.2	Member Data Documentation	33
6.5.2.1	FORMAT_COL1	33
6.5.2.2	FORMAT_DECIMAL	34
6.5.2.3	FORMAT_DOUBLE	34
6.5.2.4	FORMAT_ROUTE	34
6.5.2.5	FORMAT_STRING	34
6.5.2.6	index_line	34
6.5.2.7	NEW_LINE	34
6.5.2.8	SEPARATOR	34
6.6	utilities.FileProperty Enum Reference	35
6.6.1	Constructor & Destructor Documentation	35
6.6.1.1	FileProperty()	35
6.6.2	Member Function Documentation	35
6.6.2.1	getName()	35
6.6.3	Member Data Documentation	36
6.6.3.1	DISTANCE_ROUTES	36
6.6.3.2	FILTER_CONDITION	36
6.6.3.3	FILTER_ROUTES	36
6.6.3.4	GRAPH_ROUTES	36
6.6.3.5	name	37
6.6.3.6	SHORTEST_ROUTES	37
6.7	domain.Filter Class Reference	37
6.7.1	Member Function Documentation	38
6.7.1.1	filterBy()	38
6.7.2	Member Data Documentation	39
6.7.2.1	graph	39
6.8	test.FilterTest Class Reference	39
6.8.1	Member Function Documentation	40
6.8.1.1	setUp()	40
6.8.1.2	testFilterBy()	40

6.8.2	Member Data Documentation	40
6.8.2.1	filter	40
6.9	domain.Graph Class Reference	41
6.9.1	Constructor & Destructor Documentation	42
6.9.1.1	Graph()	42
6.9.2	Member Function Documentation	42
6.9.2.1	addRoute() [1/2]	42
6.9.2.2	addRoute() [2/2]	43
6.9.2.3	addTown()	44
6.9.2.4	calculateDistance()	45
6.9.2.5	clear()	46
6.9.2.6	containsTown()	46
6.9.2.7	create()	47
6.9.2.8	getAllRoutePossible()	48
6.9.2.9	getAllRoutes()	49
6.9.2.10	getAllTowns()	49
6.9.2.11	getConnections()	49
6.9.2.12	getRoute()	50
6.9.2.13	isRouteAvailable()	51
6.9.2.14	loadGraphRoutesFromFile()	52
6.9.3	Member Data Documentation	53
6.9.3.1	graph	53
6.9.3.2	mapRoutes	53
6.9.3.3	mapTowns	53
6.10	test.GraphTest Class Reference	54
6.10.1	Member Function Documentation	54
6.10.1.1	setUp()	55
6.10.1.2	testAddRoute()	55
6.10.1.3	testAddTown()	55
6.10.1.4	testcalculateDistance()	56

6.10.1.5	testclear()	56
6.10.1.6	testContainsTown()	56
6.10.1.7	testGetAllRoutePossible()	57
6.10.1.8	testGetAllRoutes()	57
6.10.1.9	testGetAllTowns()	58
6.10.1.10	testGetConnections()	58
6.10.1.11	testGetRoute()	58
6.10.1.12	testGraph()	59
6.10.1.13	testIsRouteAvailable()	59
6.10.1.14	testLoadGraphRoutesFromFile()	60
6.10.2	Member Data Documentation	60
6.10.2.1	graph	60
6.11	software.main Class Reference	60
6.11.1	Constructor & Destructor Documentation	60
6.11.1.1	main()	61
6.11.2	Member Function Documentation	61
6.11.2.1	runSW()	61
6.11.2.2	shutdownSW()	62
6.11.2.3	startupSW()	62
6.12	utilities.Permutation Class Reference	63
6.12.1	Member Function Documentation	63
6.12.1.1	getAllLists()	63
6.13	test.PermutationTest Class Reference	64
6.13.1	Member Function Documentation	64
6.13.1.1	setUp()	64
6.13.1.2	testGetAllLists()	65
6.14	translate.PtBR Class Reference	65
6.14.1	Member Function Documentation	66
6.14.1.1	FILE_NOT_FOUND()	66
6.14.1.2	INCORRECT_PARAMETERS()	66

6.14.1.3	PROPERTY_NOT_FOUND()	66
6.14.1.4	ROUTE_NO_AVAILABLE()	67
6.14.1.5	TOW_NO_EXIST()	67
6.14.2	Member Data Documentation	67
6.14.2.1	FILE_NOT_FOUND	67
6.14.2.2	INCORRECT_PARAMETERS	67
6.14.2.3	PROPERTY_NOT_FOUND	67
6.14.2.4	ROUTE_NO_AVAILABLE	67
6.14.2.5	TOW_NO_EXIST	68
6.15	controller.RailSystem Class Reference	68
6.15.1	Member Function Documentation	68
6.15.1.1	calculateDistance()	68
6.15.1.2	delnit()	69
6.15.1.3	filterByDistance()	69
6.15.1.4	filterByStops1()	70
6.15.1.5	filterByStops2()	70
6.15.1.6	init()	71
6.15.1.7	shortestRoute()	72
6.16	test.RailSystemTest Class Reference	73
6.16.1	Member Function Documentation	73
6.16.1.1	setUp()	73
6.16.1.2	testCalculateDistance()	73
6.16.1.3	testFilterByStops1()	74
6.16.1.4	testFilterByStops2()	74
6.16.1.5	testShortestRoute()	74
6.17	domain.Route Class Reference	75
6.17.1	Constructor & Destructor Documentation	76
6.17.1.1	Route() [1/3]	76
6.17.1.2	Route() [2/3]	76
6.17.1.3	Route() [3/3]	77

6.17.2	Member Function Documentation	77
6.17.2.1	formatKey()	77
6.17.2.2	getDistance()	78
6.17.2.3	getEndingTown()	79
6.17.2.4	getKey()	79
6.17.2.5	getKeyOfStop()	80
6.17.2.6	getNumberOfNodes()	80
6.17.2.7	getNumberOfStops()	81
6.17.2.8	getStartingTown()	81
6.17.2.9	isAvailable()	82
6.17.2.10	setAvailable()	82
6.17.2.11	setKey()	83
6.17.2.12	setRoute()	84
6.17.2.13	setTotalDistance() [1/2]	85
6.17.2.14	setTotalDistance() [2/2]	86
6.17.2.15	testCondition()	86
6.17.3	Member Data Documentation	87
6.17.3.1	available	87
6.17.3.2	key	87
6.17.3.3	MINIMAL_SIZE_OF_ID	88
6.17.3.4	PATTERN_DOUBLE	88
6.17.3.5	PATTERN_ID	88
6.17.3.6	PATTERN_NO_DUPLICATE	88
6.17.3.7	totalDistance	88
6.18	test.RouteTest Class Reference	88
6.18.1	Member Function Documentation	89
6.18.1.1	setUp()	89
6.18.1.2	testFormatKey()	89
6.18.1.3	testGetDistance()	90
6.18.1.4	testGetEndingTown()	90

6.18.1.5	testGetId()	90
6.18.1.6	testGetIdOfStop()	91
6.18.1.7	testGetNumberOfNodes()	91
6.18.1.8	testGetNumberOfStops()	91
6.18.1.9	testGetStartingTown()	92
6.18.1.10	testIsAvailable()	92
6.18.1.11	testRoute()	92
6.18.1.12	testRouteString()	93
6.18.1.13	testRouteStringDouble()	93
6.18.1.14	testSetAvailable()	94
6.18.1.15	testSetId()	94
6.18.1.16	testSetRoute()	94
6.18.1.17	testSetTotalDistanceDouble()	95
6.18.1.18	testSetTotalDistanceString()	95
6.18.1.19	testTestCondition()	95
6.18.2	Member Data Documentation	96
6.18.2.1	route	96
6.19	domain.ShortestRoute Class Reference	96
6.19.1	Member Function Documentation	96
6.19.1.1	getShortestRoute() [1/2]	97
6.19.1.2	getShortestRoute() [2/2]	97
6.19.2	Member Data Documentation	98
6.19.2.1	graph	98
6.20	test.ShortestRouteTest Class Reference	99
6.20.1	Member Function Documentation	99
6.20.1.1	setUp()	99
6.20.1.2	testGetShortestRoute()	100
6.20.1.3	testGetShortestRouteRoute()	100
6.20.2	Member Data Documentation	100
6.20.2.1	sr	100

6.21	domain.Town Class Reference	100
6.21.1	Constructor & Destructor Documentation	101
6.21.1.1	Town()	101
6.21.2	Member Function Documentation	101
6.21.2.1	formatKey()	101
6.21.2.2	getKey()	102
6.21.2.3	setKey()	103
6.21.3	Member Data Documentation	103
6.21.3.1	key	103
6.21.3.2	MINIMAL_SIZE_OF_ID	103
6.21.3.3	PATTERN_ID	104
6.22	test.TownTest Class Reference	104
6.22.1	Member Function Documentation	104
6.22.1.1	setUp()	104
6.22.1.2	testFormatKey()	105
6.22.1.3	testGetId()	105
6.22.1.4	testSetId()	105
6.22.1.5	testTown()	106
6.22.2	Member Data Documentation	106
6.22.2.1	town	106
6.23	translate.Translate Class Reference	106
6.23.1	Member Data Documentation	107
6.23.1.1	MESSAGE	107
6.24	translate.TranslateMessage Interface Reference	107
6.24.1	Member Function Documentation	107
6.24.1.1	FILE_NOT_FOUND()	108
6.24.1.2	INCORRECT_PARAMETERS()	108
6.24.1.3	PROPERTY_NOT_FOUND()	108
6.24.1.4	ROUTE_NO_AVAILABLE()	108
6.24.1.5	TOW_NO_EXIST()	109

7	File Documentation	111
7.1	D:/workspace/TrainVagas/src/controller/RailSystem.java File Reference	111
7.1.1	Detailed Description	111
7.2	D:/workspace/TrainVagas/src/domain/CalculateDistance.java File Reference	112
7.2.1	Detailed Description	112
7.3	D:/workspace/TrainVagas/src/domain/Filter.java File Reference	112
7.3.1	Detailed Description	113
7.4	D:/workspace/TrainVagas/src/domain/Graph.java File Reference	113
7.4.1	Detailed Description	113
7.5	D:/workspace/TrainVagas/src/domain/Route.java File Reference	114
7.5.1	Detailed Description	114
7.6	D:/workspace/TrainVagas/src/domain/ShortestRoute.java File Reference	114
7.6.1	Detailed Description	115
7.7	D:/workspace/TrainVagas/src/domain/Town.java File Reference	115
7.7.1	Detailed Description	115
7.8	D:/workspace/TrainVagas/src/frontend/Display.java File Reference	116
7.8.1	Detailed Description	116
7.9	D:/workspace/TrainVagas/src/software/main.java File Reference	116
7.9.1	Detailed Description	117
7.10	D:/workspace/TrainVagas/src/test/CalculateDistanceTest.java File Reference	117
7.11	D:/workspace/TrainVagas/src/test/DataFileTest.java File Reference	117
7.12	D:/workspace/TrainVagas/src/test/FilterTest.java File Reference	118
7.13	D:/workspace/TrainVagas/src/test/GraphTest.java File Reference	118
7.14	D:/workspace/TrainVagas/src/test/PermutationTest.java File Reference	118
7.15	D:/workspace/TrainVagas/src/test/RailSystemTest.java File Reference	118
7.16	D:/workspace/TrainVagas/src/test/RouteTest.java File Reference	119
7.17	D:/workspace/TrainVagas/src/test/ShortestRouteTest.java File Reference	119
7.18	D:/workspace/TrainVagas/src/test/TownTest.java File Reference	119
7.19	D:/workspace/TrainVagas/src/translate/PtBR.java File Reference	119
7.19.1	Detailed Description	120
7.20	D:/workspace/TrainVagas/src/translate/Translate.java File Reference	120
7.21	D:/workspace/TrainVagas/src/translate/TranslateMessage.java File Reference	121
7.21.1	Detailed Description	121
7.22	D:/workspace/TrainVagas/src/utilities/DataFile.java File Reference	121
7.23	D:/workspace/TrainVagas/src/utilities/FileProperty.java File Reference	121
7.24	D:/workspace/TrainVagas/src/utilities/Permutation.java File Reference	122
	Index	123

Chapter 1

Namespace Index

1.1 Packages

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

controller	9
domain	9
frontend	9
software	9
test	10
translate	10
utilities	10

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

- domain.CalculateDistance 11
- test.CalculateDistanceTest 14
- utilities.DataFile 16
- test.DataFileTest 26
- frontend.Display 29
- utilities.FileProperty 35
- domain.Filter 37
- test.FilterTest 39
- domain.Graph 41
- test.GraphTest 54
- software.main 60
- utilities.Permutation 63
- test.PermutationTest 64
- controller.RailSystem 68
- test.RailSystemTest 73
- domain.Route 75
- test.RouteTest 88
- domain.ShortestRoute 96
- test.ShortestRouteTest 99
- domain.Town 100
- test.TownTest 104
- translate.Translate 106
- translate.TranslateMessage 107
 - translate.PtBR 65

Chapter 3

Class Index

3.1 Class List

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

domain.CalculateDistance	11
test.CalculateDistanceTest	14
utilities.DataFile	16
test.DataFileTest	26
frontend.Display	29
utilities.FileProperty	35
domain.Filter	37
test.FilterTest	39
domain.Graph	41
test.GraphTest	54
software.main	60
utilities.Permutation	63
test.PermutationTest	64
translate.PtBR	65
controller.RailSystem	68
test.RailSystemTest	73
domain.Route	75
test.RouteTest	88
domain.ShortestRoute	96
test.ShortestRouteTest	99
domain.Town	100
test.TownTest	104
translate.Translate	106
translate.TranslateMessage	107

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

D:/workspace/TrainVagas/src/controller/ RailSystem.java	
Software de gerenciamento das rotas para uma solucao de mobilidade sobre trilhos	111
D:/workspace/TrainVagas/src/domain/ CalculateDistance.java	
Software de gerenciamento das rotas para uma solucao de mobilidade sobre trilhos	112
D:/workspace/TrainVagas/src/domain/ Filter.java	
Software de gerenciamento das rotas para uma solucao de mobilidade sobre trilhos	112
D:/workspace/TrainVagas/src/domain/ Graph.java	
Software de gerenciamento das rotas para uma solucao de mobilidade sobre trilhos	113
D:/workspace/TrainVagas/src/domain/ Route.java	
Software de gerenciamento das rotas para uma solucao de mobilidade sobre trilhos	114
D:/workspace/TrainVagas/src/domain/ ShortestRoute.java	
Software de gerenciamento das rotas para uma solucao de mobilidade sobre trilhos	114
D:/workspace/TrainVagas/src/domain/ Town.java	
Software de gerenciamento das rotas para uma solucao de mobilidade sobre trilhos	115
D:/workspace/TrainVagas/src/frontend/ Display.java	
Software de gerenciamento das rotas para uma solucao de mobilidade sobre trilhos	116
D:/workspace/TrainVagas/src/software/ main.java	
Software de gerenciamento das rotas para uma solucao de mobilidade sobre trilhos	116
D:/workspace/TrainVagas/src/test/ CalculateDistanceTest.java	117
D:/workspace/TrainVagas/src/test/ DataFileTest.java	117
D:/workspace/TrainVagas/src/test/ FilterTest.java	118
D:/workspace/TrainVagas/src/test/ GraphTest.java	118
D:/workspace/TrainVagas/src/test/ PermutationTest.java	118
D:/workspace/TrainVagas/src/test/ RailSystemTest.java	118
D:/workspace/TrainVagas/src/test/ RouteTest.java	119
D:/workspace/TrainVagas/src/test/ ShortestRouteTest.java	119
D:/workspace/TrainVagas/src/test/ TownTest.java	119
D:/workspace/TrainVagas/src/translate/ PtBR.java	
Software de gerenciamento das rotas para uma solucao de mobilidade sobre trilhos	119
D:/workspace/TrainVagas/src/translate/ Translate.java	120
D:/workspace/TrainVagas/src/translate/ TranslateMessage.java	
Software de gerenciamento das rotas para uma solucao de mobilidade sobre trilhos	121
D:/workspace/TrainVagas/src/utilities/ DataFile.java	121
D:/workspace/TrainVagas/src/utilities/ FileProperty.java	121
D:/workspace/TrainVagas/src/utilities/ Permutation.java	122

Chapter 5

Namespace Documentation

5.1 Package controller

Classes

- class [RailSystem](#)

5.2 Package domain

Classes

- class [CalculateDistance](#)
- class [Filter](#)
- class [Graph](#)
- class [Route](#)
- class [ShortestRoute](#)
- class [Town](#)

5.3 Package frontend

Classes

- class [Display](#)

5.4 Package software

Classes

- class [main](#)

5.5 Package test

Classes

- class [CalculateDistanceTest](#)
- class [DataFileTest](#)
- class [FilterTest](#)
- class [GraphTest](#)
- class [PermutationTest](#)
- class [RailSystemTest](#)
- class [RouteTest](#)
- class [ShortestRouteTest](#)
- class [TownTest](#)

5.6 Package translate

Classes

- class [PtBR](#)
- class [Translate](#)
- interface [TranslateMessage](#)

5.7 Package utilities

Classes

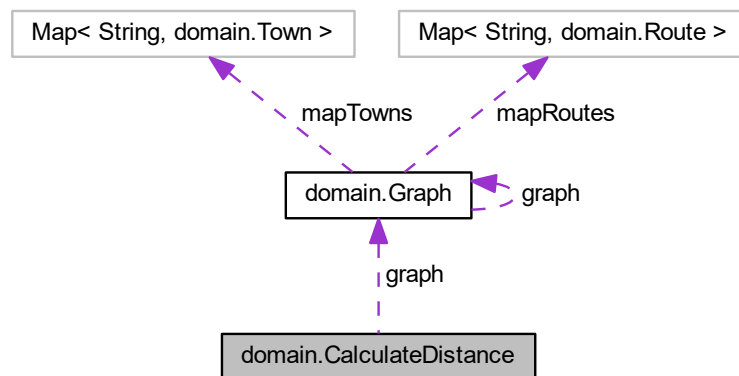
- class [DataFile](#)
- enum [FileProperty](#)
- class [Permutation](#)

Chapter 6

Class Documentation

6.1 domain.CalculateDistance Class Reference

Collaboration diagram for domain.CalculateDistance:



Public Member Functions

- [Route \[\] calculateAll \(\)](#)

Static Public Member Functions

- static [Route \[\] calculate \(String\[\] trips\)](#)

Static Package Attributes

- static [Graph graph = Graph.create\(\)](#)

6.1.1 Member Function Documentation

6.1.1.1 calculate()

```
static Route [] domain.CalculateDistance.calculate (
    String [] trips ) [static]
```

Calcula a distancia de uma lista de rotas(viagens)

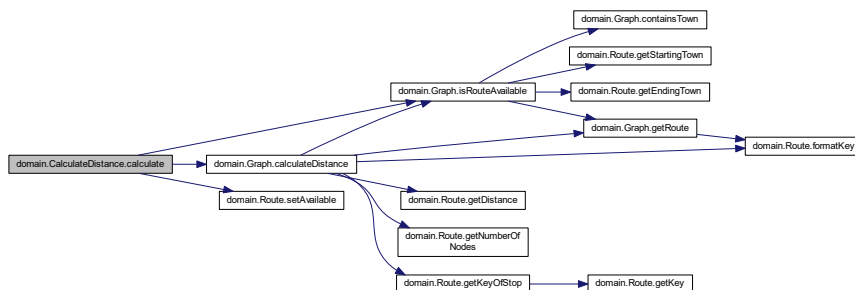
Parameters

<i>trips</i>	lista de key das rotas(viagens) para o calculo da distancia
--------------	---

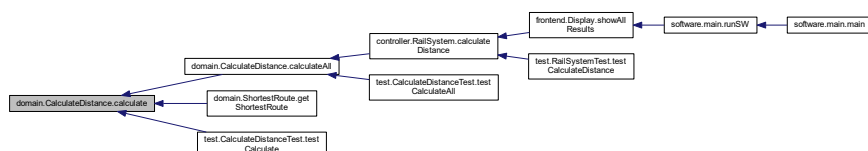
Returns

Rotas com suas respectivas distancias calculadas

Here is the call graph for this function:



Here is the caller graph for this function:



6.1.1.2 calculateAll()

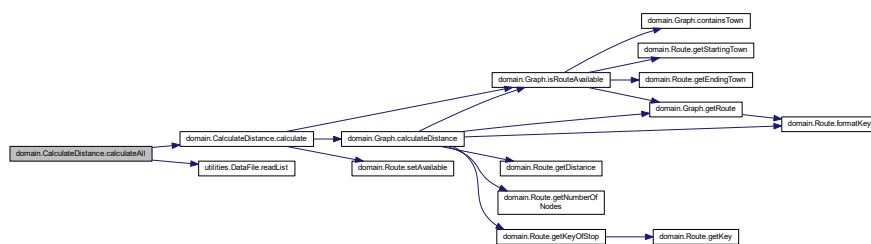
```
Route [] domain.CalculateDistance.calculateAll ( )
```

Baseada na lista de rotas contida no arquivo 'input.txt', na propriedade 'distance.routes', calcula a distancia de cada uma. Rotas nao possiveis serao setada como nao existente atraves da chamada do metodo setAvailabe(false) e o valor da distancia sera 0.0

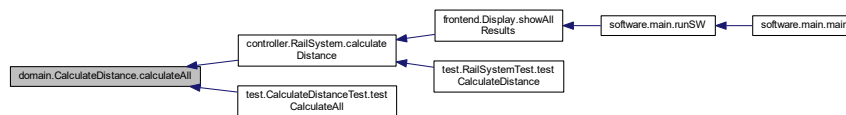
Returns

Retorna uma lista de rotas com sua respectiva distancia calculada e com a indicacao se e uma rota possivel

Here is the call graph for this function:



Here is the caller graph for this function:



6.1.2 Member Data Documentation

6.1.2.1 graph

```
Graph domain.CalculateDistance.graph = Graph.create() [static], [package]
```

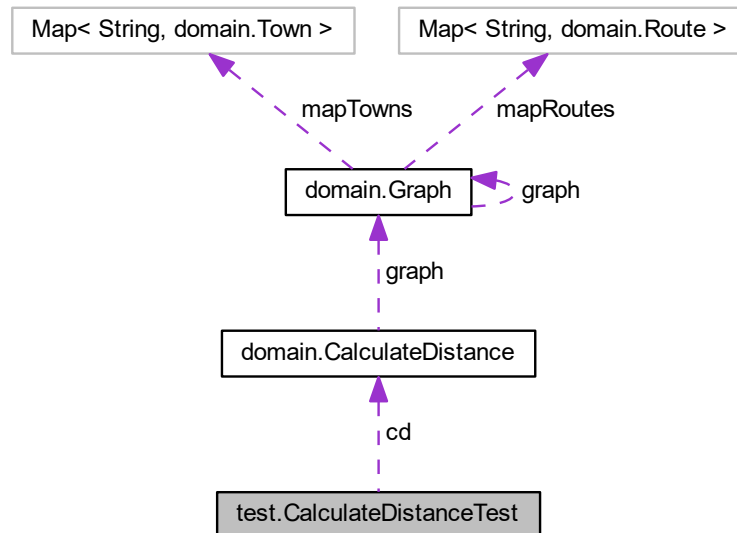
Rotas disponiveis no sistema

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

- [D:/workspace/TrainVagas/src/domain/CalculateDistance.java](#)

6.2 test.CalculateDistanceTest Class Reference

Collaboration diagram for test.CalculateDistanceTest:



Public Member Functions

- void [setUp](#) () throws Exception
- void [testCalculateAll](#) ()
- void [testCalculate](#) ()

Package Attributes

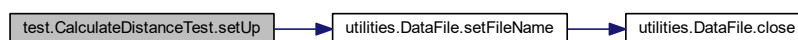
- [CalculateDistance](#) cd

6.2.1 Member Function Documentation

6.2.1.1 setUp()

```
void test.CalculateDistanceTest.setUp ( ) throws Exception
```

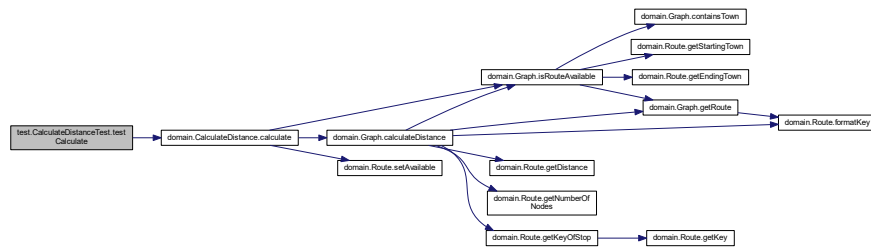
Here is the call graph for this function:



6.2.1.2 testCalculate()

```
void test.CalculateDistanceTest.testCalculate ( )
```

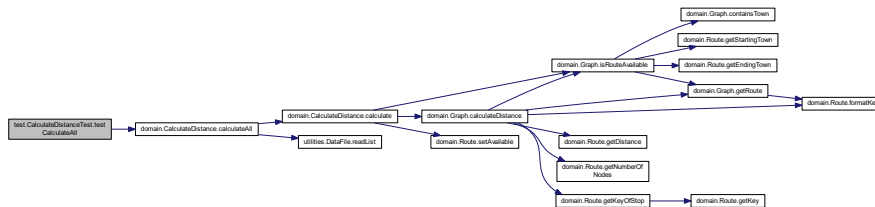
Here is the call graph for this function:



6.2.1.3 testCalculateAll()

```
void test.CalculateDistanceTest.testCalculateAll ( )
```

Here is the call graph for this function:



6.2.2 Member Data Documentation

6.2.2.1 cd

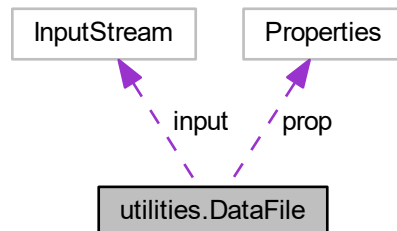
`CalculateDistance` test.CalculateDistanceTest.cd [package]

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

- [D:/workspace/TrainVagas/src/test/CalculateDistanceTest.java](#)

6.3 utilities.DataFile Class Reference

Collaboration diagram for utilities.DataFile:



Static Public Member Functions

- static void `setFileName` (String file)
- static String `getFileName` ()
- static boolean `open` ()
- static void `close` ()
- static String [] `readList` (FileProperty property)
- static String [] `readList` (FileProperty property, int index_filter)
- static String `readCondition` (FileProperty property, int index_filter)
- static double `readDouble` (FileProperty property, int index_filter)
- static int `readInteger` (FileProperty property, int index_filter)
- static String `readLiteralOperand` (FileProperty property, int index_filter)
- static boolean `checkFileExist` ()

Static Public Attributes

- static final int `DISTANCE_CONDITION` =3
- static final int `STOP1_CONDITION` =1
- static final int `STOP2_CONDITION` =2
- static final String `PATH_FILE` = "input.txt"
- static final String `FILE_TESTS_CASES` = "FILE_TEST_CASE.txt"

Static Package Attributes

- static Properties `prop` = new Properties()
- static InputStream `input` = null

Static Private Attributes

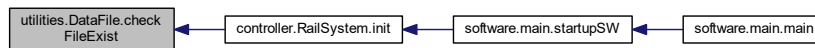
- static final String `PATTERN_LIST` = "[^a-zA-Z0-9.,]"
- static final String `PATTERN_OPERAND` = "[^a-zA-Z,]"
- static final String `PATTERN_DOUBLE` = "[^0-9.,]"
- static final String `PATTERN_INTEGER` = "[^0-9]"
- static final String `PATTERN_CONDITION` = "[^<>=]"
- static final String `SEPARATOR` = ","
- static String `file_name` = `PATH_FILE`

6.3.1 Member Function Documentation

6.3.1.1 checkFileExist()

```
static boolean utilities.DataFile.checkFileExist ( ) [static]
```

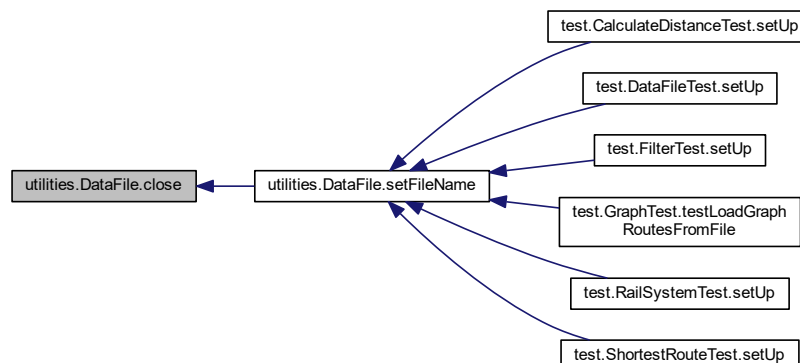
Here is the caller graph for this function:



6.3.1.2 close()

```
static void utilities.DataFile.close ( ) [static]
```

Fecha o arquivo de entrada Here is the caller graph for this function:



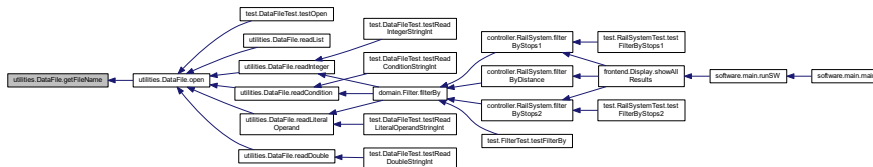
6.3.1.3 getFileName()

```
static String utilities.DataFile.getFileName ( ) [static]
```

Returns

Retorna nome do arquivo em uso

Here is the caller graph for this function:



6.3.1.4 open()

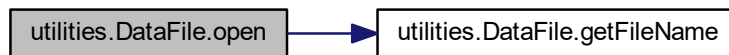
```
static boolean utilities.DataFile.open ( ) [static]
```

Abre o arquivo de entrada

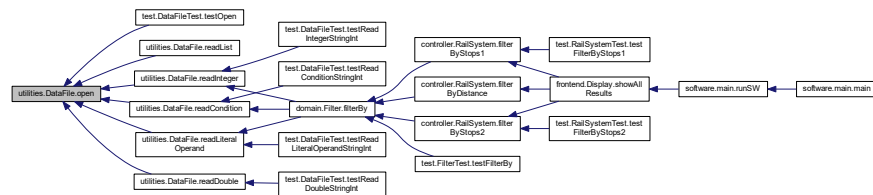
Returns

Se o arquivo foi aberto com sucesso retorna true

Here is the call graph for this function:



Here is the caller graph for this function:



6.3.1.5 readCondition()

```
static String utilities.DataFile.readCondition (
    FileProperty property,
    int index_filter ) [static]
```

Le uma condicao do arquivo de entrada

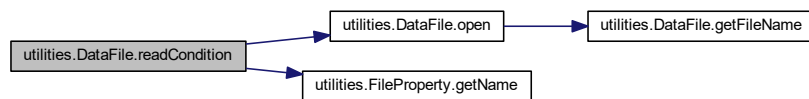
Parameters

<i>property</i>	propriedade onde a condicao sera buscado
<i>index_filter</i>	index da propriedade a ser buscada ex: propriedade[index].condition

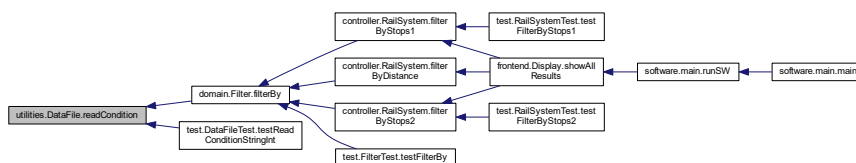
Returns

retorna a condicao literal lido do arquivo

Here is the call graph for this function:



Here is the caller graph for this function:



6.3.1.6 readDouble()

```
static double utilities.DataFile.readDouble (
    FileProperty property,
    int index_filter ) [static]
```

Le um double do arquivo de entrada

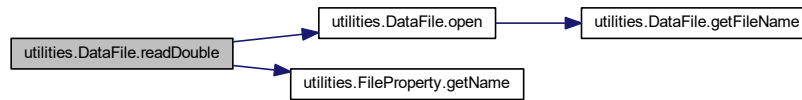
Parameters

<i>property</i>	propriedade onde o double sera buscado
<i>index_filter</i>	index identifica qual propriedade dentro do array que existe no arquivo

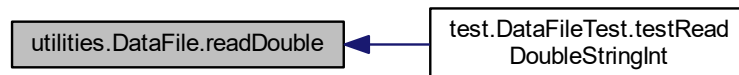
Returns

retorna o double lido do arquivo

Here is the call graph for this function:



Here is the caller graph for this function:

**6.3.1.7 readInteger()**

```

static int utilities.DataFile.readInteger (
    FileProperty property,
    int index_filter ) [static]
  
```

Le um inteiro do arquivo de entrada

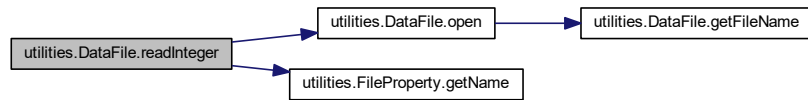
Parameters

<i>property</i>	propriedade onde o inteiro sera buscado
<i>index_filter</i>	index identifica qual propriedade dentro do array que existe no arquivo

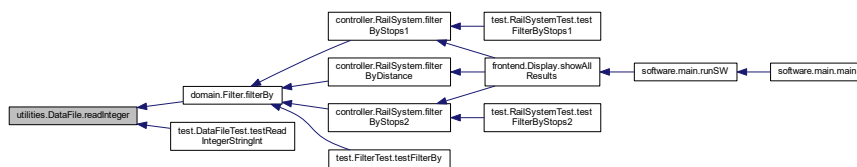
Returns

retorna o inteiro lido do arquivo

Here is the call graph for this function:



Here is the caller graph for this function:



6.3.1.8 readList() [1/2]

```
static String [] utilities.DataFile.readList (
    FileProperty property ) [static]
```

Le uma lista de string do arquivo de entrada

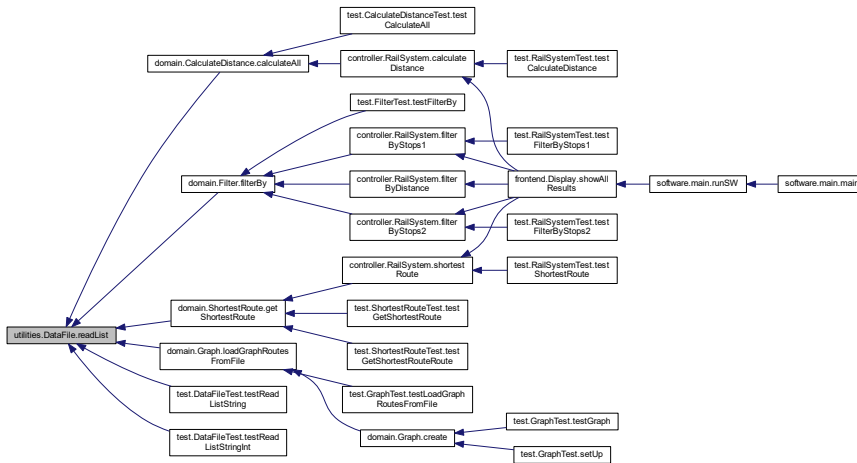
Parameters

<i>property</i>	identificador da propriedade a ser lida
-----------------	---

Returns

retorna o array de string lido do arquivo

Here is the caller graph for this function:



6.3.1.9 readList() [2/2]

```
static String [] utilities.DataFile.readList (
    FileProperty property,
    int index_filter ) [static]
```

Le um array de string do arquivo de entrada

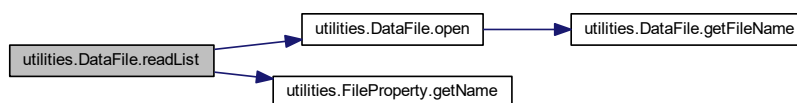
Parameters

<i>property</i>	propriedade onde o array de string sera buscado
<i>index_filter</i>	index da propriedade a ser buscada ex: propriedade[index].condition

Returns

retorna o array de string lido do arquivo

Here is the call graph for this function:



6.3.1.10 readLiteralOperand()

```
static String utilities.DataFile.readLiteralOperand (
    FileProperty property,
    int index_filter ) [static]
```

Le um operando na forma literal do arquivo de entrada

Parameters

<i>property</i>	propriedade a ser lida
-----------------	------------------------

Returns

retorna o operando, na forma literal, lido do arquivo

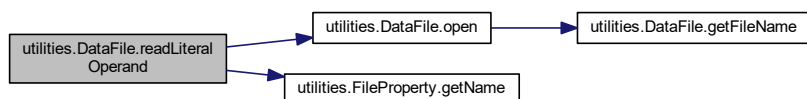
Parameters

<i>index_filter</i>	index identifica qual propriedade dentro do array que existe no arquivo
---------------------	---

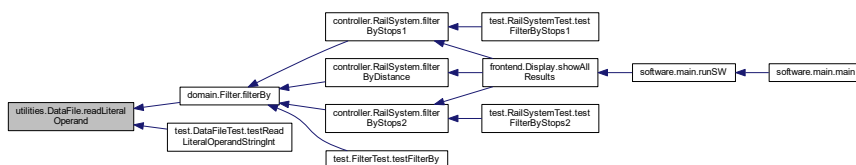
Returns

retorna o operando, na forma literal, lido do arquivo

Here is the call graph for this function:



Here is the caller graph for this function:



6.3.1.11 setFileName()

```
static void utilities.DataFile.setFileName (  
    String file ) [static]
```

Configura o nome do arquivo

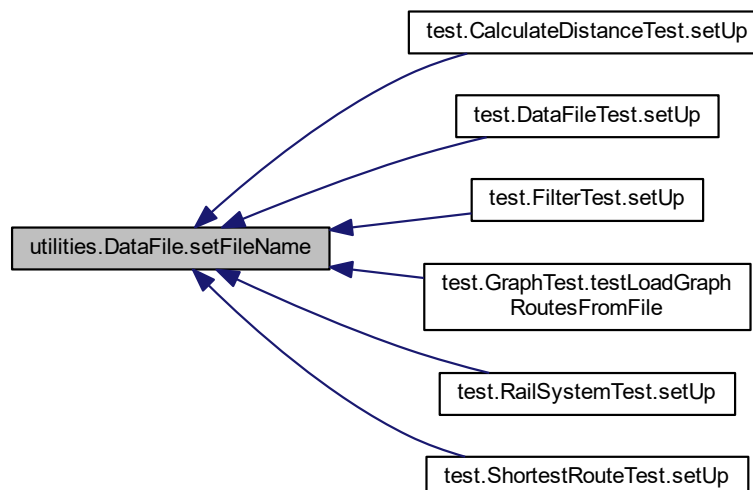
Parameters

<i>file</i>	path + nome do arquivo a ser utilizado para entrada dos valores
-------------	---

Here is the call graph for this function:



Here is the caller graph for this function:



6.3.2 Member Data Documentation

6.3.2.1 DISTANCE_CONDITION

```
final int utilities.DataFile.DISTANCE_CONDITION =3 [static]
```

6.3.2.2 file_name

```
String utilities.DataFile.file_name = PATH_FILE [static], [private]
```

6.3.2.3 FILE_TESTS_CASES

```
final String utilities.DataFile.FILE_TESTS_CASES = "FILE_TEST_CASE.txt" [static]
```

6.3.2.4 input

```
InputStream utilities.DataFile.input = null [static], [package]
```

6.3.2.5 PATH_FILE

```
final String utilities.DataFile.PATH_FILE = "input.txt" [static]
```

Nome do Arquivo de entrada

6.3.2.6 PATTERN_CONDITION

```
final String utilities.DataFile.PATTERN_CONDITION ="[^<>=!]" [static], [private]
```

6.3.2.7 PATTERN_DOUBLE

```
final String utilities.DataFile.PATTERN_DOUBLE ="[^0-9.,]" [static], [private]
```

6.3.2.8 PATTERN_INTEGER

```
final String utilities.DataFile.PATTERN_INTEGER ="[^0-9]" [static], [private]
```

6.3.2.9 PATTERN_LIST

```
final String utilities.DataFile.PATTERN_LIST = "[^a-zA-Z0-9.]" [static], [private]
```

6.3.2.10 PATTERN_OPERAND

```
final String utilities.DataFile.PATTERN_OPERAND = "[^a-zA-Z,]" [static], [private]
```

6.3.2.11 prop

```
Properties utilities.DataFile.prop = new Properties() [static], [package]
```

6.3.2.12 SEPARATOR

```
final String utilities.DataFile.SEPARATOR = "," [static], [private]
```

6.3.2.13 STOP1_CONDITION

```
final int utilities.DataFile.STOP1_CONDITION =1 [static]
```

6.3.2.14 STOP2_CONDITION

```
final int utilities.DataFile.STOP2_CONDITION =2 [static]
```

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

- [D:/workspace/TrainVagas/src/utilities/DataFile.java](#)

6.4 test.DataFileTest Class Reference

Public Member Functions

- void [setUp](#) () throws Exception
- void [testOpen](#) ()
- void [testReadListString](#) ()
- void [testReadListStringInt](#) ()
- void [testReadConditionStringInt](#) ()
- void [testReadDoubleStringInt](#) ()
- void [testReadIntegerStringInt](#) ()
- void [testReadLiteralOperandStringInt](#) ()

6.4.1 Member Function Documentation

6.4.1.1 setUp()

```
void test.DataFileTest.setUp ( ) throws Exception
```

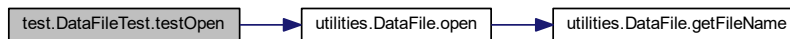
Here is the call graph for this function:



6.4.1.2 testOpen()

```
void test.DataFileTest.testOpen ( )
```

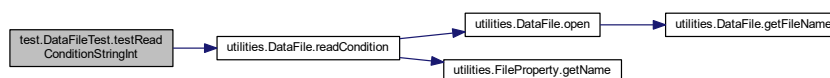
Here is the call graph for this function:



6.4.1.3 testReadConditionStringInt()

```
void test.DataFileTest.testReadConditionStringInt ( )
```

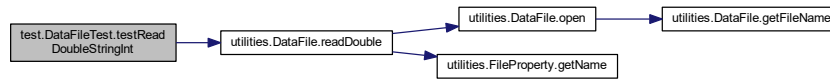
Here is the call graph for this function:



6.4.1.4 testReadDoubleStringInt()

```
void test.DataFileTest.testReadDoubleStringInt ( )
```

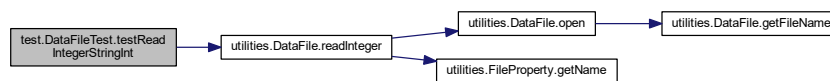
Here is the call graph for this function:



6.4.1.5 testReadIntegerStringInt()

```
void test.DataFileTest.testReadIntegerStringInt ( )
```

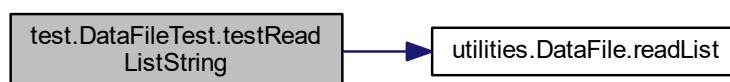
Here is the call graph for this function:



6.4.1.6 testReadListString()

```
void test.DataFileTest.testReadListString ( )
```

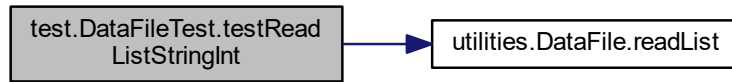
Here is the call graph for this function:



6.4.1.7 testReadListStringInt()

```
void test.DataFileTest.testReadListStringInt ( )
```

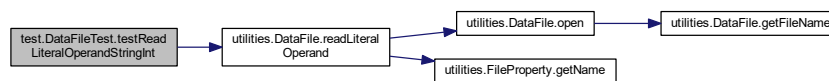
Here is the call graph for this function:



6.4.1.8 testReadLiteralOperandStringInt()

```
void test.DataFileTest.testReadLiteralOperandStringInt ( )
```

Here is the call graph for this function:



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

- <D:/workspace/TrainVagas/src/test/DataFileTest.java>

6.5 frontend.Display Class Reference

Static Public Member Functions

- static void [showAllResults](#) ()

Static Package Attributes

- static int [index_line](#) = 1
- static final String [FORMAT_DOUBLE](#) = "%.0f"
- static final String [FORMAT_DECIMAL](#) = "%d"
- static final String [FORMAT_ROUTE](#) = "[%s] "
- static final String [FORMAT_STRING](#) = "%-20s"
- static final String [FORMAT_COL1](#) = "%-7s"
- static final String [SEPARATOR](#) = " | "
- static final String [NEW_LINE](#) = "\n"

Static Private Member Functions

- static void [printHead](#) ()
- static void [printResultShortest](#) (Map< String, [Route](#)[]> shortest)
- static void [printResultCalculateDistance](#) ([Route](#)[] routes)
- static void [printResultFilter](#) ([Route](#)[] routes)
- static void [printLine](#) (String line)

6.5.1 Member Function Documentation

6.5.1.1 printHead()

```
static void frontend.Display.printHead ( ) [static], [private]
```

Imprime cabecalho Here is the caller graph for this function:



6.5.1.2 printLine()

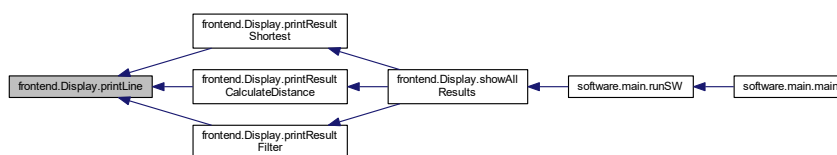
```
static void frontend.Display.printLine (
    String line ) [static], [private]
```

Imprime a linha na tela adicionando o numero da linha

Parameters

<i>line</i>	Formatada e tabulada
-------------	----------------------

Here is the caller graph for this function:



6.5.1.3 printResultCalculateDistance()

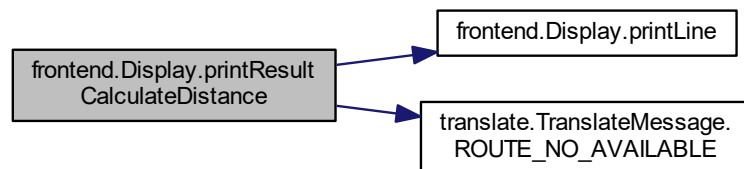
```
static void frontend.Display.printResultCalculateDistance (  
    Route [] routes ) [static], [private]
```

Imprime resultado dos calculos de distancias das rotas

Parameters

<i>routes</i>	Lista das rotas e seus respectivo valores de distancia
---------------	--

Here is the call graph for this function:



Here is the caller graph for this function:



6.5.1.4 printResultFilter()

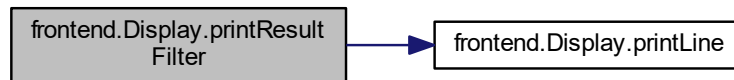
```
static void frontend.Display.printResultFilter (  
    Route [] routes ) [static], [private]
```

Imprime resultados dos filtros

Parameters

<i>routes</i>	
---------------	--

Here is the call graph for this function:



Here is the caller graph for this function:



6.5.1.5 printResultShortest()

```

static void frontend.Display.printResultShortest (
    Map< String, Route[]> shortest ) [static], [private]
  
```

Imprime resultado relacionado da consulta de menores rotas

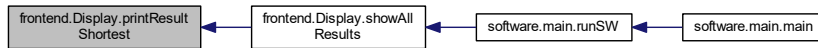
Parameters

<i>shortesd</i>	Lista das menores rotas
-----------------	-------------------------

Here is the call graph for this function:



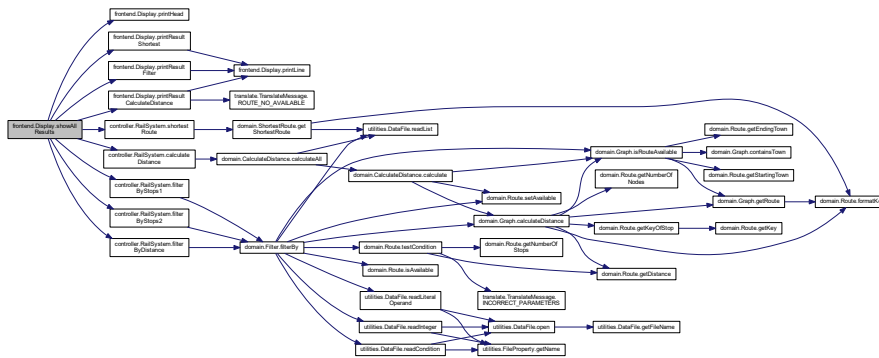
Here is the caller graph for this function:



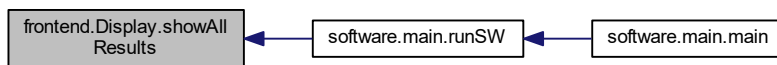
6.5.1.6 showAllResults()

```
static void frontend.Display.showAllResults ( ) [static]
```

Mostra todos os resultados. Here is the call graph for this function:



Here is the caller graph for this function:



6.5.2 Member Data Documentation

6.5.2.1 FORMAT_COL1

```
final String frontend.Display.FORMAT_COL1 = "%-7s" [static], [package]
```

6.5.2.2 FORMAT_DECIMAL

```
final String frontend.Display.FORMAT_DECIMAL = "%d" [static], [package]
```

6.5.2.3 FORMAT_DOUBLE

```
final String frontend.Display.FORMAT_DOUBLE = "%.0f" [static], [package]
```

6.5.2.4 FORMAT_ROUTE

```
final String frontend.Display.FORMAT_ROUTE = "[%s] " [static], [package]
```

6.5.2.5 FORMAT_STRING

```
final String frontend.Display.FORMAT_STRING = "%-20s" [static], [package]
```

6.5.2.6 index_line

```
int frontend.Display.index_line = 1 [static], [package]
```

6.5.2.7 NEW_LINE

```
final String frontend.Display.NEW_LINE = "\n" [static], [package]
```

6.5.2.8 SEPARATOR

```
final String frontend.Display.SEPARATOR = " | " [static], [package]
```

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

- [D:/workspace/TrainVagas/src/frontend/Display.java](#)

6.6 utilities.FileProperty Enum Reference

Public Member Functions

- `FileProperty` (String *name*)
- String `getName` ()

Public Attributes

- `GRAPH_ROUTES` =("graph.routes")
- `DISTANCE_ROUTES` =("distance.routes")
- `FILTER_CONDITION` =("filtertrips[%d].condition")
- `FILTER_ROUTES` =("filtertrips[%d].routes")
- `SHORTEST_ROUTES` =("shortestRouter.trip")

Private Attributes

- String *name*

6.6.1 Constructor & Destructor Documentation

6.6.1.1 FileProperty()

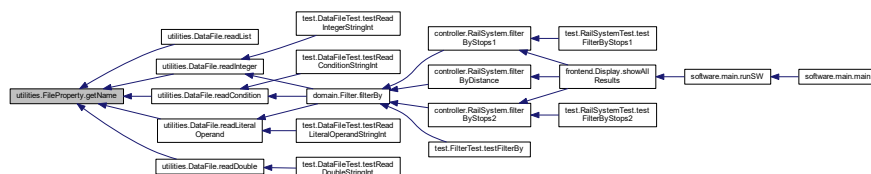
```
utilities.FileProperty.FileProperty (
    String name )
```

6.6.2 Member Function Documentation

6.6.2.1 getName()

```
String utilities.FileProperty.getName ( )
```

Here is the caller graph for this function:



6.6.3 Member Data Documentation

6.6.3.1 DISTANCE_ROUTES

```
utilities.FileProperty.DISTANCE_ROUTES = ("distance.routes")
```

Questoes: 1) The distance of the route A-B-C. 2) The distance of the route A-D. 3) The distance of the route A-D-C. 4) The distance of the route A-E-B-C-D. 5) The distance of the route A-E-D.

Mapeamento das rotas para calculo da distancia O hifem e opcional Separador virgula Possibilidade de adicionar novas rotas

6.6.3.2 FILTER_CONDITION

```
utilities.FileProperty.FILTER_CONDITION = ("filtertrips[%d].condition")
```

Questoes: 6) The number of trips starting at C and ending at C with a maximum of 3 stops. In the sample data below, there are two such trips: C-D-C (2 stops). and C-E-B-C (3 stops). 7) The number of trips starting at A and ending at C with exactly 4 stops. In the sample data below, there are three such trips: A to C (via B,C,D); A to C (via D,C,D); and A to C (via D,E,B). 10)The number of different routes from C to C with a distance of less than 30. In the sample data, the trips are: CDC, CEBC, CEBCDC, CDCEBC, CDEBC, CEBCEBC, CEBCEBCEBC.

Verifica se as rotas indicadas na propriedade filtertrips[X].routes satisfazem a condicao indicada em filtertrips[X].condition. Condicoes permitidas: <, >, <=, >=, ==, !=. Operandos permitidos STOPS, DISTANCE. O hifem e opcional. Rotas invalidas serao desconsideradas. Importante: Separador por virgula. Possibilidade de adicionar novos filtros, diferenciar atraves do index [X].

6.6.3.3 FILTER_ROUTES

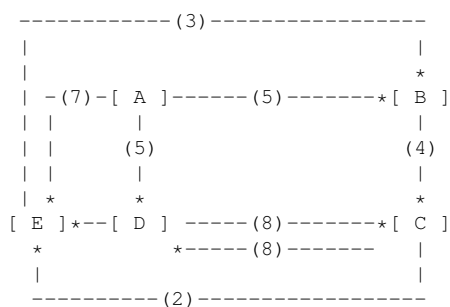
```
utilities.FileProperty.FILTER_ROUTES = ("filtertrips[%d].routes")
```

6.6.3.4 GRAPH_ROUTES

```
utilities.FileProperty.GRAPH_ROUTES = ("graph.routes")
```

Mapa de estacoes

Diagrama esta baseado na descricao do teste. '*' Indica rotas de chegada na estacao (D) Indica a distancia



Mapeamento de todas as rotas disponiveis O SW esta trabalhando dinamicamente, podendo ser adicionadas novas rotas Para numeros decimais usar '.' Separador por virgula

6.6.3.5 name

```
String utilities.FileProperty.name [private]
```

6.6.3.6 SHORTEST_ROUTES

```
utilities.FileProperty.SHORTEST_ROUTES = ("shortestRouter.trip")
```

Questoes: 8) The length of the shortest route (in terms of distance to travel) from A to C. 9) The length of the shortest route (in terms of distance to travel) from B to B.

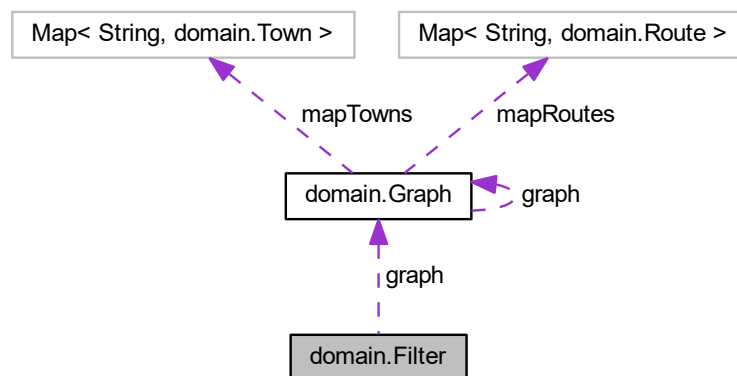
Encontra as opcoes de rotas menores(distancia) entre o ponto de partida e o de chegada Possibilidade de adicionar quantas viagens(rotas) necessarias. Rotas invalidas serao desconsideradas O hifem e opcional Separador virgula

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

- [D:/workspace/TrainVagas/src/utilities/FileProperty.java](#)

6.7 domain.Filter Class Reference

Collaboration diagram for domain.Filter:



Public Member Functions

- [Route \[\] filterBy](#) (int index_filter)

Package Attributes

- [Graph graph](#) = [Graph.create\(\)](#)

6.7.1 Member Function Documentation

6.7.1.1 filterBy()

```
Route [] domain.Filter.filterBy (
    int index_filter )
```

Verifica se as rotas indicadas no arquivo 'input.txt' na propriedade 'filtertrips[X].routes' satisfazem a condicao indicada em 'filtertrips[X].condition'. Condicoes permitidas: '<','>','<=','>=','==','!=' Operandos permitidos STOPS,DISTAN←
NCE Rotas invalidas serao desconsideradas

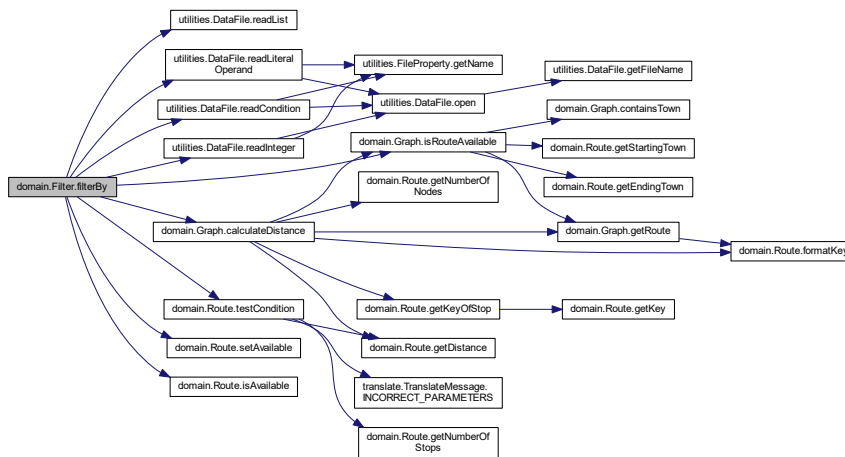
Parameters

<i>num_test</i>	indice do filtro
-----------------	------------------

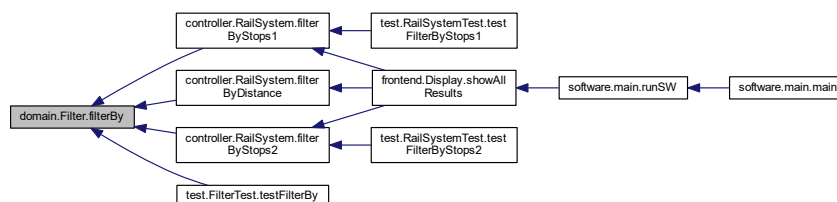
Returns

Rotas filtradas pela condicao

Here is the call graph for this function:



Here is the caller graph for this function:



6.7.2 Member Data Documentation

6.7.2.1 graph

`Graph` `domain.Filter.graph = Graph.create()` [package]

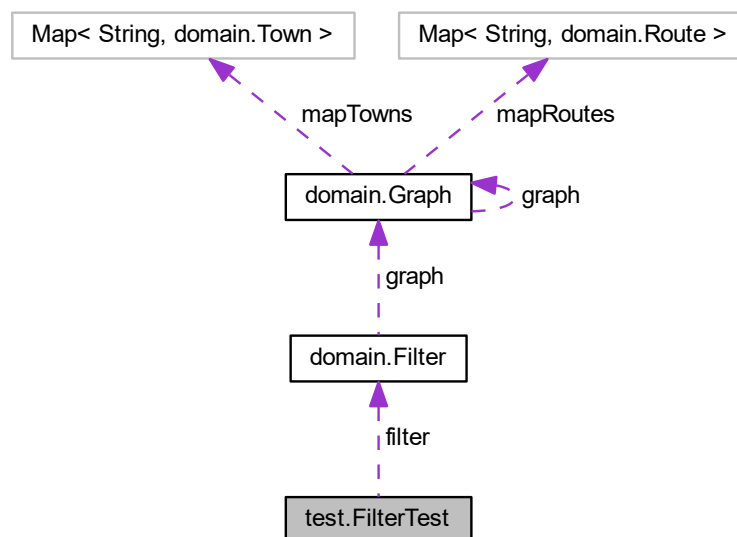
Rotas disponiveis no sistema

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

- [D:/workspace/TrainVagas/src/domain/Filter.java](#)

6.8 test.FilterTest Class Reference

Collaboration diagram for test.FilterTest:



Public Member Functions

- void `setUp()` throws Exception
- void `testFilterBy()`

Package Attributes

- `Filter filter`

6.8.1 Member Function Documentation

6.8.1.1 setUp()

```
void test.FilterTest.setUp ( ) throws Exception
```

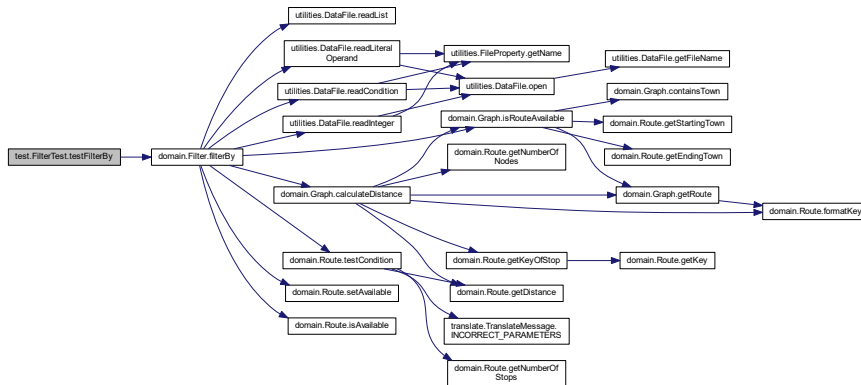
Here is the call graph for this function:



6.8.1.2 testFilterBy()

```
void test.FilterTest.testFilterBy ( )
```

Here is the call graph for this function:



6.8.2 Member Data Documentation

6.8.2.1 filter

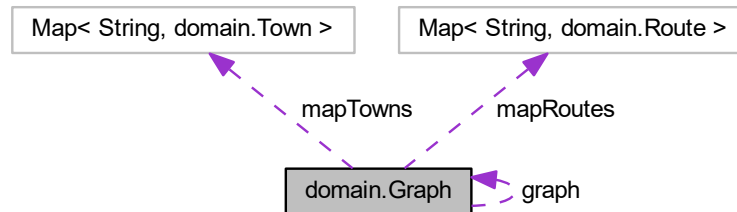
```
Filter test.FilterTest.filter [package]
```

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

- [D:/workspace/TrainVagas/src/test/FilterTest.java](#)

6.9 domain.Graph Class Reference

Collaboration diagram for domain.Graph:



Public Member Functions

- void [clear](#) ()
- String [] [getAllTowns](#) ()
- String [] [getAllRoutes](#) ()
- String [] [getAllRoutePossible](#) (Route trip)
- String [] [getConnections](#) (String key_town)
- void [addRoute](#) (String key_route)
- void [addRoute](#) (String[] routes)
- void [addTown](#) (String key_town)
- Route [getRoute](#) (String key_route)
- void [loadGraphRoutesFromFile](#) ()
- boolean [containsTown](#) (String key_town)
- boolean [isRouteAvailable](#) (String route)
- double [calculateDistance](#) (String trip)

Static Public Member Functions

- static synchronized Graph [create](#) ()

Static Package Attributes

- static Graph [graph](#) = null

Private Member Functions

- Graph ()

Private Attributes

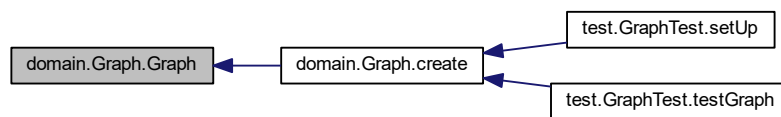
- Map< String, Route > [mapRoutes](#) = null
- Map< String, Town > [mapTowns](#) = null

6.9.1 Constructor & Destructor Documentation

6.9.1.1 Graph()

```
domain.Graph.Graph ( ) [private]
```

Construtor privado. Para instanciar o objeto sera utilizado o metodo [create\(\)](#) Here is the caller graph for this function:



6.9.2 Member Function Documentation

6.9.2.1 addRoute() [1/2]

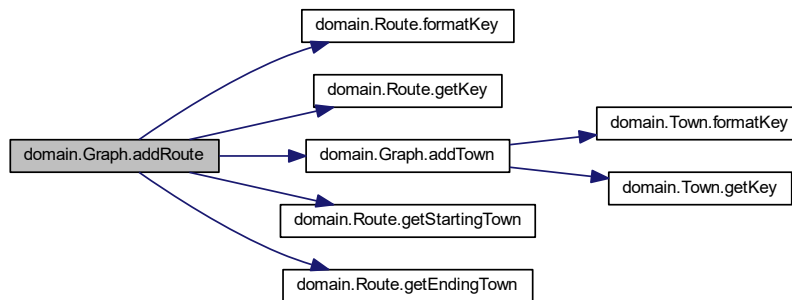
```
void domain.Graph.addRoute (
    String key_route )
```

Cria e armazena no container uma rota com o key indicado por key_route

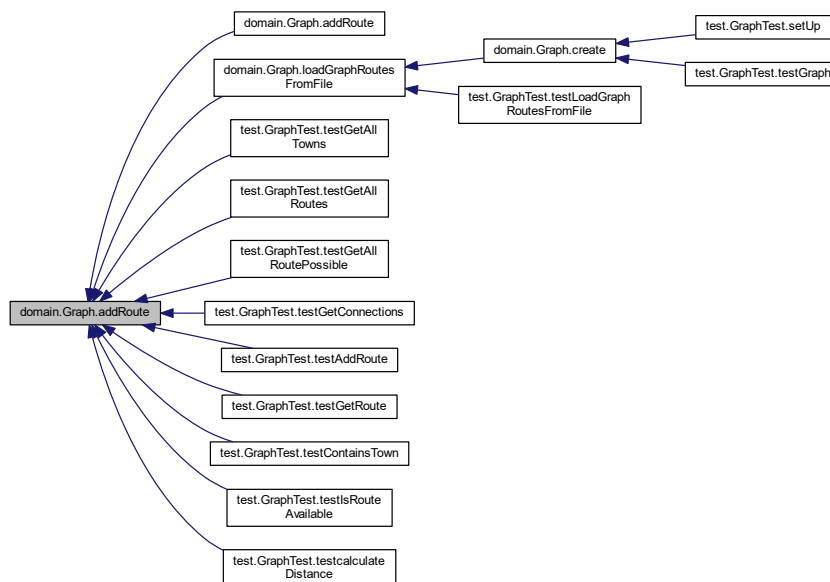
Parameters

<i>key</i>	Chave utilizada no map
------------	------------------------

Here is the call graph for this function:



Here is the caller graph for this function:



6.9.2.2 addRoute() [2/2]

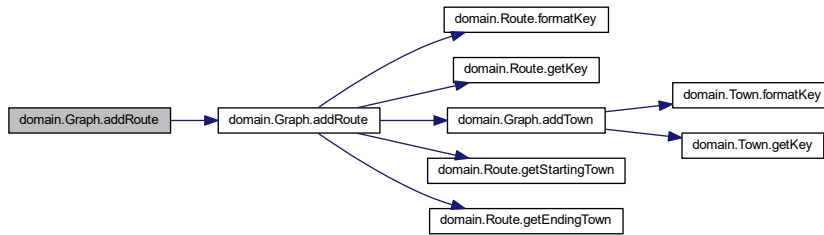
```
void domain.Graph.addRoute (
    String [] routes )
```

Cria e armazena no container uma lista de rotas

Parameters

<code>routes</code>	Lista dos keys para criaçao das rotas
---------------------	---------------------------------------

Here is the call graph for this function:



6.9.2.3 addTown()

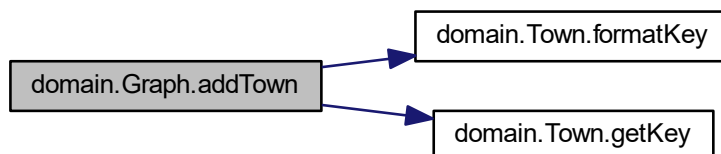
```
void domain.Graph.addTown (
    String key_town )
```

Cria e armazena no container uma cidade identificada por `key_town`

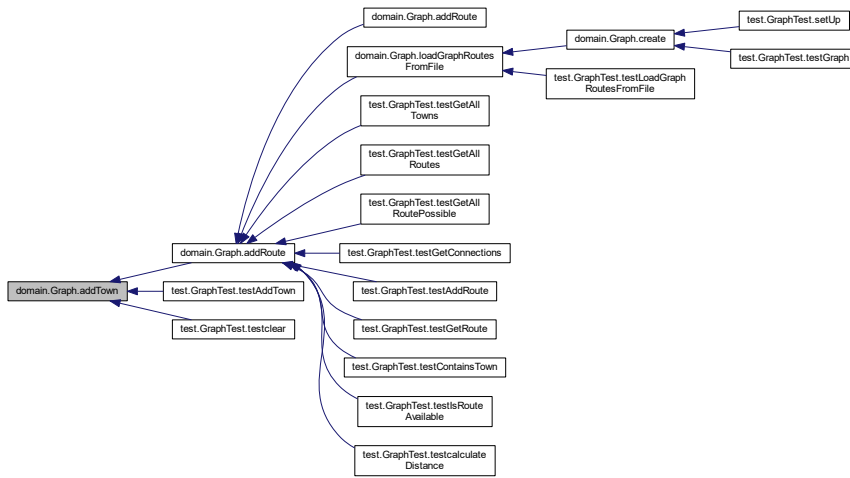
Parameters

<i>key</i>	Chave utilizada no map
------------	------------------------

Here is the call graph for this function:



Here is the caller graph for this function:



6.9.2.4 calculateDistance()

```
double domain.Graph.calculateDistance (
    String trip )
```

Calcula a distancia total de uma rota(viagem)

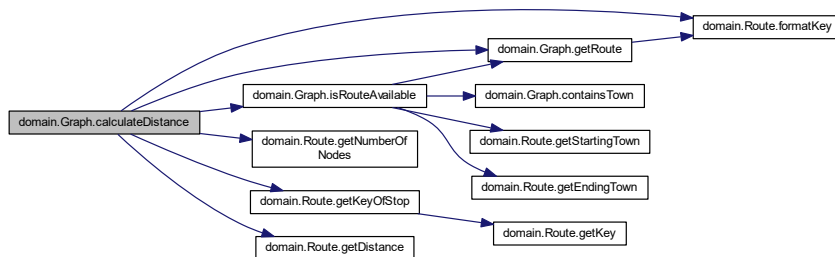
Parameters

<i>route</i>	Key da rota(viagem) que se deseja calcular a distancia
--------------	--

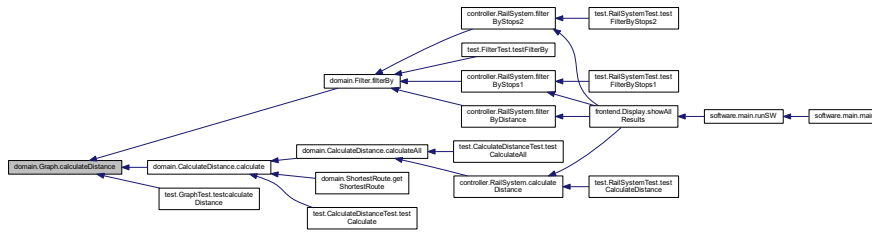
Returns

Valor da distancia. Retornara 0 se a rota nao for possivel

Here is the call graph for this function:



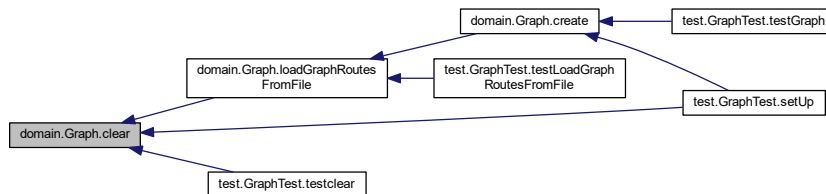
Here is the caller graph for this function:



6.9.2.5 clear()

```
void domain.Graph.clear ( )
```

Exclui todos as rotas e casas dos containers Here is the caller graph for this function:



6.9.2.6 containsTown()

```
boolean domain.Graph.containsTown (
    String key_town )
```

Verifica se a cidade existe e se a referencia do objeto no container e valida

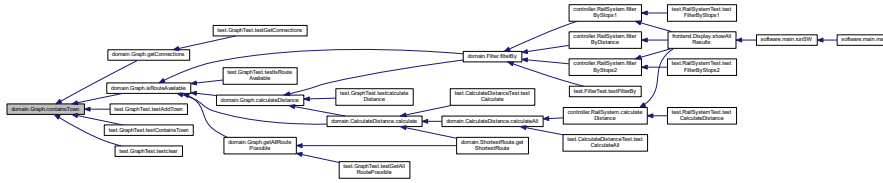
Parameters

<i>key_town</i>	ke da chave para ser verificada
-----------------	---------------------------------

Returns

Retorna true se a cidade existe e com referencia valida

Here is the caller graph for this function:



6.9.2.7 create()

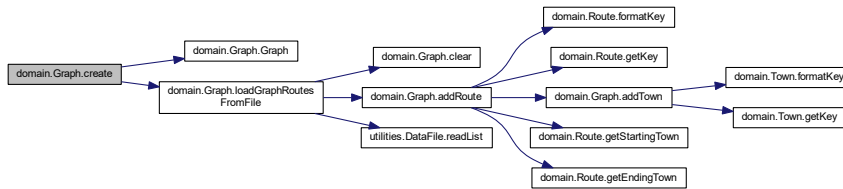
```
static synchronized Graph domain.Graph.create ( ) [static]
```

Metodo construtor para manter um Singleton de Graph.

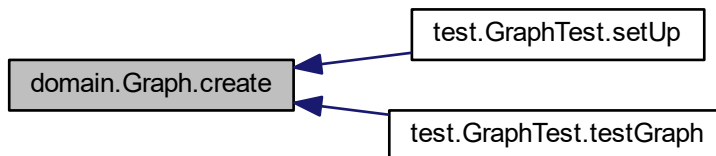
Padrao Singleton

Returns

Here is the call graph for this function:



Here is the caller graph for this function:



6.9.2.8 getAllRoutePossible()

```
String [] domain.Graph.getAllRoutePossible (
    Route trip )
```

Retorna todas as possíveis rotas para uma viagem

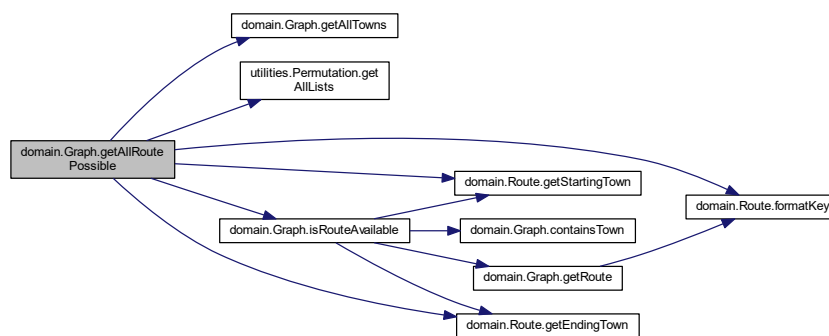
Parameters

<i>trip</i>	E a rota da viagem contendo a cidade de destino e origem da viagem
-------------	--

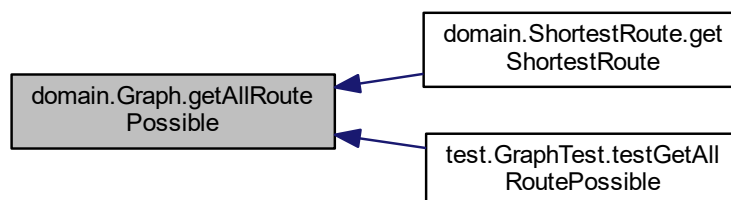
Returns

Retorna todas as possíveis rotas para a viagem

Here is the call graph for this function:



Here is the caller graph for this function:



6.9.2.9 getAllRoutes()

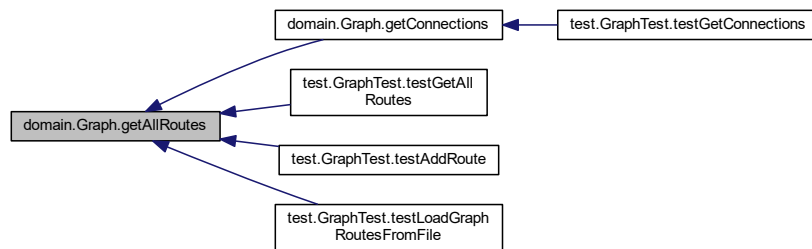
```
String [] domain.Graph.getAllRoutes ( )
```

Retorna a lista dos keys de todas as rotas disponiveis

Returns

Lsita de key das rotas

Here is the caller graph for this function:



6.9.2.10 getAllTowns()

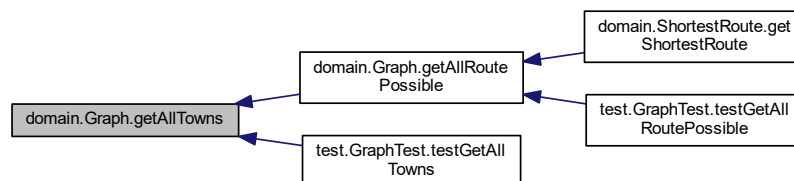
```
String [] domain.Graph.getAllTowns ( )
```

Retorna a lista dos keys de todas as cidades disponiveis

Returns

Lista de keys das cidades

Here is the caller graph for this function:



6.9.2.11 getConnections()

```
String [] domain.Graph.getConnections (
    String key_town )
```

Retorna as conexoes disponiveis para uma determinada cidade

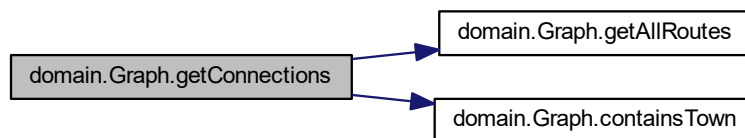
Parameters

<i>key_town</i>	Key da cidade que se deseja listar as conexoes existentes
-----------------	---

Returns

Lista das conexoes existentes para a cidade. Caso rotas ou a cidade nao exista retornara null

Here is the call graph for this function:



Here is the caller graph for this function:

**6.9.2.12 getRoute()**

```
Route domain.Graph.getRoute (
    String key_route )
```

Retorna rota indicado por `key_route`

Parameters

<i>key_route</i>	Key da rota que se deseja retornar
------------------	------------------------------------

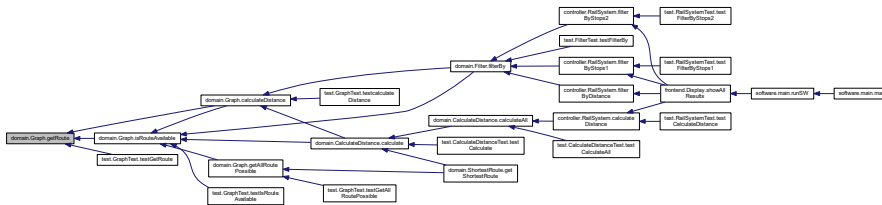
Returns

Retorna a rota correspondente ao key_route. Se a rota nao existir retorna null.

Here is the call graph for this function:



Here is the caller graph for this function:



6.9.2.13 isRouteAvailable()

```
boolean domain.Graph.isRouteAvailable (
    String route )
```

Verifica se a rota e valida e disponivel

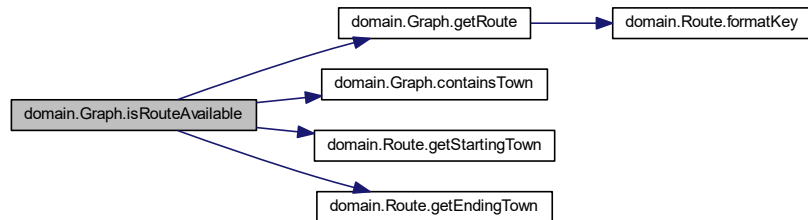
Parameters

route	Key da rota a ser verificado
-------	------------------------------

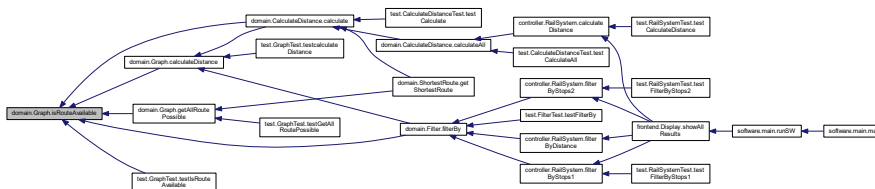
Returns

Retorna true se a rota for possivel

Here is the call graph for this function:



Here is the caller graph for this function:



6.9.2.14 loadGraphRoutesFromFile()

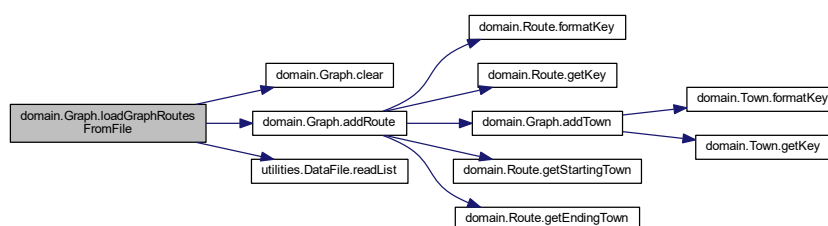
```
void domain.Graph.loadGraphRoutesFromFile ( )
```

Carrega as rotas do mapa contidas no arquivo de entrada "input.txt na propriedade "graph.routes"

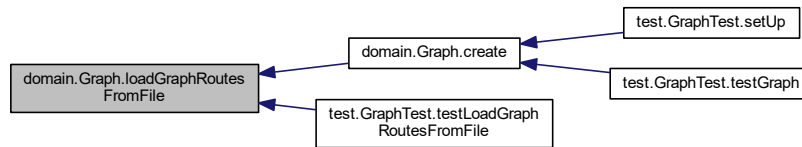
See also

input.txt

Here is the call graph for this function:



Here is the caller graph for this function:



6.9.3 Member Data Documentation

6.9.3.1 graph

```
Graph domain.Graph.graph = null [static], [package]
```

Referencia do Singleton

6.9.3.2 mapRoutes

```
Map<String , Route> domain.Graph.mapRoutes = null [private]
```

Container de Rotas disponiveis Formato do key String ex: 'AB'

6.9.3.3 mapTowns

```
Map<String , Town> domain.Graph.mapTowns = null [private]
```

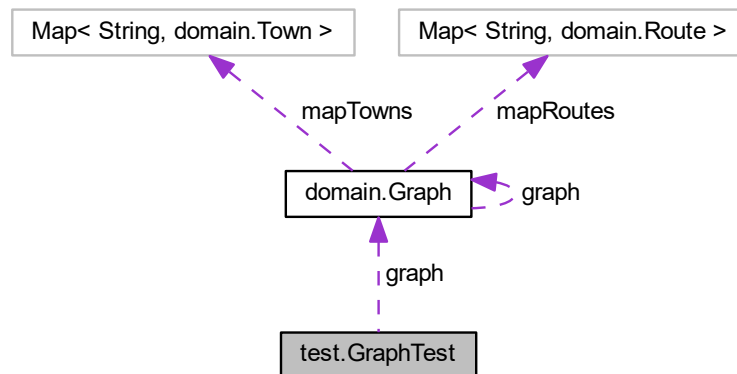
Container de cidades Formato do key String ex: 'A'

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

- [D:/workspace/TrainVagas/src/domain/Graph.java](#)

6.10 test.GraphTest Class Reference

Collaboration diagram for test.GraphTest:



Public Member Functions

- void [setUp](#) () throws Exception
- void [testGraph](#) ()
- void [testGetAllTowns](#) ()
- void [testGetAllRoutes](#) ()
- void [testGetAllRoutePossible](#) ()
- void [testGetConnections](#) ()
- void [testAddRoute](#) ()
- void [testAddTown](#) ()
- void [testGetRoute](#) ()
- void [testLoadGraphRoutesFromFile](#) ()
- void [testContainsTown](#) ()
- void [testIsRouteAvailable](#) ()
- void [testcalculateDistance](#) ()
- void [testclear](#) ()

Package Attributes

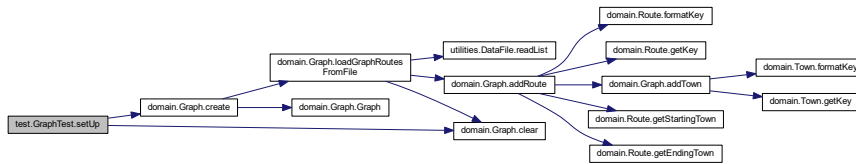
- [Graph](#) graph

6.10.1 Member Function Documentation

6.10.1.1 setUp()

```
void test.GraphTest.setUp ( ) throws Exception
```

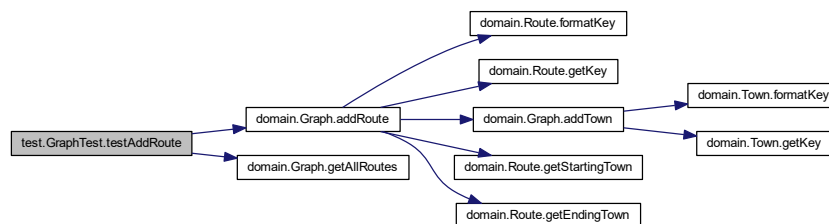
Here is the call graph for this function:



6.10.1.2 testAddRoute()

```
void test.GraphTest.testAddRoute ( )
```

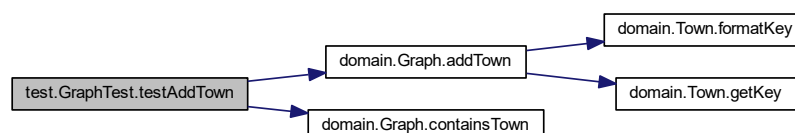
Here is the call graph for this function:



6.10.1.3 testAddTown()

```
void test.GraphTest.testAddTown ( )
```

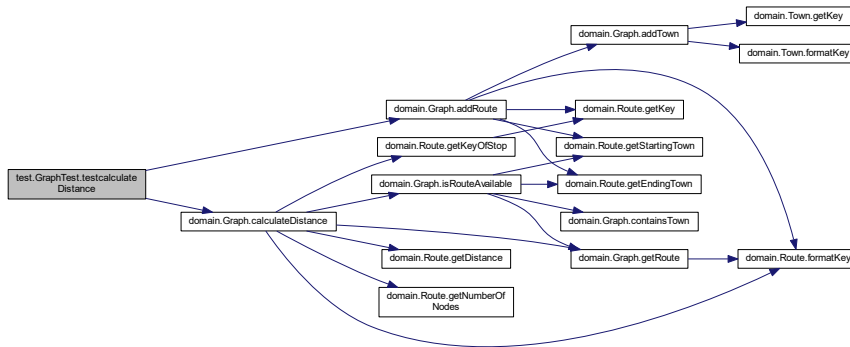
Here is the call graph for this function:



6.10.1.4 testcalculateDistance()

```
void test.GraphTest.testcalculateDistance ( )
```

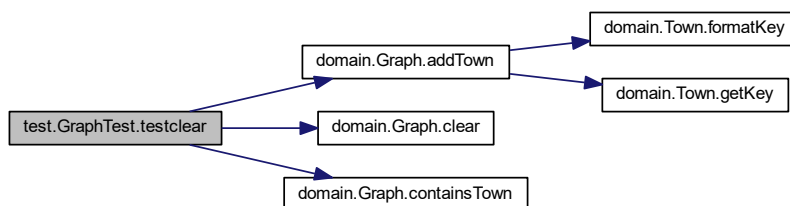
Here is the call graph for this function:



6.10.1.5 testclear()

```
void test.GraphTest.testclear ( )
```

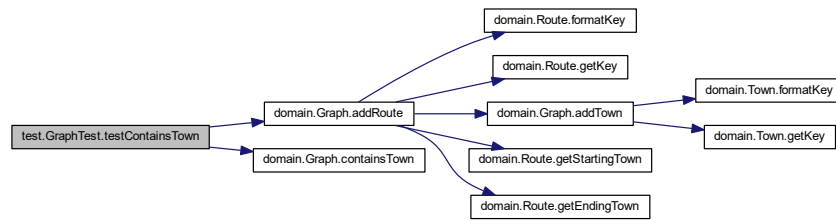
Here is the call graph for this function:



6.10.1.6 testContainsTown()

```
void test.GraphTest.testContainsTown ( )
```

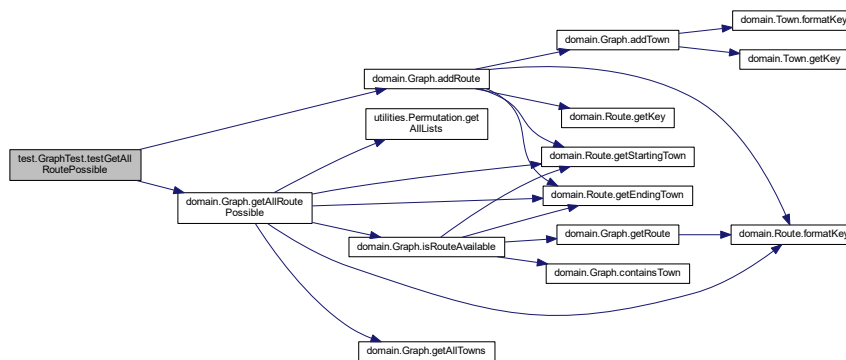
Here is the call graph for this function:



6.10.1.7 testGetAllRoutePossible()

```
void test.GraphTest.testGetAllRoutePossible ( )
```

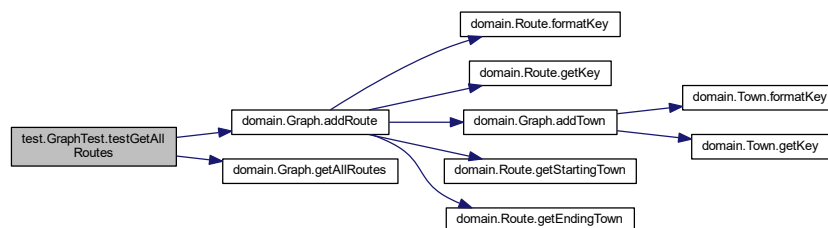
Here is the call graph for this function:



6.10.1.8 testGetAllRoutes()

```
void test.GraphTest.testGetAllRoutes ( )
```

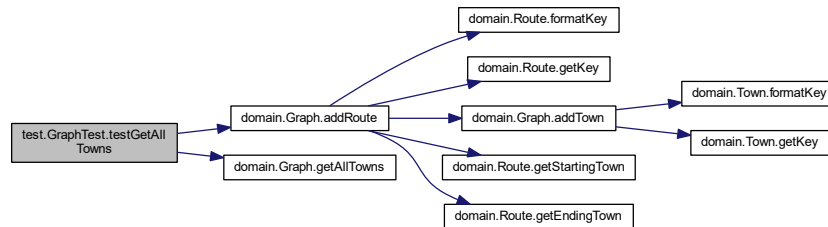
Here is the call graph for this function:



6.10.1.9 testGetAllTowns()

```
void test.GraphTest.testGetAllTowns ( )
```

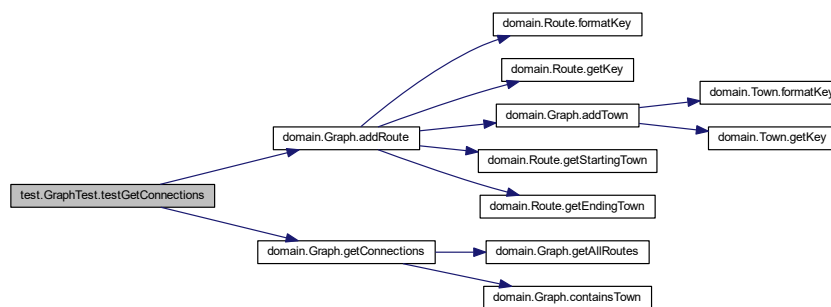
Here is the call graph for this function:



6.10.1.10 testGetConnections()

```
void test.GraphTest.testGetConnections ( )
```

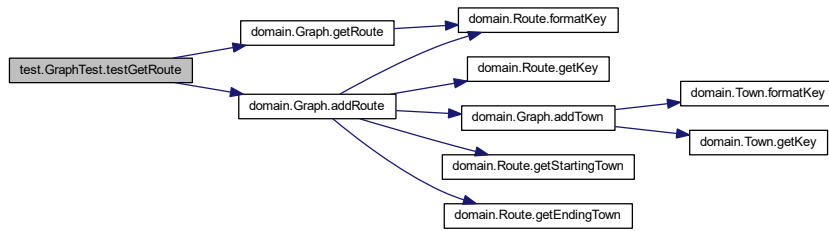
Here is the call graph for this function:



6.10.1.11 testGetRoute()

```
void test.GraphTest.testGetRoute ( )
```

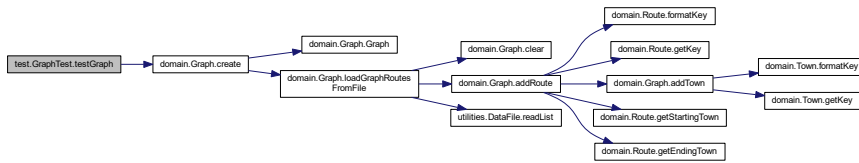

Here is the call graph for this function:



6.10.1.12 testGraph()

```
void test.GraphTest.testGraph ( )
```

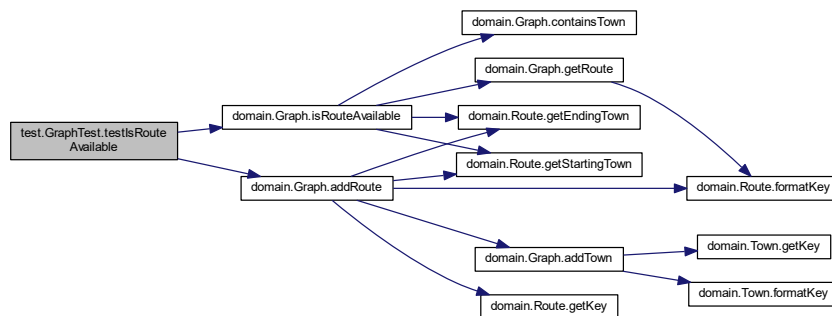
Here is the call graph for this function:



6.10.1.13 testIsRouteAvailable()

```
void test.GraphTest.testIsRouteAvailable ( )
```

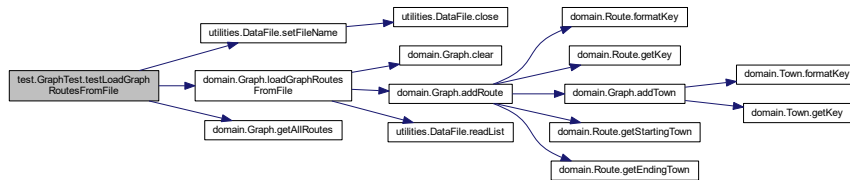
Here is the call graph for this function:



6.10.1.14 testLoadGraphRoutesFromFile()

```
void test.GraphTest.testLoadGraphRoutesFromFile ( )
```

Here is the call graph for this function:



6.10.2 Member Data Documentation

6.10.2.1 graph

```
Graph test.GraphTest.graph [package]
```

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

- <D:/workspace/TrainVagas/src/test/GraphTest.java>

6.11 software.main Class Reference

Static Public Member Functions

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

Static Package Functions

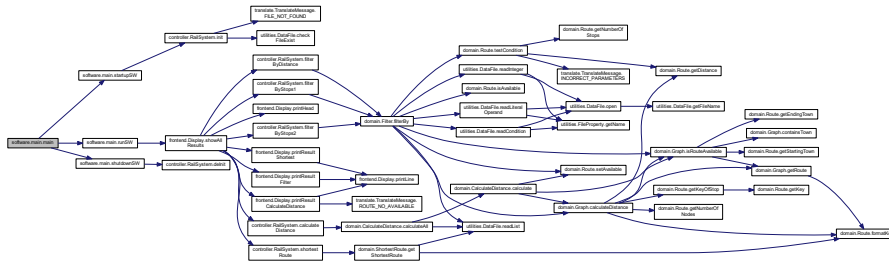
- static boolean [startupSW](#) ()
- static void [runSW](#) ()
- static void [shutdownSW](#) ()

6.11.1 Constructor & Destructor Documentation

6.11.1.1 main()

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

Here is the call graph for this function:

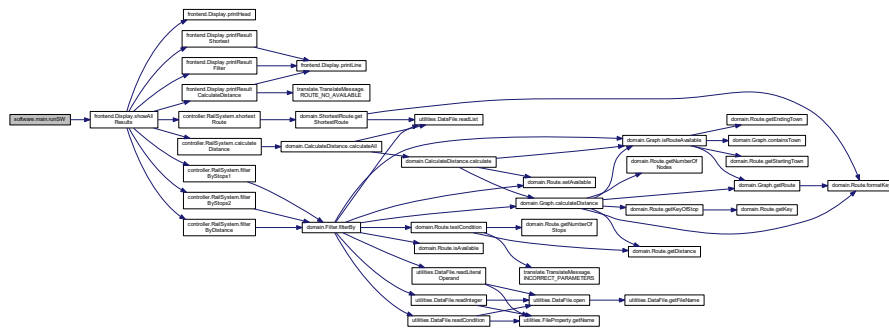


6.11.2 Member Function Documentation

6.11.2.1 runSW()

```
static void software.main.runSW ( ) [static], [package]
```

Execucao do SW Here is the call graph for this function:



Here is the caller graph for this function:



6.11.2.2 shutdownSW()

```
static void software.main.shutdownSW ( ) [static], [package]
```

Executa o shutdown do SW Here is the call graph for this function:



Here is the caller graph for this function:



6.11.2.3 startupSW()

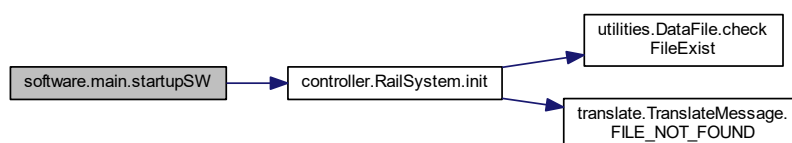
```
static boolean software.main.startupSW ( ) [static], [package]
```

Inicializacao do SW

Returns

Retorna true se a inicializacao foi bem sucedida

Here is the call graph for this function:



Here is the caller graph for this function:



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

- [D:/workspace/TrainVagas/src/software/main.java](#)

6.12 utilities.Permutation Class Reference

Static Public Member Functions

- static `String []` [getAllLists](#) (`String []` elements, `int` lengthOfList)

6.12.1 Member Function Documentation

6.12.1.1 getAllLists()

```
static String [] utilities.Permutation.getAllLists (  
    String [] elements,  
    int lengthOfList ) [static]
```

Cria todas as possiveis combinacoes entre os elementos de uma lista

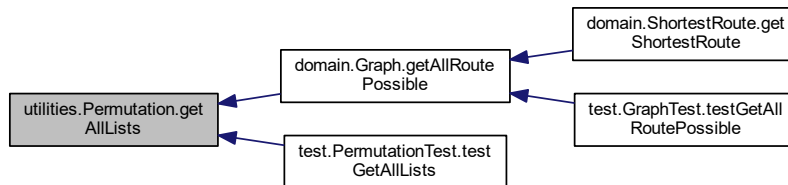
Parameters

<i>elements</i>	Lista de elementos
<i>lengthOfList</i>	quantidade de elementos

Returns

Lista com todas as combinacoes

Here is the caller graph for this function:



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

- <D:/workspace/TrainVagas/src/utilities/Permutation.java>

6.13 test.PermutationTest Class Reference

Public Member Functions

- void [setUp](#) () throws Exception
- void [testGetAllLists](#) ()

6.13.1 Member Function Documentation

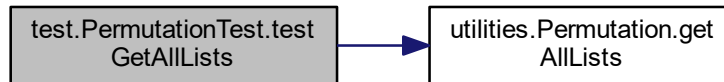
6.13.1.1 setUp()

```
void test.PermutationTest.setUp ( ) throws Exception
```

6.13.1.2 testGetAllLists()

```
void test.PermutationTest.testGetAllLists ( )
```

Here is the call graph for this function:

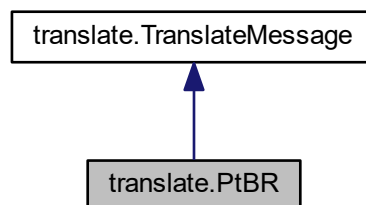


The documentation for this class was generated from the following file:

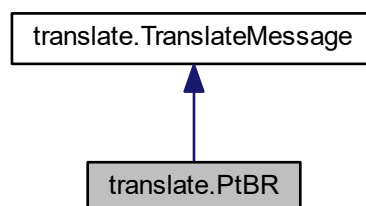
- [D:/workspace/TrainVagas/src/test/PermutationTest.java](#)

6.14 translate.PtBR Class Reference

Inheritance diagram for `translate.PtBR`:



Collaboration diagram for `translate.PtBR`:



Public Member Functions

- String [TOW_NO_EXIST](#) ()
- String [ROUTE_NO_AVAILABLE](#) ()
- String [INCORRECT_PARAMETERS](#) ()
- String [FILE_NOT_FOUND](#) ()
- String [PROPERTY_NOT_FOUND](#) ()

Static Package Attributes

- static final String [TOW_NO_EXIST](#) = "Cidade nao existente"
- static final String [ROUTE_NO_AVAILABLE](#) = "Rota nao disponivel"
- static final String [INCORRECT_PARAMETERS](#) = "Parametros do filtro errado.Verificar o arquivo de entrada"
- static final String [FILE_NOT_FOUND](#) = "Arquivo de entrada(input.txt) nao encontrado"
- static final String [PROPERTY_NOT_FOUND](#) = "Propriedade nao encontrada no arquivo de entrada"

6.14.1 Member Function Documentation

6.14.1.1 [FILE_NOT_FOUND](#)()

```
String translate.PtBR.FILE_NOT_FOUND ( )
```

Implements [translate.TranslateMessage](#).

6.14.1.2 [INCORRECT_PARAMETERS](#)()

```
String translate.PtBR.INCORRECT_PARAMETERS ( )
```

Implements [translate.TranslateMessage](#).

6.14.1.3 [PROPERTY_NOT_FOUND](#)()

```
String translate.PtBR.PROPERTY_NOT_FOUND ( )
```

Implements [translate.TranslateMessage](#).

6.14.1.4 ROUTE_NO_AVAILABLE()

```
String translate.PtBR.ROUTE_NO_AVAILABLE ( )
```

Implements [translate.TranslateMessage](#).

6.14.1.5 TOW_NO_EXIST()

```
String translate.PtBR.TOW_NO_EXIST ( )
```

Implements [translate.TranslateMessage](#).

6.14.2 Member Data Documentation

6.14.2.1 FILE_NOT_FOUND

```
final String translate.PtBR.FILE_NOT_FOUND = "Arquivo de entrada(input.txt) nao encontrado"  
[static], [package]
```

6.14.2.2 INCORRECT_PARAMETERS

```
final String translate.PtBR.INCORRECT_PARAMETERS = "Parametros do filtro errado.Verificar o  
arquivo de entrada" [static], [package]
```

6.14.2.3 PROPERTY_NOT_FOUND

```
final String translate.PtBR.PROPERTY_NOT_FOUND = "Propriedade nao encontrada no arquivo de  
entrada" [static], [package]
```

6.14.2.4 ROUTE_NO_AVAILABLE

```
final String translate.PtBR.ROUTE_NO_AVAILABLE = "Rota nao disponivel" [static], [package]
```

6.14.2.5 TOW_NO_EXIST

```
final String translate.PtBR.TOW_NO_EXIST = "Cidade nao existente" [static], [package]
```

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

- [D:/workspace/TrainVagas/src/translate/PtBR.java](#)

6.15 controller.RailSystem Class Reference

Static Public Member Functions

- static [Route \[\] filterByStops1 \(\)](#)
- static [Route \[\] filterByStops2 \(\)](#)
- static [Route \[\] filterByDistance \(\)](#)
- static [Route \[\] calculateDistance \(\)](#)
- static [Map< String,Route\[\]> shortestRoute \(\)](#)
- static boolean [init \(\)](#)
- static void [delInit \(\)](#)

6.15.1 Member Function Documentation

6.15.1.1 calculateDistance()

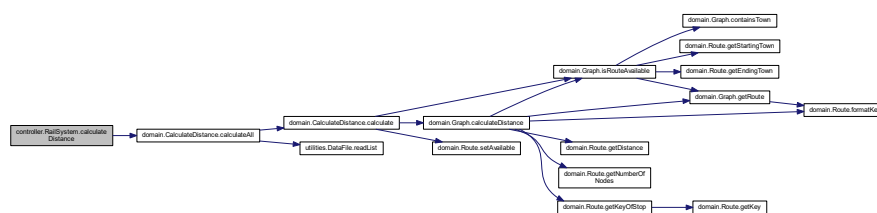
```
static Route [] controller.RailSystem.calculateDistance ( ) [static]
```

Calcula as distancias da rotas mapeadas no arquivo na propriedade 'distance.routes'

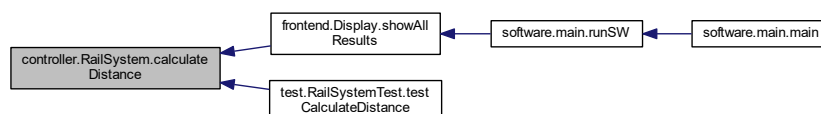
Returns

Retorna lista de rotas com as distancias calculadas

Here is the call graph for this function:



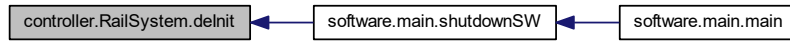
Here is the caller graph for this function:



6.15.1.2 delnit()

```
static void controller.RailSystem.deInit ( ) [static]
```

Finaliza o sistema de rotas Here is the caller graph for this function:



6.15.1.3 filterByDistance()

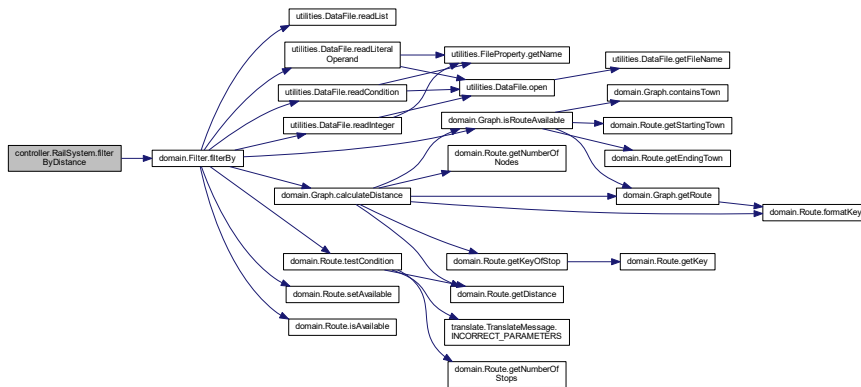
```
static Route [] controller.RailSystem.filterByDistance ( ) [static]
```

Esse filtro esta mapeado no arquivo de entrada com o index = 3(DataFile.DISTANCE_CONDITION) Condicao do filtro DISTANCE < 30

Returns

Retorna uma lista de rotas filtradas pela condicao

Here is the call graph for this function:



Here is the caller graph for this function:



6.15.1.4 filterByStops1()

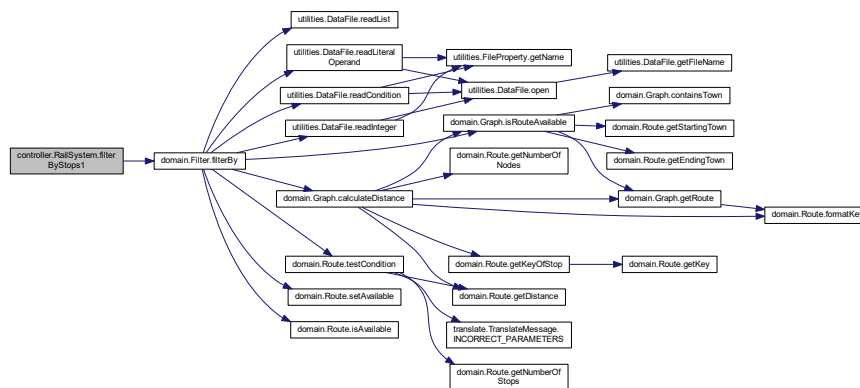
```
static Route [] controller.RailSystem.filterByStops1 ( ) [static]
```

Esse filtro esta mapeado no arquivo de entrada com o index = 1([DataFile.STOP1_CONDITION](#)) Condicao do filtro STOPS <= 3

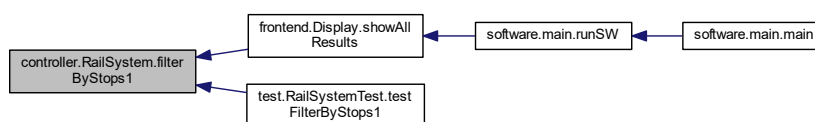
Returns

Retorna rotas filtradas pela condicao

Here is the call graph for this function:



Here is the caller graph for this function:



6.15.1.5 filterByStops2()

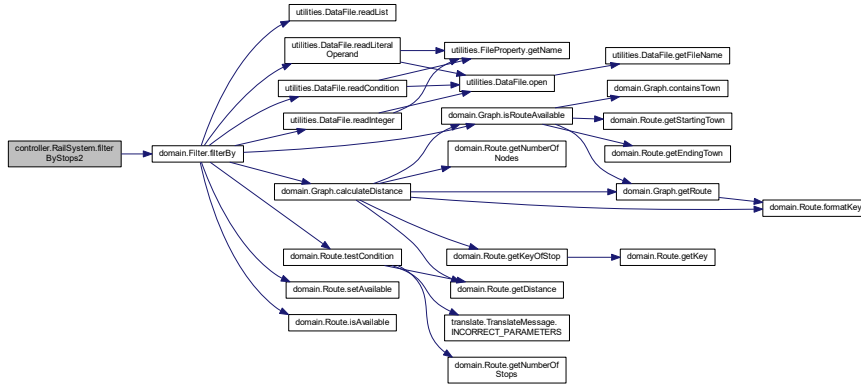
```
static Route [] controller.RailSystem.filterByStops2 ( ) [static]
```

Esse filtro esta mapeado no arquivo de entrada com o index = 2([DataFile.STOP2_CONDITION](#)) Condicao do filtro STOPS == 4

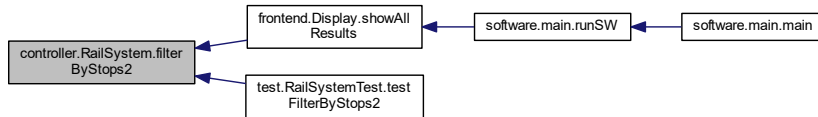
Returns

Retorna uma lista de rotas filtradas pela condicao

Here is the call graph for this function:



Here is the caller graph for this function:



6.15.1.6 init()

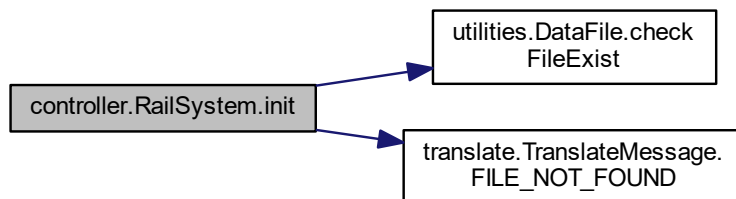
```
static boolean controller.RailSystem.init ( ) [static]
```

Inicializa o sistema de rotas

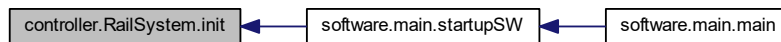
Returns

Se a inicializacao ocorreu com sucesso retorna true

Here is the call graph for this function:



Here is the caller graph for this function:



6.15.1.7 shortestRoute()

```
static Map<String ,Route[]> controller.RailSystem.shortestRoute ( ) [static]
```

Mapeia todas as rotas mais curtas de todas as viagens

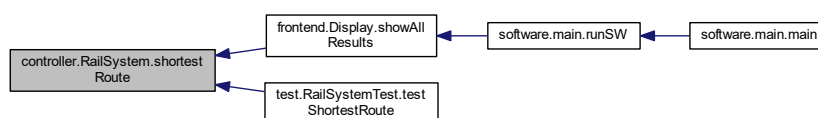
Returns

Retorna uma lista de rotas mais curtas para cada viagem/rota

Here is the call graph for this function:



Here is the caller graph for this function:



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

- [D:/workspace/TrainVagas/src/controller/RailSystem.java](#)

6.16 test.RailSystemTest Class Reference

Public Member Functions

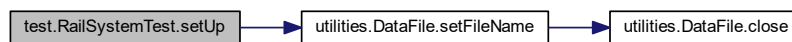
- void [setUp](#) () throws Exception
- void [testFilterByStops1](#) ()
- void [testFilterByStops2](#) ()
- void [testCalculateDistance](#) ()
- void [testShortestRoute](#) ()

6.16.1 Member Function Documentation

6.16.1.1 setUp()

void test.RailSystemTest.setUp () throws Exception

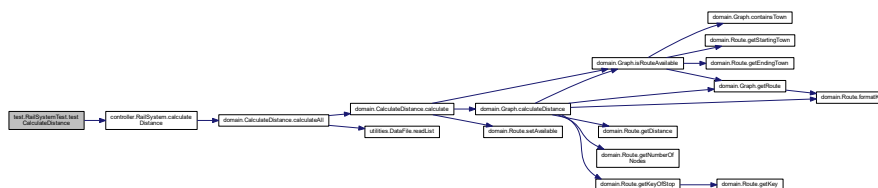
Here is the call graph for this function:



6.16.1.2 testCalculateDistance()

void test.RailSystemTest.testCalculateDistance ()

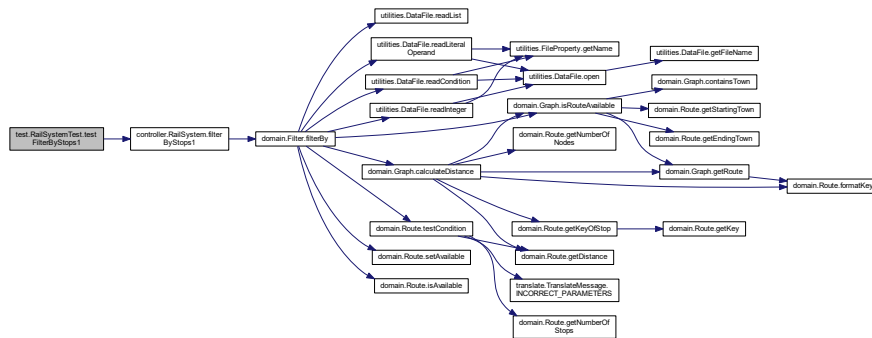
Here is the call graph for this function:



6.16.1.3 testFilterByStops1()

```
void test.RailSystemTest.testFilterByStops1 ( )
```

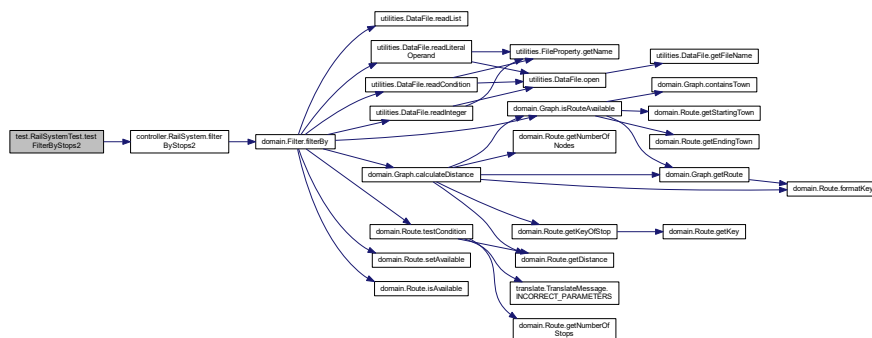
Here is the call graph for this function:



6.16.1.4 testFilterByStops2()

```
void test.RailSystemTest.testFilterByStops2 ( )
```

Here is the call graph for this function:



6.16.1.5 testShortestRoute()

```
void test.RailSystemTest.testShortestRoute ( )
```


Here is the call graph for this function:



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

- [D:/workspace/TrainVagas/src/test/RailSystemTest.java](#)

6.17 domain.Route Class Reference

Public Member Functions

- [Route](#) ()
- [Route](#) (String id, double total_distance)
- [Route](#) (String route)
- String [getStartingTown](#) ()
- double [getDistance](#) ()
- int [getNumberOfNodes](#) ()
- int [getNumberOfStops](#) ()
- String [getKeyOfStop](#) (int n_stop)
- void [setRoute](#) (String route)
- void [setKey](#) (String r)
- void [setTotalDistance](#) (String val)
- void [setTotalDistance](#) (double val)
- String [getKey](#) ()
- String [getEndingTown](#) ()
- boolean [isAvailable](#) ()
- void [setAvailable](#) (boolean exist)
- boolean [testCondition](#) (String condition, String operand, double valor2)

Static Public Member Functions

- static String [formatKey](#) (String key_route)

Private Attributes

- String [key](#)
- double [totalDistance](#)
- boolean [available](#)

Static Private Attributes

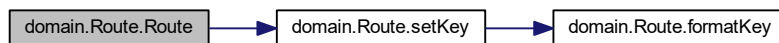
- static final int `MINIMAL_SIZE_OF_ID` = 2
- static final String `PATTERN_ID` = "[^a-zA-Z]"
- static final String `PATTERN_DOUBLE` = "[^0-9.]"
- static final String `PATTERN_NO_DUPLICATE` = "(\\D)\\1{1}"

6.17.1 Constructor & Destructor Documentation

6.17.1.1 Route() [1/3]

```
domain.Route.Route ( )
```

Constructor Here is the call graph for this function:



6.17.1.2 Route() [2/3]

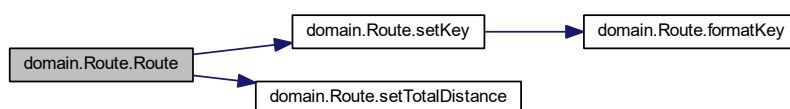
```
domain.Route.Route (
    String id,
    double total_distance )
```

Constructor

Parameters

<i>id</i>	key da rota
<i>total_distance</i>	distancia da rota

Here is the call graph for this function:



6.17.1.3 Route() [3/3]

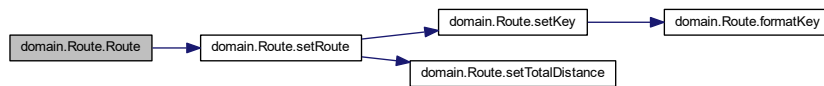
```
domain.Route.Route (
    String route )
```

Construtor

Parameters

<i>route</i>	Key+distancia em formato de string Ex: AB50
--------------	---

Here is the call graph for this function:



6.17.2 Member Function Documentation

6.17.2.1 formatKey()

```
static String domain.Route.formatKey (
    String key_route ) [static]
```

Formata uma string para o padrao do key de rota

Parameters

<i>key_route</i>	key a ser formatado
------------------	---------------------

6.17.2.5 getKeyOfStop()

```
String domain.Route.getKeyOfStop (
    int n_stop )
```

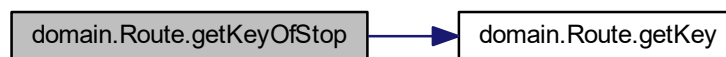
Retorna o key do trecho entre uma parada ate a proxima estacao

Parameters

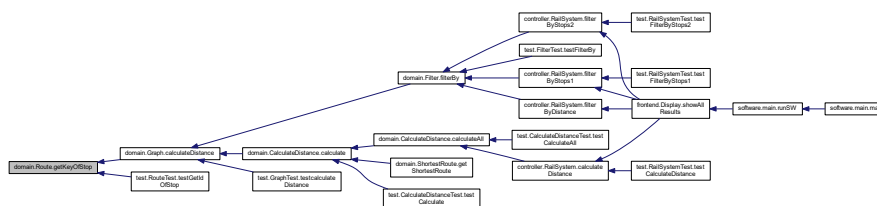
<code>n_stop</code>	numero da parada que se deseja o retorno do trecho
---------------------	--

Returns

Here is the call graph for this function:



Here is the caller graph for this function:



6.17.2.6 getNumberOfNodes()

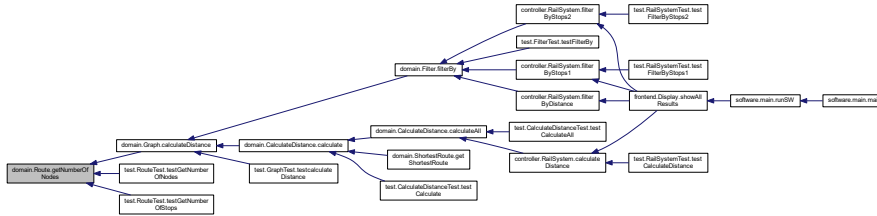
```
int domain.Route.getNumberOfNodes ( )
```

Retorna o numero de node des uma rota(viagem)

Returns

Numero de nodes

Here is the caller graph for this function:



6.17.2.7 `getNumberOfStops()`

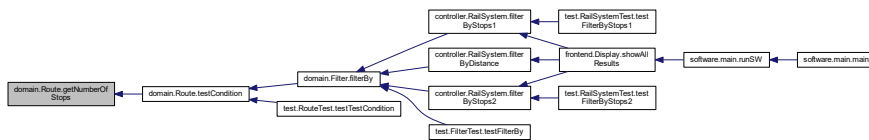
```
int domain.Route.getNumberOfStops ( )
```

Retorna o numero de paradas que serao feitas em uma rota(viagem) a partida nao e considerada como uma parada

Returns

Numero de paradas

Here is the caller graph for this function:



6.17.2.8 `getStartingTown()`

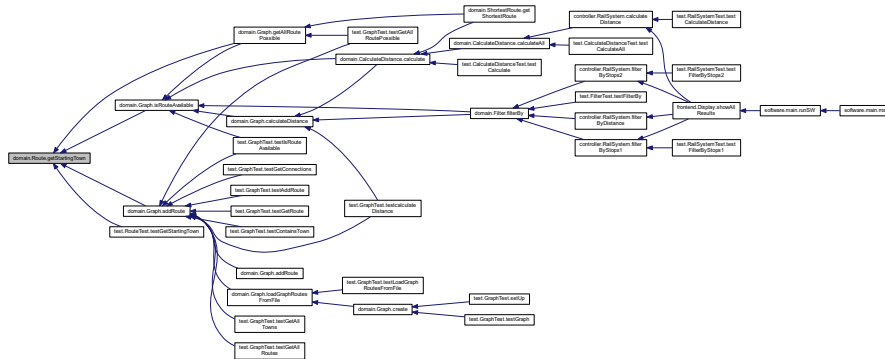
```
String domain.Route.getStartingTown ( )
```

Retona cidade de origem de uma rota(viagem)

Returns

Retorna a cidade de origem de uma rota(viagem)

Here is the caller graph for this function:



6.17.2.9 isAvailable()

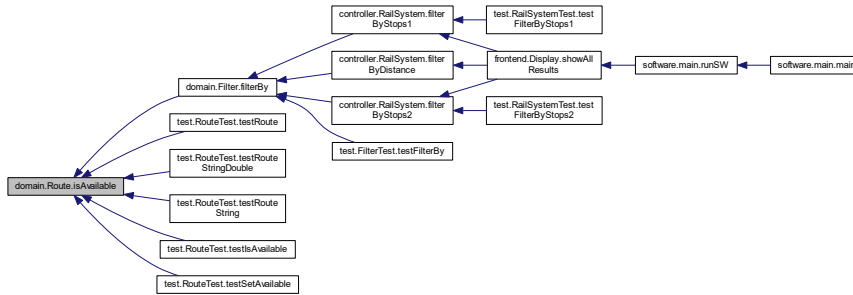
```
boolean domain.Route.isAvailable ( )
```

Retorna a disponibilidade de uma rota

Returns

Retorna se a rota e valida/disponivel

Here is the caller graph for this function:



6.17.2.10 setAvailable()

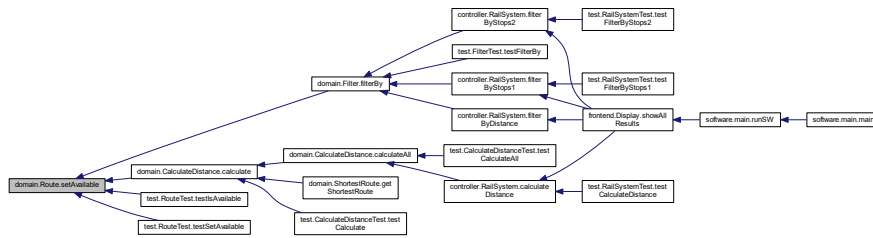
```
void domain.Route.setAvailable (
    boolean exist )
```

Set a disponibilidade da rota

Parameters

`exist`

Here is the caller graph for this function:



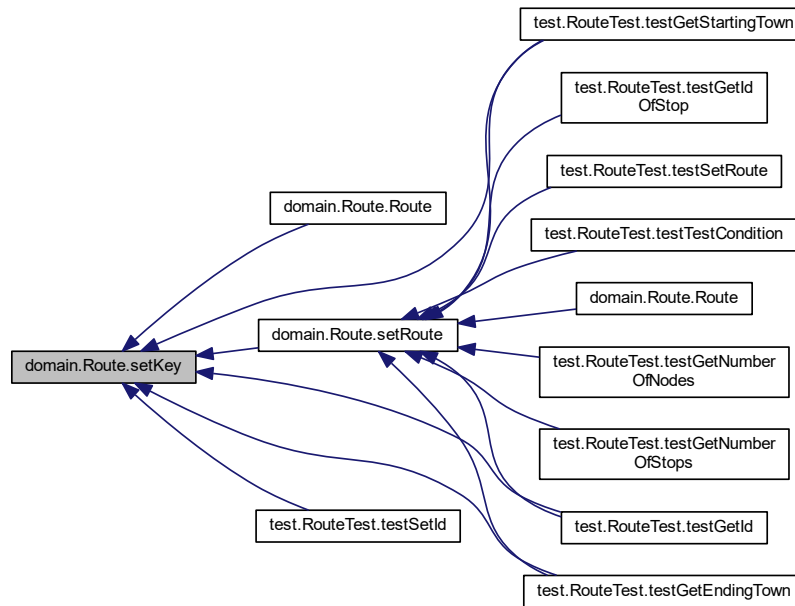
6.17.2.11 setKey()

```
void domain.Route.setKey (
    String r )
```

Here is the call graph for this function:



Here is the caller graph for this function:



6.17.2.12 setRoute()

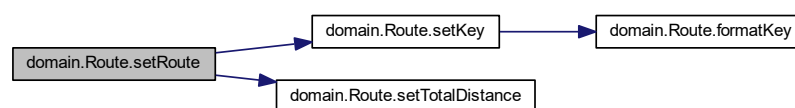
```
void domain.Route.setRoute (
    String route )
```

Configura o key e a distancia da rota atraves de uma string

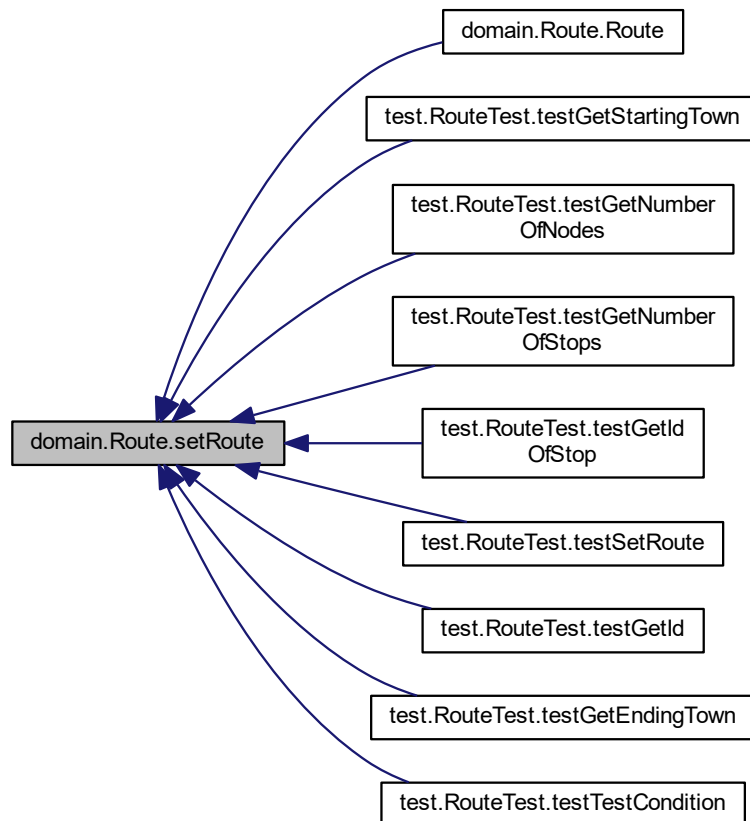
Parameters

<i>route</i>	Key+distancia em formato de string Ex: AB50
--------------	---

Here is the call graph for this function:



Here is the caller graph for this function:



6.17.2.13 setTotalDistance() [1/2]

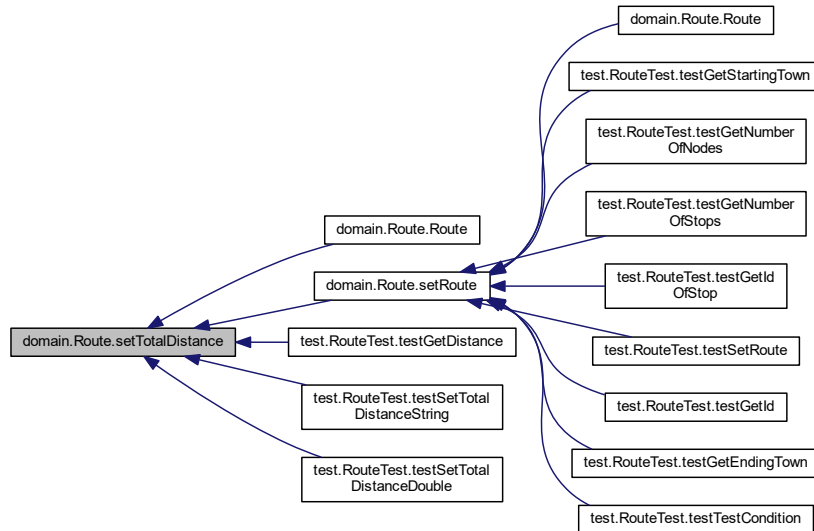
```
void domain.Route.setTotalDistance (
    String val )
```

Set a distancia de uma rota(viagem)

Parameters

<i>val</i>	valor da distancia(no formato de texto) que sera usado para definir a distancia total da rota.
------------	--

Here is the caller graph for this function:



6.17.2.14 setTotalDistance() [2/2]

```
void domain.Route.setTotalDistance (
    double val )
```

Set a distancia de uma rota(viagem)

Parameters

<i>val</i>	valor da distancia que sera usado para definir a distancia total da rota
------------	--

6.17.2.15 testCondition()

```
boolean domain.Route.testCondition (
    String condition,
    String operand,
    double valor2 )
```

Executa o teste da condicao

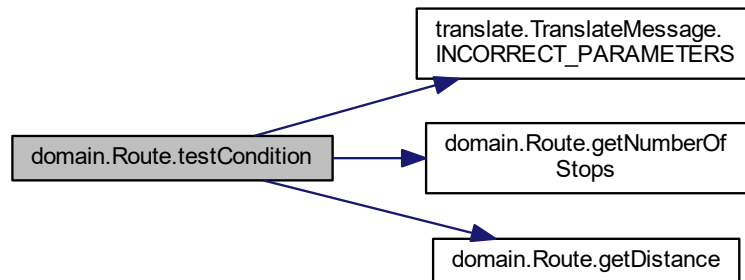
Parameters

<i>condition</i>	Condicao literal(texto) a ser considerada
<i>operand</i>	o operando literal (texto) que se deseja verificar
<i>valor2</i>	valor usado como referencia na verificacao

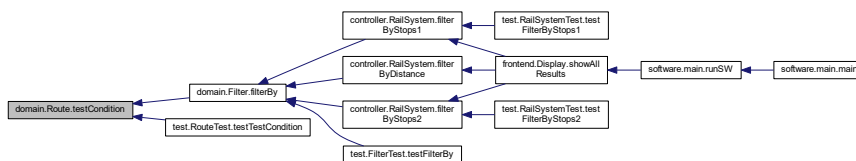
Returns

Retorna o resultado do teste

Here is the call graph for this function:



Here is the caller graph for this function:

**6.17.3 Member Data Documentation****6.17.3.1 available**

```
boolean domain.Route.available [private]
```

Indica se a rota existe

6.17.3.2 key

```
String domain.Route.key [private]
```

Identificador usado nos maps

6.17.3.3 MINIMAL_SIZE_OF_ID

```
final int domain.Route.MINIMAL_SIZE_OF_ID = 2 [static], [private]
```

6.17.3.4 PATTERN_DOUBLE

```
final String domain.Route.PATTERN_DOUBLE = "[^0-9.,]" [static], [private]
```

6.17.3.5 PATTERN_ID

```
final String domain.Route.PATTERN_ID = "[^a-zA-Z]" [static], [private]
```

6.17.3.6 PATTERN_NO_DUPLICATE

```
final String domain.Route.PATTERN_NO_DUPLICATE = "(\\D)\\1{1}" [static], [private]
```

6.17.3.7 totalDistance

```
double domain.Route.totalDistance [private]
```

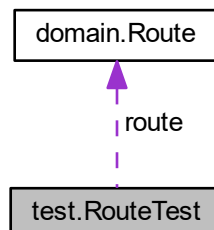
Distancia total da rota ou viagem

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

- <D:/workspace/TrainVagas/src/domain/Route.java>

6.18 test.RouteTest Class Reference

Collaboration diagram for test.RouteTest:



Public Member Functions

- void [setUp](#) () throws Exception
- void [testRoute](#) ()
- void [testRouteStringDouble](#) ()
- void [testRouteString](#) ()
- void [testGetStartingTown](#) ()
- void [testGetDistance](#) ()
- void [testGetNumberOfNodes](#) ()
- void [testGetNumberOfStops](#) ()
- void [testGetIdOfStop](#) ()
- void [testSetRoute](#) ()
- void [testSetId](#) ()
- void [testSetTotalDistanceString](#) ()
- void [testSetTotalDistanceDouble](#) ()
- void [testGetId](#) ()
- void [testGetEndingTown](#) ()
- void [testIsAvailable](#) ()
- void [testSetAvailable](#) ()
- void [testFormatKey](#) ()
- void [testTestCondition](#) ()

Package Attributes

- [Route route](#)

6.18.1 Member Function Documentation

6.18.1.1 setUp()

`void test.RouteTest.setUp () throws Exception`

6.18.1.2 testFormatKey()

`void test.RouteTest.testFormatKey ()`

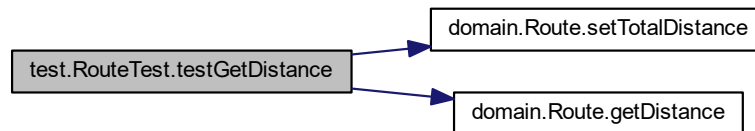
Here is the call graph for this function:



6.18.1.3 testGetDistance()

```
void test.RouteTest.testGetDistance ( )
```

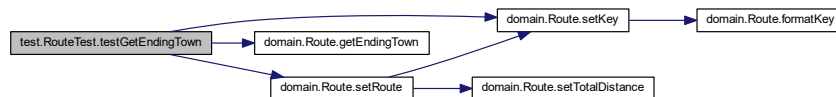
Here is the call graph for this function:



6.18.1.4 testGetEndingTown()

```
void test.RouteTest.testGetEndingTown ( )
```

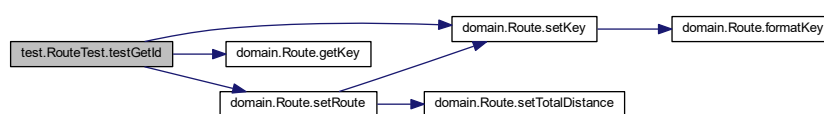
Here is the call graph for this function:



6.18.1.5 testGetId()

```
void test.RouteTest.testGetId ( )
```

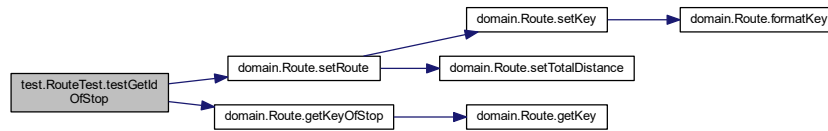
Here is the call graph for this function:



6.18.1.6 testGetIdOfStop()

```
void test.RouteTest.testGetIdOfStop ( )
```

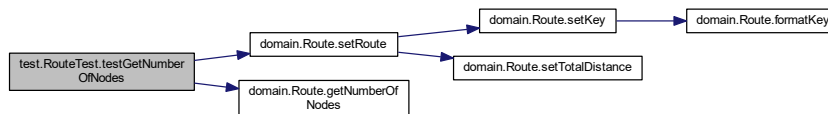
Here is the call graph for this function:



6.18.1.7 testGetNumberOfNodes()

```
void test.RouteTest.testGetNumberOfNodes ( )
```

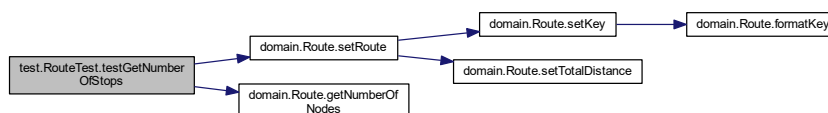
Here is the call graph for this function:



6.18.1.8 testGetNumberOfStops()

```
void test.RouteTest.testGetNumberOfStops ( )
```

Here is the call graph for this function:



6.18.1.9 testGetStartingTown()

```
void test.RouteTest.testGetStartingTown ( )
```

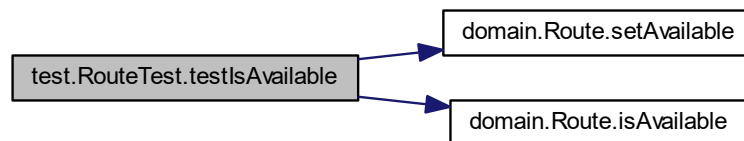
Here is the call graph for this function:



6.18.1.10 testIsAvailable()

```
void test.RouteTest.testIsAvailable ( )
```

Here is the call graph for this function:



6.18.1.11 testRoute()

```
void test.RouteTest.testRoute ( )
```

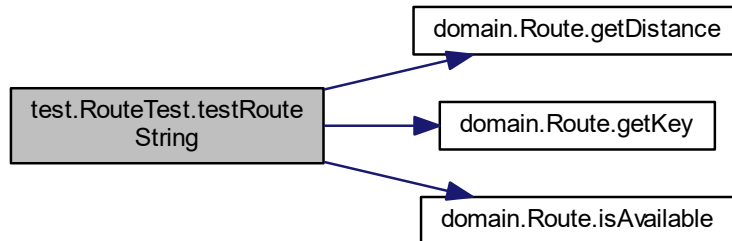
Here is the call graph for this function:



6.18.1.12 testRouteString()

```
void test.RouteTest.testRouteString ( )
```

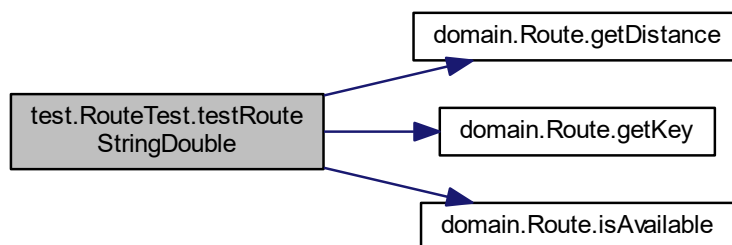
Here is the call graph for this function:



6.18.1.13 testRouteStringDouble()

```
void test.RouteTest.testRouteStringDouble ( )
```

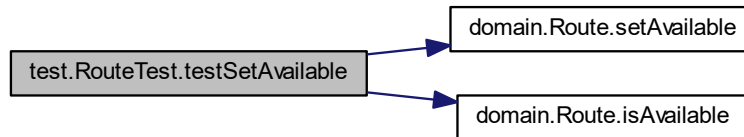
Here is the call graph for this function:



6.18.1.14 testSetAvailable()

```
void test.RouteTest.testSetAvailable ( )
```

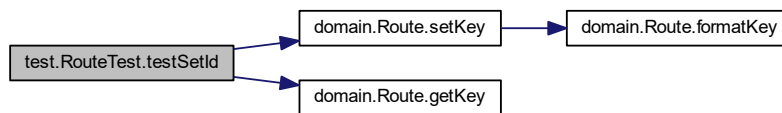
Here is the call graph for this function:



6.18.1.15 testSetId()

```
void test.RouteTest.testSetId ( )
```

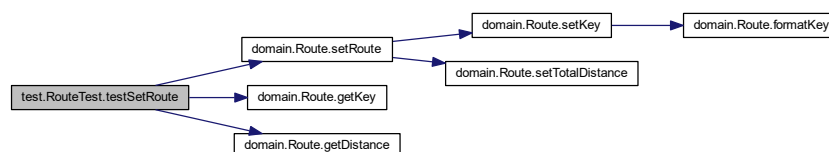
Here is the call graph for this function:



6.18.1.16 testSetRoute()

```
void test.RouteTest.testSetRoute ( )
```

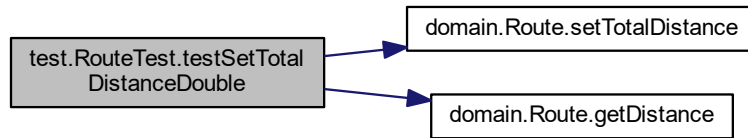
Here is the call graph for this function:



6.18.1.17 testSetTotalDistanceDouble()

```
void test.RouteTest.testSetTotalDistanceDouble ( )
```

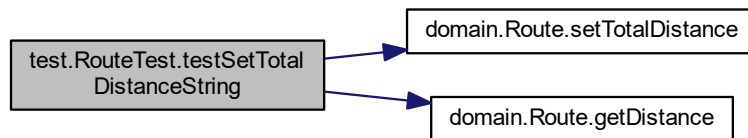
Here is the call graph for this function:



6.18.1.18 testSetTotalDistanceString()

```
void test.RouteTest.testSetTotalDistanceString ( )
```

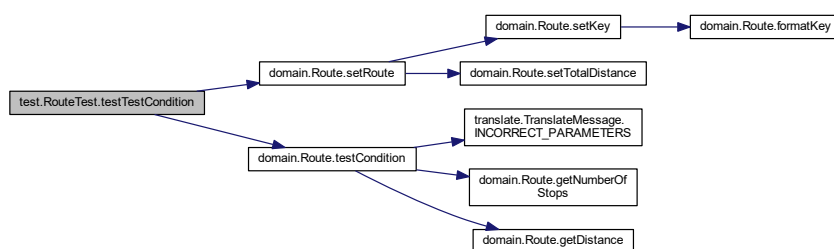
Here is the call graph for this function:



6.18.1.19 testTestCondition()

```
void test.RouteTest.testTestCondition ( )
```

Here is the call graph for this function:



6.18.2 Member Data Documentation

6.18.2.1 route

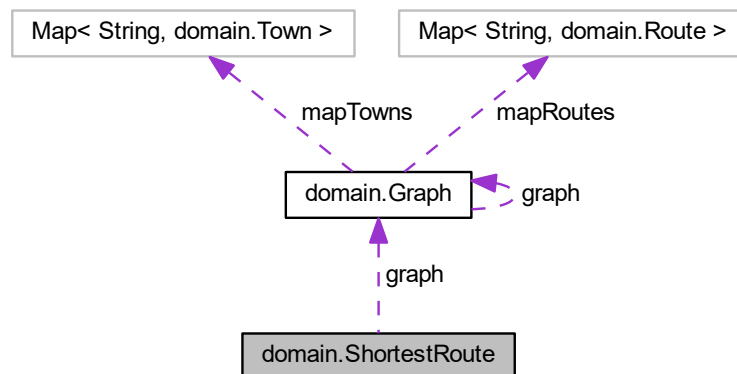
`Route` `test.RouteTest.route` [package]

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

- `D:/workspace/TrainVagas/src/test/RouteTest.java`

6.19 domain.ShortestRoute Class Reference

Collaboration diagram for `domain.ShortestRoute`:



Public Member Functions

- `Map<String,Route[]> getShortestRoute ()`
- `Route [] getShortestRoute (Route trip)`

Package Attributes

- `Graph graph = Graph.create()`

6.19.1 Member Function Documentation

6.19.1.1 getShortestRoute() [1/2]

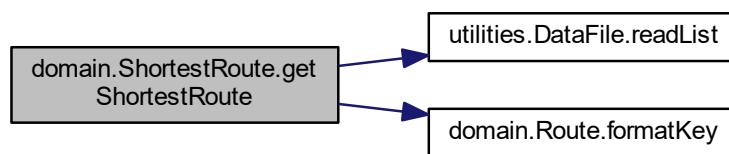
```
Map<String ,Route[]> domain.ShortestRoute.getShortestRoute ( )
```

Retorna uma lista das rotas mais curtas para cada viagem(rota) com entrada no arquivo input.txt

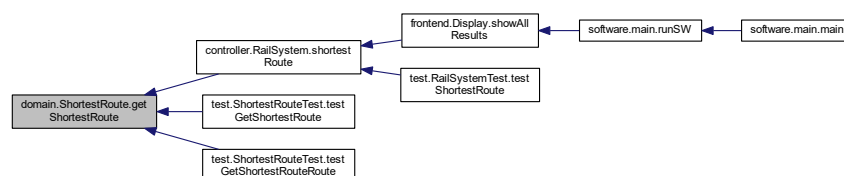
Returns

Lista de rotas mais curtas para cada viagem(rota)

Here is the call graph for this function:



Here is the caller graph for this function:



6.19.1.2 getShortestRoute() [2/2]

```
Route [] domain.ShortestRoute.getShortestRoute (
    Route trip )
```

Retorna uma lista das rotas mais curtas para uma viagem(rota)

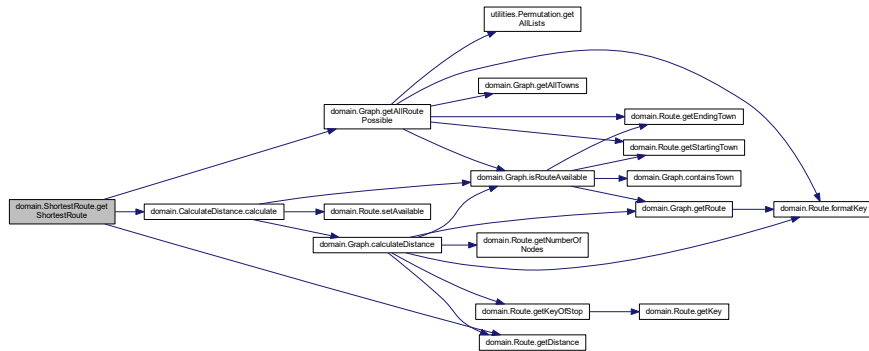
Parameters

<code>trip</code>	key de uma viagem(rota) que se deseja solicitar as rotas mais curtas
-------------------	--

Returns

Lista de rotas mais curtas

Here is the call graph for this function:



6.19.2 Member Data Documentation

6.19.2.1 graph

```
Graph domain.ShortestRoute.graph = Graph.create() [package]
```

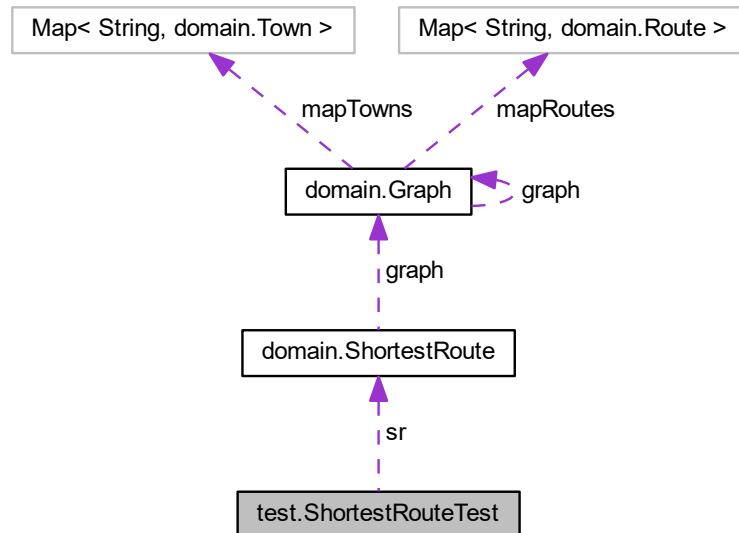
Rotas disponiveis no sistema

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

- <D:/workspace/TrainVagas/src/domain/ShortestRoute.java>

6.20 test.ShortestRouteTest Class Reference

Collaboration diagram for test.ShortestRouteTest:



Public Member Functions

- void [setUp](#) () throws Exception
- void [testGetShortestRoute](#) ()
- void [testGetShortestRouteRoute](#) ()

Package Attributes

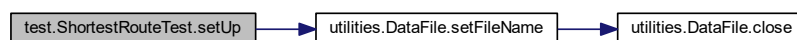
- [ShortestRoute](#) `sr`

6.20.1 Member Function Documentation

6.20.1.1 setUp()

`void test.ShortestRouteTest.setUp () throws Exception`

Here is the call graph for this function:



6.20.1.2 testGetShortestRoute()

```
void test.ShortestRouteTest.testGetShortestRoute ( )
```

Here is the call graph for this function:



6.20.1.3 testGetShortestRouteRoute()

```
void test.ShortestRouteTest.testGetShortestRouteRoute ( )
```

Here is the call graph for this function:



6.20.2 Member Data Documentation

6.20.2.1 sr

```
ShortestRoute test.ShortestRouteTest.sr [package]
```

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

- [D:/workspace/TrainVagas/src/test/ShortestRouteTest.java](#)

6.21 domain.Town Class Reference

Public Member Functions

- [Town](#) (String id)
- String [getKey](#) ()
- void [setKey](#) (String id)

Static Public Member Functions

- static String [formatKey](#) (String key_route)

Private Attributes

- String [key](#)

Static Private Attributes

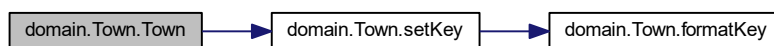
- static final int [MINIMAL_SIZE_OF_ID](#) = 1
- static final String [PATTERN_ID](#) = "[^a-zA-Z]"

6.21.1 Constructor & Destructor Documentation

6.21.1.1 Town()

```
domain.Town.Town (
    String id )
```

Here is the call graph for this function:



6.21.2 Member Function Documentation

6.21.2.1 formatKey()

```
static String domain.Town.formatKey (
    String key_route ) [static]
```

Formata uma string para o padrao do key de casa

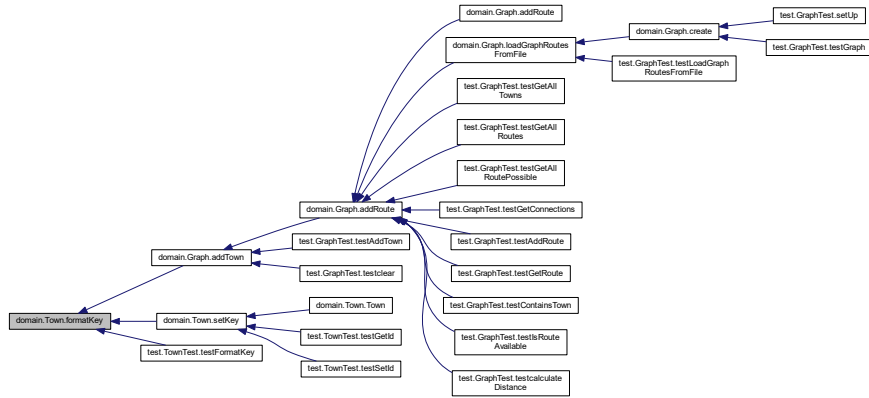
Parameters

<i>key_route</i>	key a ser formatado
------------------	---------------------

Returns

string formatada

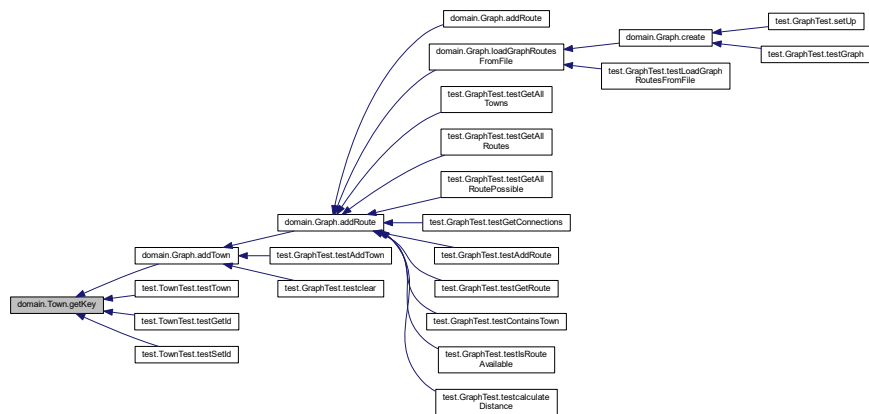
Here is the caller graph for this function:



6.21.2.2 getKey()

```
String domain.Town.getKey ( )
```

Here is the caller graph for this function:



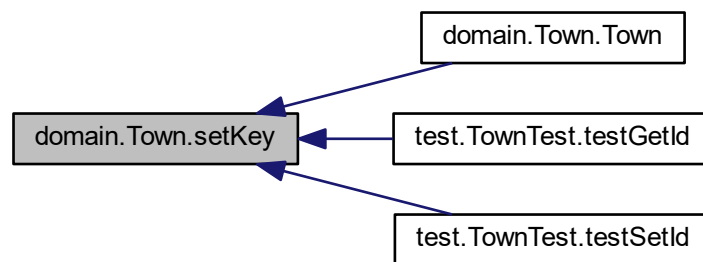
6.21.2.3 setKey()

```
void domain.Town.setKey (  
    String id )
```

Here is the call graph for this function:



Here is the caller graph for this function:



6.21.3 Member Data Documentation

6.21.3.1 key

```
String domain.Town.key [private]
```

Identificador usado nos maps

6.21.3.2 MINIMAL_SIZE_OF_ID

```
final int domain.Town.MINIMAL_SIZE_OF_ID = 1 [static], [private]
```

6.21.3.3 PATTERN_ID

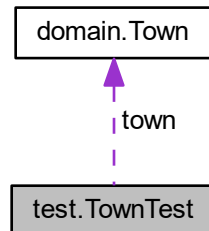
```
final String domain.Town.PATTERN_ID ="[^a-zA-Z]" [static], [private]
```

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

- <D:/workspace/TrainVagas/src/domain/Town.java>

6.22 test.TownTest Class Reference

Collaboration diagram for test.TownTest:



Public Member Functions

- void [setUp](#) () throws Exception
- void [testTown](#) ()
- void [testGetId](#) ()
- void [testSetId](#) ()
- void [testFormatKey](#) ()

Package Attributes

- [Town town](#)

6.22.1 Member Function Documentation

6.22.1.1 setUp()

```
void test.TownTest.setUp ( ) throws Exception
```

6.22.1.2 testFormatKey()

```
void test.TownTest.testFormatKey ( )
```

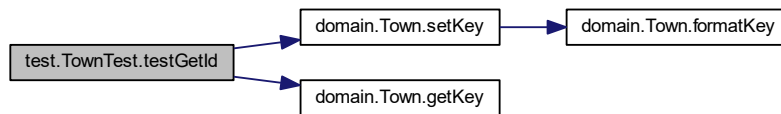
Here is the call graph for this function:



6.22.1.3 testGetId()

```
void test.TownTest.testGetId ( )
```

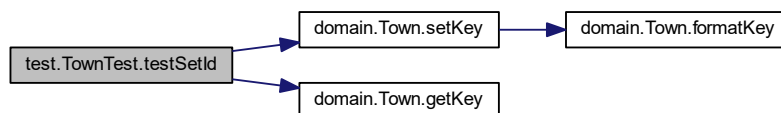
Here is the call graph for this function:



6.22.1.4 testSetId()

```
void test.TownTest.testSetId ( )
```

Here is the call graph for this function:



6.22.1.5 testTown()

```
void test.TownTest.testTown ( )
```

Here is the call graph for this function:



6.22.2 Member Data Documentation

6.22.2.1 town

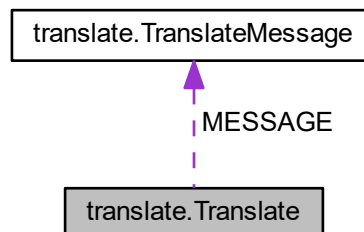
```
Town test.TownTest.town [package]
```

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

- [D:/workspace/TrainVagas/src/test/TownTest.java](#)

6.23 translate.Translate Class Reference

Collaboration diagram for `translate.Translate`:



Static Public Attributes

- static final [TranslateMessage MESSAGE](#) = new [PtBR\(\)](#)

6.23.1 Member Data Documentation

6.23.1.1 MESSAGE

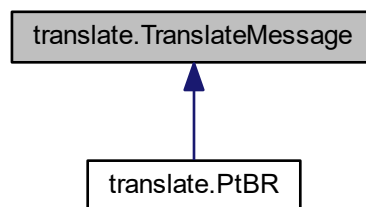
```
final TranslateMessage translate.Translate.MESSAGE = new PtBR\(\) [static]
```

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

- D:/workspace/TrainVagas/src/translate/[Translate.java](#)

6.24 translate.TranslateMessage Interface Reference

Inheritance diagram for translate.TranslateMessage:



Public Member Functions

- String [TOW_NO_EXIST](#) ()
- String [ROUTE_NO_AVAILABLE](#) ()
- String [INCORRECT_PARAMETERS](#) ()
- String [FILE_NOT_FOUND](#) ()
- String [PROPERTY_NOT_FOUND](#) ()

6.24.1 Member Function Documentation

6.24.1.1 FILE_NOT_FOUND()

String translate.TranslateMessage.FILE_NOT_FOUND ()

Implemented in [translate.PtBR](#).

Here is the caller graph for this function:

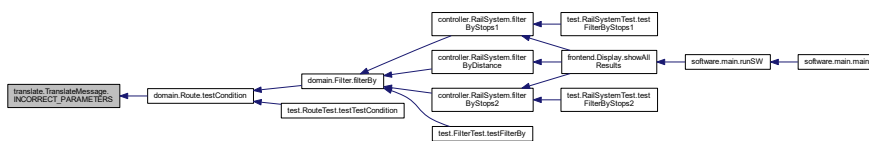


6.24.1.2 INCORRECT_PARAMETERS()

String translate.TranslateMessage.INCORRECT_PARAMETERS ()

Implemented in [translate.PtBR](#).

Here is the caller graph for this function:



6.24.1.3 PROPERTY_NOT_FOUND()

String translate.TranslateMessage.PROPERTY_NOT_FOUND ()

Implemented in [translate.PtBR](#).

6.24.1.4 ROUTE_NO_AVAILABLE()

String translate.TranslateMessage.ROUTE_NO_AVAILABLE ()

Implemented in [translate.PtBR](#).

Here is the caller graph for this function:



6.24.1.5 TOW_NO_EXIST()

```
String translate.TranslateMessage.TOW_NO_EXIST ( )
```

Implemented in [translate.PtBR](#).

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

- [D:/workspace/TrainVagas/src/translate/TranslateMessage.java](#)

Chapter 7

File Documentation

7.1 D:/workspace/TrainVagas/src/controller/RailSystem.java File Reference

Software de gerenciamento das rotas para uma solucao de mobilidade sobre trilhos.

Classes

- class [controller.RailSystem](#)

Packages

- package [controller](#)

7.1.1 Detailed Description

Software de gerenciamento das rotas para uma solucao de mobilidade sobre trilhos.

Classe Controle intermedia a interacao entre o View(TUI) e o domain

Authors

Jefferson Alves

Date

24.04.2017

Version

0.1

7.2 D:/workspace/TrainVagas/src/domain/CalculateDistance.java File Reference

Software de gerenciamento das rotas para uma solucao de mobilidade sobre trilhos.

Classes

- class [domain.CalculateDistance](#)

Packages

- package [domain](#)

7.2.1 Detailed Description

Software de gerenciamento das rotas para uma solucao de mobilidade sobre trilhos.

Calcula as distancias das rotas indicadas no arquivo 'input.txt' na propriedade 'distance.routes'

Authors

Jefferson Alves

Date

24.04.2017

Version

0.1

7.3 D:/workspace/TrainVagas/src/domain/Filter.java File Reference

Software de gerenciamento das rotas para uma solucao de mobilidade sobre trilhos.

Classes

- class [domain.Filter](#)

Packages

- package [domain](#)

7.3.1 Detailed Description

Software de gerenciamento das rotas para uma solucao de mobilidade sobre trilhos.

Filtra as rotas indicadas no arquivo 'input.txt' na propriedade 'filtertrips[X].routes' utilizando a condicao indicada na propriedade filtertrips[1].condition

Authors

Jefferson Alves

Date

24.04.2017

Version

0.1

7.4 D:/workspace/TrainVagas/src/domain/Graph.java File Reference

Software de gerenciamento das rotas para uma solucao de mobilidade sobre trilhos.

Classes

- class [domain.Graph](#)

Packages

- package [domain](#)

7.4.1 Detailed Description

Software de gerenciamento das rotas para uma solucao de mobilidade sobre trilhos.

Gerenciamento dos containers de cidade e rotas existentes entre as cidades As rotas sao carregadas via o arquivo 'input.txt' propriedade 'graph.routes' e baseado nelas sao mapeadas as cidades

Authors

Jefferson Alves

Date

24.04.2017

Version

0.1

7.5 D:/workspace/TrainVagas/src/domain/Route.java File Reference

Software de gerenciamento das rotas para uma solucao de mobilidade sobre trilhos.

Classes

- class [domain.Route](#)

Packages

- package [domain](#)

7.5.1 Detailed Description

Software de gerenciamento das rotas para uma solucao de mobilidade sobre trilhos.

Abstracao de rotas(viagens)

Authors

Jefferson Alves

Date

24.04.2017

Version

0.1

7.6 D:/workspace/TrainVagas/src/domain/ShortestRoute.java File Reference

Software de gerenciamento das rotas para uma solucao de mobilidade sobre trilhos.

Classes

- class [domain.ShortestRoute](#)

Packages

- package [domain](#)

7.6.1 Detailed Description

Software de gerenciamento das rotas para uma solucao de mobilidade sobre trilhos.

Identifica as rotas mais curtas para as viagens(rotas) que foram indicada no arquivo input.txt na propriedade shourtestRouter.trip

Authors

Jefferson Alves

Date

24.04.2017

Version

0.1

7.7 D:/workspace/TrainVagas/src/domain/Town.java File Reference

Software de gerenciamento das rotas para uma solucao de mobilidade sobre trilhos.

Classes

- class [domain.Town](#)

Packages

- package [domain](#)

7.7.1 Detailed Description

Software de gerenciamento das rotas para uma solucao de mobilidade sobre trilhos.

Abstracao de cidades

Authors

Jefferson Alves

Date

24.04.2017

Version

0.1

7.8 D:/workspace/TrainVagas/src/frontend/Display.java File Reference

Software de gerenciamento das rotas para uma solucao de mobilidade sobre trilhos.

Classes

- class [frontend.Display](#)

Packages

- package [frontend](#)

7.8.1 Detailed Description

Software de gerenciamento das rotas para uma solucao de mobilidade sobre trilhos.

Responsavel pelo View - Text User interface(TUI)

Authors

Jefferson Alves

Date

24.04.2017

Version

0.1 Recebe o conteudo a ser mostrado atraves de arrays do tipo roto

Layout do TUI

Output | Resultado | Detalhes #1 | 9 | [ABC] #2 | 5 | [AD] #3 | 13 | [ADC] #4 | 22 | [AEBCD] #5 | Rota nao disponivel
| [AED] #6 | 2 | [CDC] [CEBC] #7 | 3 | [ABCDC] [ADCDC] [ADEBC] #8 | 9 | [ABC] #9 | 9 | [BCEB] #10 | 7 | [CDC]
[CEBC] [CEBCDC] [CDCEBC] [CDEBC] [CEBCEBC] [CEBCEBCEBC]

7.9 D:/workspace/TrainVagas/src/software/main.java File Reference

Software de gerenciamento das rotas para uma solucao de mobilidade sobre trilhos.

Classes

- class [software.main](#)

Packages

- package [software](#)

7.9.1 Detailed Description

Software de gerenciamento das rotas para uma solucao de mobilidade sobre trilhos.

Metodo main do software estruturado em tres partes

```
statupSW()      Realiza todas as inicializacoes necessarias
runSW()        Executa a aplicacao principal
shutdownSW()   Finaliza o software
```

Authors

Jefferson Alves

Date

24.04.2017

Version

0.1

Parameters

<i>none</i>	
-------------	--

7.10 D:/workspace/TrainVagas/src/test/CalculateDistanceTest.java File Reference

Classes

- class [test.CalculateDistanceTest](#)

Packages

- package [test](#)

7.11 D:/workspace/TrainVagas/src/test/DataFileTest.java File Reference

Classes

- class [test.DataFileTest](#)

Packages

- package [test](#)

7.12 D:/workspace/TrainVagas/src/test/FilterTest.java File Reference

Classes

- class [test.FilterTest](#)

Packages

- package [test](#)

7.13 D:/workspace/TrainVagas/src/test/GraphTest.java File Reference

Classes

- class [test.GraphTest](#)

Packages

- package [test](#)

7.14 D:/workspace/TrainVagas/src/test/PermutationTest.java File Reference

Classes

- class [test.PermutationTest](#)

Packages

- package [test](#)

7.15 D:/workspace/TrainVagas/src/test/RailSystemTest.java File Reference

Classes

- class [test.RailSystemTest](#)

Packages

- package [test](#)

7.16 D:/workspace/TrainVagas/src/test/RouteTest.java File Reference

Classes

- class [test.RouteTest](#)

Packages

- package [test](#)

7.17 D:/workspace/TrainVagas/src/test/ShortestRouteTest.java File Reference

Classes

- class [test.ShortestRouteTest](#)

Packages

- package [test](#)

7.18 D:/workspace/TrainVagas/src/test/TownTest.java File Reference

Classes

- class [test.TownTest](#)

Packages

- package [test](#)

7.19 D:/workspace/TrainVagas/src/translate/PtBR.java File Reference

Software de gerenciamento das rotas para uma solucao de mobilidade sobre trilhos.

Classes

- class [translate.PtBR](#)

Packages

- package [translate](#)

7.19.1 Detailed Description

Software de gerenciamento das rotas para uma solucao de mobilidade sobre trilhos.

Textos em protugues

Authors

Jefferson Alves

Date

24.04.2017

Version

0.1

Gerencia os textos traduzidos

Authors

Jefferson Alves

Date

24.04.2017

Version

0.1

7.20 D:/workspace/TrainVagas/src/translate/Translate.java File Reference

Classes

- class [translate.Translate](#)

Packages

- package [translate](#)

7.21 D:/workspace/TrainVagas/src/translate/TranslateMessage.java File Reference

Software de gerenciamento das rotas para uma solucao de mobilidade sobre trilhos.

Classes

- interface [translate.TranslateMessage](#)

Packages

- package [translate](#)

7.21.1 Detailed Description

Software de gerenciamento das rotas para uma solucao de mobilidade sobre trilhos.

Interface para acesso aos textos traduzidos

Authors

Jefferson Alves

Date

24.04.2017

Version

0.1

7.22 D:/workspace/TrainVagas/src/utilities/DataFile.java File Reference

Classes

- class [utilities.DataFile](#)

Packages

- package [utilities](#)

7.23 D:/workspace/TrainVagas/src/utilities/FileProperty.java File Reference

Classes

- enum [utilities.FileProperty](#)

Packages

- package [utilities](#)

7.24 D:/workspace/TrainVagas/src/utilities/Permutation.java File Reference

Classes

- class [utilities.Permutation](#)

Packages

- package [utilities](#)

Index

- addRoute
 - domain::Graph, [42, 43](#)
- addTown
 - domain::Graph, [44](#)
- available
 - domain::Route, [87](#)
- calculate
 - domain::CalculateDistance, [12](#)
- calculateAll
 - domain::CalculateDistance, [12](#)
- calculateDistance
 - controller::RailSystem, [68](#)
 - domain::Graph, [45](#)
- cd
 - test::CalculateDistanceTest, [15](#)
- checkFileExist
 - utilities::DataFile, [17](#)
- clear
 - domain::Graph, [46](#)
- close
 - utilities::DataFile, [17](#)
- containsTown
 - domain::Graph, [46](#)
- controller, [9](#)
- controller.RailSystem, [68](#)
- controller::RailSystem
 - calculateDistance, [68](#)
 - delNit, [68](#)
 - filterByDistance, [69](#)
 - filterByStops1, [69](#)
 - filterByStops2, [70](#)
 - init, [71](#)
 - shortestRoute, [72](#)
- create
 - domain::Graph, [47](#)
- D:/workspace/TrainVagas/src/controller/RailSystem.↔
java, [111](#)
- D:/workspace/TrainVagas/src/domain/Calculate↔
Distance.java, [112](#)
- D:/workspace/TrainVagas/src/domain/Filter.java, [112](#)
- D:/workspace/TrainVagas/src/domain/Graph.java, [113](#)
- D:/workspace/TrainVagas/src/domain/Route.java, [114](#)
- D:/workspace/TrainVagas/src/domain/ShortestRoute.↔
java, [114](#)
- D:/workspace/TrainVagas/src/domain/Town.java, [115](#)
- D:/workspace/TrainVagas/src/frontend/Display.java, [116](#)
- D:/workspace/TrainVagas/src/software/main.java, [116](#)
- D:/workspace/TrainVagas/src/test/CalculateDistance↔
Test.java, [117](#)
- D:/workspace/TrainVagas/src/test/DataFileTest.java,
[117](#)
- D:/workspace/TrainVagas/src/test/FilterTest.java, [118](#)
- D:/workspace/TrainVagas/src/test/GraphTest.java, [118](#)
- D:/workspace/TrainVagas/src/test/PermutationTest.java,
[118](#)
- D:/workspace/TrainVagas/src/test/RailSystemTest.java,
[118](#)
- D:/workspace/TrainVagas/src/test/RouteTest.java, [119](#)
- D:/workspace/TrainVagas/src/test/ShortestRouteTest.↔
java, [119](#)
- D:/workspace/TrainVagas/src/test/TownTest.java, [119](#)
- D:/workspace/TrainVagas/src/translate/PtBR.java, [119](#)
- D:/workspace/TrainVagas/src/translate/Translate.java,
[120](#)
- D:/workspace/TrainVagas/src/translate/Translate↔
Message.java, [121](#)
- D:/workspace/TrainVagas/src/utilities/DataFile.java, [121](#)
- D:/workspace/TrainVagas/src/utilities/FileProperty.java,
[121](#)
- D:/workspace/TrainVagas/src/utilities/Permutation.java,
[122](#)
- DISTANCE_CONDITION
 - utilities::DataFile, [24](#)
- DISTANCE_ROUTES
 - utilities::FileProperty, [36](#)
- delNit
 - controller::RailSystem, [68](#)
- domain, [9](#)
- domain.CalculateDistance, [11](#)
- domain.Filter, [37](#)
- domain.Graph, [41](#)
- domain.Route, [75](#)
- domain.ShortestRoute, [96](#)
- domain.Town, [100](#)
- domain::CalculateDistance
 - calculate, [12](#)
 - calculateAll, [12](#)
 - graph, [13](#)
- domain::Filter
 - filterBy, [38](#)
 - graph, [39](#)
- domain::Graph
 - addRoute, [42, 43](#)
 - addTown, [44](#)
 - calculateDistance, [45](#)
 - clear, [46](#)

- containsTown, 46
- create, 47
- getAllRoutePossible, 47
- getAllRoutes, 48
- getAllTowns, 49
- getConnections, 49
- getRoute, 50
- Graph, 42
- graph, 53
- isRouteAvailable, 51
- loadGraphRoutesFromFile, 52
- mapRoutes, 53
- mapTowns, 53
- domain::Route
 - available, 87
 - formatKey, 77
 - getDistance, 78
 - getEndingTown, 79
 - getKey, 79
 - getKeyOfStop, 79
 - getNumberOfNodes, 80
 - getNumberOfStops, 81
 - getStartingTown, 81
 - isAvailable, 82
 - key, 87
 - MINIMAL_SIZE_OF_ID, 87
 - PATTERN_DOUBLE, 88
 - PATTERN_ID, 88
 - PATTERN_NO_DUPLICATE, 88
 - Route, 76, 77
 - setAvailable, 82
 - setKey, 83
 - setRoute, 84
 - setTotalDistance, 85, 86
 - testCondition, 86
 - totalDistance, 88
- domain::ShortestRoute
 - getShortestRoute, 96, 97
 - graph, 98
- domain::Town
 - formatKey, 101
 - getKey, 102
 - key, 103
 - MINIMAL_SIZE_OF_ID, 103
 - PATTERN_ID, 103
 - setKey, 102
 - Town, 101
- FILE_NOT_FOUND
 - translate::PtBR, 66, 67
 - translate::TranslateMessage, 107
- FILE_TESTS_CASES
 - utilities::DataFile, 25
- FILTER_CONDITION
 - utilities::FileProperty, 36
- FILTER_ROUTES
 - utilities::FileProperty, 36
- FORMAT_COL1
 - frontend::Display, 33
- FORMAT_DECIMAL
 - frontend::Display, 33
- FORMAT_DOUBLE
 - frontend::Display, 34
- FORMAT_ROUTE
 - frontend::Display, 34
- FORMAT_STRING
 - frontend::Display, 34
- file_name
 - utilities::DataFile, 25
- FileProperty
 - utilities::FileProperty, 35
- filter
 - test::FilterTest, 40
- filterBy
 - domain::Filter, 38
- filterByDistance
 - controller::RailSystem, 69
- filterByStops1
 - controller::RailSystem, 69
- filterByStops2
 - controller::RailSystem, 70
- formatKey
 - domain::Route, 77
 - domain::Town, 101
- frontend, 9
- frontend.Display, 29
- frontend::Display
 - FORMAT_COL1, 33
 - FORMAT_DECIMAL, 33
 - FORMAT_DOUBLE, 34
 - FORMAT_ROUTE, 34
 - FORMAT_STRING, 34
 - index_line, 34
 - NEW_LINE, 34
 - printHead, 30
 - printLine, 30
 - printResultCalculateDistance, 30
 - printResultFilter, 31
 - printResultShortest, 32
 - SEPARATOR, 34
 - showAllResults, 33
- GRAPH_ROUTES
 - utilities::FileProperty, 36
- getAllLists
 - utilities::Permutation, 63
- getAllRoutePossible
 - domain::Graph, 47
- getAllRoutes
 - domain::Graph, 48
- getAllTowns
 - domain::Graph, 49
- getConnections
 - domain::Graph, 49
- getDistance
 - domain::Route, 78
- getEndingTown
 - domain::Route, 79

- getFileName
 - utilities::DataFile, 17
- getKey
 - domain::Route, 79
 - domain::Town, 102
- getKeyOfStop
 - domain::Route, 79
- getName
 - utilities::FileProperty, 35
- getNumberOfNodes
 - domain::Route, 80
- getNumberOfStops
 - domain::Route, 81
- getRoute
 - domain::Graph, 50
- getShortestRoute
 - domain::ShortestRoute, 96, 97
- getStartingTown
 - domain::Route, 81
- Graph
 - domain::Graph, 42
- graph
 - domain::CalculateDistance, 13
 - domain::Filter, 39
 - domain::Graph, 53
 - domain::ShortestRoute, 98
 - test::GraphTest, 60
- INCORRECT_PARAMETERS
 - translate::PtBR, 66, 67
 - translate::TranslateMessage, 108
- index_line
 - frontend::Display, 34
- init
 - controller::RailSystem, 71
- input
 - utilities::DataFile, 25
- isAvailable
 - domain::Route, 82
- isRouteAvailable
 - domain::Graph, 51
- key
 - domain::Route, 87
 - domain::Town, 103
- loadGraphRoutesFromFile
 - domain::Graph, 52
- MESSAGE
 - translate::Translate, 107
- MINIMAL_SIZE_OF_ID
 - domain::Route, 87
 - domain::Town, 103
- main
 - software::main, 60
- mapRoutes
 - domain::Graph, 53
- mapTowns
 - domain::Graph, 53
- NEW_LINE
 - frontend::Display, 34
- name
 - utilities::FileProperty, 36
- open
 - utilities::DataFile, 18
- PATH_FILE
 - utilities::DataFile, 25
- PATTERN_CONDITION
 - utilities::DataFile, 25
- PATTERN_DOUBLE
 - domain::Route, 88
 - utilities::DataFile, 25
- PATTERN_INTEGER
 - utilities::DataFile, 25
- PATTERN_ID
 - domain::Route, 88
 - domain::Town, 103
- PATTERN_LIST
 - utilities::DataFile, 25
- PATTERN_NO_DUPLICATE
 - domain::Route, 88
- PATTERN_OPERAND
 - utilities::DataFile, 26
- PROPERTY_NOT_FOUND
 - translate::PtBR, 66, 67
 - translate::TranslateMessage, 108
- printHead
 - frontend::Display, 30
- printLine
 - frontend::Display, 30
- printResultCalculateDistance
 - frontend::Display, 30
- printResultFilter
 - frontend::Display, 31
- printResultShortest
 - frontend::Display, 32
- prop
 - utilities::DataFile, 26
- ROUTE_NO_AVAILABLE
 - translate::PtBR, 66, 67
 - translate::TranslateMessage, 108
- readCondition
 - utilities::DataFile, 18
- readDouble
 - utilities::DataFile, 19
- readInteger
 - utilities::DataFile, 20
- readList
 - utilities::DataFile, 21, 22
- readLiteralOperand
 - utilities::DataFile, 23
- Route
 - domain::Route, 76, 77

- route
 - test::RouteTest, 96
- runSW
 - software::main, 61
- SEPARATOR
 - frontend::Display, 34
 - utilities::DataFile, 26
- SHORTEST_ROUTES
 - utilities::FileProperty, 37
- STOP1_CONDITION
 - utilities::DataFile, 26
- STOP2_CONDITION
 - utilities::DataFile, 26
- setAvailable
 - domain::Route, 82
- setFileName
 - utilities::DataFile, 23
- setKey
 - domain::Route, 83
 - domain::Town, 102
- setRoute
 - domain::Route, 84
- setTotalDistance
 - domain::Route, 85, 86
- setUp
 - test::CalculateDistanceTest, 14
 - test::DataFileTest, 27
 - test::FilterTest, 40
 - test::GraphTest, 54
 - test::PermutationTest, 64
 - test::RailSystemTest, 73
 - test::RouteTest, 89
 - test::ShortestRouteTest, 99
 - test::TownTest, 104
- shortestRoute
 - controller::RailSystem, 72
- showAllResults
 - frontend::Display, 33
- shutdownSW
 - software::main, 61
- software, 9
- software.main, 60
- software::main
 - main, 60
 - runSW, 61
 - shutdownSW, 61
 - startupSW, 62
- sr
 - test::ShortestRouteTest, 100
- startupSW
 - software::main, 62
- TOW_NO_EXIST
 - translate::PtBR, 67
 - translate::TranslateMessage, 108
- test, 10
- test.CalculateDistanceTest, 14
- test.DataFileTest, 26
- test.FilterTest, 39
- test.GraphTest, 54
- test.PermutationTest, 64
- test.RailSystemTest, 73
- test.RouteTest, 88
- test.ShortestRouteTest, 99
- test.TownTest, 104
- test::CalculateDistanceTest
 - cd, 15
 - setUp, 14
 - testCalculate, 14
 - testCalculateAll, 15
- test::DataFileTest
 - setUp, 27
 - testOpen, 27
 - testReadConditionStringInt, 27
 - testReadDoubleStringInt, 27
 - testReadIntegerStringInt, 28
 - testReadListString, 28
 - testReadListStringInt, 28
 - testReadLiteralOperandStringInt, 29
- test::FilterTest
 - filter, 40
 - setUp, 40
 - testFilterBy, 40
- test::GraphTest
 - graph, 60
 - setUp, 54
 - testAddRoute, 55
 - testAddTown, 55
 - testContainsTown, 56
 - testGetAllRoutePossible, 57
 - testGetAllRoutes, 57
 - testGetAllTowns, 57
 - testGetConnections, 58
 - testGetRoute, 58
 - testGraph, 59
 - testIsRouteAvailable, 59
 - testLoadGraphRoutesFromFile, 59
 - testcalculateDistance, 55
 - testclear, 56
- test::PermutationTest
 - setUp, 64
 - testGetAllLists, 64
- test::RailSystemTest
 - setUp, 73
 - testCalculateDistance, 73
 - testFilterByStops1, 73
 - testFilterByStops2, 74
 - testShortestRoute, 74
- test::RouteTest
 - route, 96
 - setUp, 89
 - testFormatKey, 89
 - testGetDistance, 89
 - testGetEndingTown, 90
 - testGetId, 90
 - testGetIdOfStop, 90

- testGetNumberOfNodes, 91
- testGetNumberOfStops, 91
- testGetStartingTown, 91
- testIsAvailable, 92
- testRoute, 92
- testRouteString, 92
- testRouteStringDouble, 93
- testSetAvailable, 93
- testSetId, 94
- testSetRoute, 94
- testSetTotalDistanceDouble, 94
- testSetTotalDistanceString, 95
- testTestCondition, 95
- test::ShortestRouteTest
 - setUp, 99
 - sr, 100
 - testGetShortestRoute, 99
 - testGetShortestRouteRoute, 100
- test::TownTest
 - setUp, 104
 - testFormatKey, 104
 - testGetId, 105
 - testSetId, 105
 - testTown, 105
 - town, 106
- testAddRoute
 - test::GraphTest, 55
- testAddTown
 - test::GraphTest, 55
- testCalculate
 - test::CalculateDistanceTest, 14
- testCalculateAll
 - test::CalculateDistanceTest, 15
- testCalculateDistance
 - test::RailSystemTest, 73
- testCondition
 - domain::Route, 86
- testContainsTown
 - test::GraphTest, 56
- testFilterBy
 - test::FilterTest, 40
- testFilterByStops1
 - test::RailSystemTest, 73
- testFilterByStops2
 - test::RailSystemTest, 74
- testFormatKey
 - test::RouteTest, 89
 - test::TownTest, 104
- testGetAllLists
 - test::PermutationTest, 64
- testGetAllRoutePossible
 - test::GraphTest, 57
- testGetAllRoutes
 - test::GraphTest, 57
- testGetAllTowns
 - test::GraphTest, 57
- testGetConnections
 - test::GraphTest, 58
- testGetDistance
 - test::RouteTest, 89
- testGetEndingTown
 - test::RouteTest, 90
- testGetId
 - test::RouteTest, 90
 - test::TownTest, 105
- testGetIdOfStop
 - test::RouteTest, 90
- testGetNumberOfNodes
 - test::RouteTest, 91
- testGetNumberOfStops
 - test::RouteTest, 91
- testGetRoute
 - test::GraphTest, 58
- testGetShortestRoute
 - test::ShortestRouteTest, 99
- testGetShortestRouteRoute
 - test::ShortestRouteTest, 100
- testGetStartingTown
 - test::RouteTest, 91
- testGraph
 - test::GraphTest, 59
- testIsAvailable
 - test::RouteTest, 92
- testIsRouteAvailable
 - test::GraphTest, 59
- testLoadGraphRoutesFromFile
 - test::GraphTest, 59
- testOpen
 - test::DataFileTest, 27
- testReadConditionStringInt
 - test::DataFileTest, 27
- testReadDoubleStringInt
 - test::DataFileTest, 27
- testReadIntegerStringInt
 - test::DataFileTest, 28
- testReadListString
 - test::DataFileTest, 28
- testReadListStringInt
 - test::DataFileTest, 28
- testReadLiteralOperandStringInt
 - test::DataFileTest, 29
- testRoute
 - test::RouteTest, 92
- testRouteString
 - test::RouteTest, 92
- testRouteStringDouble
 - test::RouteTest, 93
- testSetAvailable
 - test::RouteTest, 93
- testSetId
 - test::RouteTest, 94
 - test::TownTest, 105
- testSetRoute
 - test::RouteTest, 94
- testSetTotalDistanceDouble
 - test::RouteTest, 94

testSetTotalDistanceString
 test::RouteTest, 95
 testShortestRoute
 test::RailSystemTest, 74
 testTestCondition
 test::RouteTest, 95
 testTown
 test::TownTest, 105
 testcalculateDistance
 test::GraphTest, 55
 testclear
 test::GraphTest, 56
 totalDistance
 domain::Route, 88
 Town
 domain::Town, 101
 town
 test::TownTest, 106
 translate, 10
 translate.PtBR, 65
 translate.Translate, 106
 translate.TranslateMessage, 107
 translate::PtBR
 FILE_NOT_FOUND, 66, 67
 INCORRECT_PARAMETERS, 66, 67
 PROPERTY_NOT_FOUND, 66, 67
 ROUTE_NO_AVAILABLE, 66, 67
 TOW_NO_EXIST, 67
 translate::Translate
 MESSAGE, 107
 translate::TranslateMessage
 FILE_NOT_FOUND, 107
 INCORRECT_PARAMETERS, 108
 PROPERTY_NOT_FOUND, 108
 ROUTE_NO_AVAILABLE, 108
 TOW_NO_EXIST, 108

 utilities, 10
 utilities.DataFile, 16
 utilities.FileProperty, 35
 utilities.Permutation, 63
 utilities::DataFile
 checkFileExist, 17
 close, 17
 DISTANCE_CONDITION, 24
 FILE_TESTS_CASES, 25
 file_name, 25
 getFileNames, 17
 input, 25
 open, 18
 PATH_FILE, 25
 PATTERN_CONDITION, 25
 PATTERN_DOUBLE, 25
 PATTERN_INTEGER, 25
 PATTERN_LIST, 25
 PATTERN_OPERAND, 26
 prop, 26
 readCondition, 18
 readDouble, 19
 readInteger, 20
 readList, 21, 22
 readLiteralOperand, 23
 SEPARATOR, 26
 STOP1_CONDITION, 26
 STOP2_CONDITION, 26
 setFileName, 23
 utilities::FileProperty
 DISTANCE_ROUTES, 36
 FILTER_CONDITION, 36
 FILTER_ROUTES, 36
 FileProperty, 35
 GRAPH_ROUTES, 36
 getName, 35
 name, 36
 SHORTEST_ROUTES, 37
 utilities::Permutation
 getAllLists, 63