JavaScript is disabled on your browser.
Skip navigation links

- Overview
- Package
- Class
- Use
- Tree
- Deprecated
- Index
- Help

- Prev Class
- Next Class

- Frames
- No Frames

- All Classes

- Summary:
- Nested |
- Field |
- Constr |
- Method

- Detail:
- Field |
- Constr |
- Method

com.uem.dbconfig

Class DatabaseConnection

- java.lang.Object
- ♦ com.uem.dbconfig.DatabaseConnection

```
public final class DatabaseConnection
extends java.lang.Object
```

Provides all the methods needed to connect and query the relational database
Author:
zessin

- ♦

Field Summary

Fields

Modifier and Type	Field and Description
private java.sql.Connection	connection
static DatabaseConnection	databaseConnection
private static java.lang.String	MYSQL_DRIVER

private static java.lang.String	ORACLE_DRIVER
private static java.lang.String	POSTGRESQL_DRIVER
private java.sql.Statement	statement

Constructor Summary

Constructors

Modifier	Constructor and Description
private	DatabaseConnection() Initializes the correct JDBC Driver according to the RDBMS and starts the connection with the database

Method Summary

All Methods Static Methods Instance Methods Concrete Methods

Modifier and Type	Method and Description
void	closeStatement() Closes the statement used for previous queries
static DatabaseConnection	getConnection() Singleton method that provides the active instance of this class
private void	loadMysqlDriver() Initializes the driver for MySQL RDMBS
private void	loadOracleDriver() Initializes the driver for Oracle RDMBS
private void	loadPostgresqlDriver() Initializes the driver for PostgreSQL RDMBS
java.sql.ResultSet	query(java.lang.String query) Executes a query in the database

Methods inherited from class java.lang.Object

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

Field Detail

databaseConnection

```
public static DatabaseConnection databaseConnection
```

connection

```
private final java.sql.Connection connection
```

◊ **statement**

```
private java.sql.Statement statement
```

◊ **MYSQL_DRIVER**

```
private static final java.lang.String MYSQL_DRIVER
```

See Also:
Constant Field Values

◊ **ORACLE_DRIVER**

```
private static final java.lang.String ORACLE_DRIVER
```

See Also:
Constant Field Values

◊ **POSTGRESQL_DRIVER**

```
private static final java.lang.String POSTGRESQL_DRIVER
```

See Also:
Constant Field Values

♦ Constructor Detail

◊ **DatabaseConnection**

```
private DatabaseConnection()
```

Initializes the correct JDBC Driver according to the RDBMS and starts the connection with the database

♦ Method Detail

◊ **getConnection**

```
public static DatabaseConnection getConnection()
```

Singleton method that provides the active instance of this class
Returns:
The active instance of the DatabaseConnection class

◊ **query**

```
public java.sql.ResultSet query(java.lang.String query)
                         throws java.sql.SQLException
```

Executes a query in the database
Parameters:
`query` - The query to be executed
Returns:
The ResultSet obtained with the query (null if nothing was found)
Throws:
`java.sql.SQLException` - When the query couldn't be executed for some reason

◊ **closeStatement**

```
public void closeStatement()
                    throws java.sql.SQLException
```

Closes the statement used for previous queries
Throws:

`java.sql.SQLException` - When the statement couldn't be closed for some reason

◊ **loadMysqlDriver**

```
private void loadMysqlDriver()
```

Initializes the driver for MySQL RDMBS

◊ **loadOracleDriver**

```
private void loadOracleDriver()
```

Initializes the driver for Oracle RDMBS

◊ **loadPostgresqlDriver**

```
private void loadPostgresqlDriver()
```

Initializes the driver for PostgreSQL RDMBS

Skip navigation links

- Overview
- Package
- Class
- Use
- Tree
- Deprecated
- Index
- Help

- Prev Class
- Next Class

- Frames
- No Frames

- All Classes

- Summary:
- Nested |
- Field |
- Constr |
- Method

- Detail:
- Field |
- Constr |
- Method

JavaScript is disabled on your browser.
Skip navigation links

- Overview
- Package
- Class
- Use
- Tree
- Deprecated
- Index
- Help

- Prev Class
- Next Class

- Frames
- No Frames

- All Classes

- Summary:
- Nested |
- Enum Constants |
- Field |
- Method

- Detail:
- Enum Constants |
- Field |
- Method

com.uem.dbconfig

Enum DatabaseType

- java.lang.Object
- ♦ java.lang.Enum<DatabaseType>
  - ♦ ◊ com.uem.dbconfig.DatabaseType

- All Implemented Interfaces:
  java.io.Serializable, java.lang.Comparable<DatabaseType>

```
public enum DatabaseType
extends java.lang.Enum<DatabaseType>
```

Provides all the RDBMSs which are supported by the application
Author:
zessin

- ♦

Enum Constant Summary

Enum Constants

Enum Constant and Description
MYSQL
ORACLE

POSTGRESQL

Method Summary

All Methods Static Methods Concrete Methods

Modifier and Type	Method and Description
static DatabaseType	getDatabaseTypeByProperty(java.lang.String database Returns the correct DatabaseType according to the property in the properties
static DatabaseType	valueOf(java.lang.String name) Returns the enum constant of this type with the specified name.
static DatabaseType[]	values() Returns an array containing the constants of this enum type, in the order they

Methods inherited from class java.lang.Enum

clone, compareTo, equals, finalize, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

Methods inherited from class java.lang.Object

getClass, notify, notifyAll, wait, wait, wait

Enum Constant Detail

MYSQL

```
public static final DatabaseType MYSQL
```

ORACLE

```
public static final DatabaseType ORACLE
```

POSTGRESQL

```
public static final DatabaseType POSTGRESQL
```

Method Detail

values

```
public static DatabaseType[] values()
```

Returns an array containing the constants of this enum type, in the order they are declared. This method may be used to iterate over the constants as follows:

```
for (DatabaseType c : DatabaseType.values())
    System.out.println(c);
```

Returns:
an array containing the constants of this enum type, in the order they are declared

◊ valueOf

```
public static DatabaseType valueOf(java.lang.String name)
```

Returns the enum constant of this type with the specified name. The string must match *exactly* an identifier used to declare an enum constant in this type. (Extraneous whitespace characters are not permitted.)
Parameters:
`name` - the name of the enum constant to be returned.
Returns:
the enum constant with the specified name
Throws:
`java.lang.IllegalArgumentException` - if this enum type has no constant with the specified name
`java.lang.NullPointerException` - if the argument is null

◊ getDatabaseTypeByProperty

```
public static DatabaseType getDatabaseTypeByProperty(java.lang.String datab
```

Returns the correct DatabaseType according to the property in the properties file
Parameters:
`databaseTypeProperty` - The value of the property in the file
Returns:
The correct DatabaseType for the property (null if not found)

Skip navigation links

- Overview
- Package
- Class
- Use
- Tree
- Deprecated
- Index
- Help

- Prev Class
- Next Class

- Frames
- No Frames

- All Classes

- Summary:
- Nested |
- Enum Constants |
- Field |
- Method

- Detail:
- Enum Constants |
- Field |
- Method

JavaScript is disabled on your browser.
Skip navigation links

- Overview
- Package
- Class
- Use
- Tree
- Deprecated
- Index
- Help

- Prev Class
- Next Class

- Frames
- No Frames

- All Classes

- Summary:
- Nested |
- Field |
- Constr |
- Method

- Detail:
- Field |
- Constr |
- Method

com.uem.dbstructure

Class Column

- java.lang.Object
- ♦ com.uem.dbstructure.Column

```
public class Column
extends java.lang.Object
```

Represents a column from a table in the relational database
Author:
zessin

♦

Field Summary

Fields

Modifier and Type	Field and Description
private java.lang.String	name
private Table	table

Constructor Summary

Constructors

Constructor and Description
`Column(Table table, java.lang.String name)` Initializes the Column with a name, associating it with a Table

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method and Description
`boolean`	`equals(java.lang.Object other)`
`java.lang.String`	`getName()`
`Table`	`getTable()`
`void`	`setName(java.lang.String name)`
`void`	`setTable(Table table)`
`java.lang.String`	`toString()`

Methods inherited from class java.lang.Object

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

Field Detail

table

```
private Table table
```

name

```
private java.lang.String name
```

Constructor Detail

Column

```
public Column(Table table,
              java.lang.String name)
```

Initializes the Column with a name, associating it with a Table
Parameters:
`table` - The Table which the Column belongs to
`name` - The name of the Column

Method Detail

◊ getTable

```
public Table getTable()
```

◊ setTable

```
public void setTable(Table table)
```

◊ getName

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

◊ setName

```
public void setName(java.lang.String name)
```

◊ equals

```
public boolean equals(java.lang.Object other)
```

Overrides:
`equals` in class `java.lang.Object`

◊ toString

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

Overrides:
`toString` in class `java.lang.Object`

Skip navigation links

- Overview
- Package
- Class
- Use
- Tree
- Deprecated
- Index
- Help

- Prev Class
- Next Class

- Frames
- No Frames

- All Classes

- Summary:
- Nested |
- Field |
- Constr |
- Method

- Detail:
- Field |
- Constr |
- Method

JavaScript is disabled on your browser.
Skip navigation links

- Overview
- Package
- Class
- Use
- Tree
- Deprecated
- Index
- Help

- Prev Class
- Next Class

- Frames
- No Frames

- All Classes

- Summary:
- Nested |
- Field |
- Constr |
- Method

- Detail:
- Field |
- Constr |
- Method

com.uem.dbstructure

Class Constraint

- java.lang.Object
- ♦ com.uem.dbstructure.Constraint

```
public class Constraint
extends java.lang.Object
```

Represents a constraint related to columns in the relational database
Author:
zessin

♦

Field Summary

Fields

Modifier and Type	Field and Description
private Column	column
private java.lang.String	name
private Column	referencedColumn

private Table	referencedTable
private Table	table
private ConstraintType	type

Constructor Summary

Constructors

Constructor and Description
Constraint(java.lang.String name, Table table, Column column, Table referencedTable, Column referencedColumn, ConstraintType type) Initializes the Constraint with the needed information

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method and Description
boolean	equals(java.lang.Object other)
Column	getColumn()
java.lang.String	getName()
Column	getReferencedColumn()
Table	getReferencedTable()
Table	getTable()
ConstraintType	getType()
void	setColumn(Column column)
void	setName(java.lang.String name)
void	setReferencedColumn(Column referencedColumn)
void	setReferencedTable(Table referencedTable)
void	setTable(Table table)
void	setType(ConstraintType type)
java.lang.String	toString()

Methods inherited from class java.lang.Object

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

♦ Field Detail

◊ name

```
private java.lang.String name
```

◊ table

```
private Table table
```

◊ column

```
private Column column
```

◊ referencedTable

```
private Table referencedTable
```

◊ referencedColumn

```
private Column referencedColumn
```

◊ type

```
private ConstraintType type
```

♦ Constructor Detail

◊ Constraint

```
public Constraint(java.lang.String name,
                  Table table,
                  Column column,
                  Table referencedTable,
                  Column referencedColumn,
                  ConstraintType type)
```

Initializes the Constraint with the needed information
Parameters:
`name` - The name of the constraint
`table` - The Table which the Constraint belongs to
`column` - The Column which the Constraint belongs to
`referencedTable` - The referenced Table, when the Constraint is a FOREIGN_KEY
`referencedColumn` - The referenced Column, when the Constraint is a FOREIGN_KEY
`type` - The type of the Constraint

♦ Method Detail

◊ getName

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

◊ setName

```
public void setName(java.lang.String name)
```

◊ getTable

```
public Table getTable()
```

◊ setTable

```
public void setTable(Table table)
```

- ◊ **getColumn**

```
public Column getColumn()
```

◊ **setColumn**

```
public void setColumn(Column column)
```

◊ **getReferencedTable**

```
public Table getReferencedTable()
```

◊ **setReferencedTable**

```
public void setReferencedTable(Table referencedTable)
```

◊ **getReferencedColumn**

```
public Column getReferencedColumn()
```

◊ **setReferencedColumn**

```
public void setReferencedColumn(Column referencedColumn)
```

◊ **getType**

```
public ConstraintType getType()
```

◊ **setType**

```
public void setType(ConstraintType type)
```

◊ **equals**

```
public boolean equals(java.lang.Object other)
```

Overrides:
`equals` in class `java.lang.Object`

◊ **toString**

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

Overrides:
`toString` in class `java.lang.Object`

Skip navigation links

- Overview
- Package
- Class
- Use
- Tree
- Deprecated
- Index
- Help

- Prev Class
- Next Class

- Frames
- No Frames

- All Classes

- Summary:
- Nested |
- Field |
- Constr |
- Method

- Detail:
- Field |
- Constr |
- Method

JavaScript is disabled on your browser.
Skip navigation links

- Overview
- Package
- Class
- Use
- Tree
- Deprecated
- Index
- Help

- Prev Class
- Next Class

- Frames
- No Frames

- All Classes

- Summary:
- Nested |
- Enum Constants |
- Field |
- Method

- Detail:
- Enum Constants |
- Field |
- Method

com.uem.dbstructure

Enum ConstraintType

- java.lang.Object
- ♦ java.lang.Enum<ConstraintType>
  - ♦ ◊ com.uem.dbstructure.ConstraintType

- All Implemented Interfaces:
  java.io.Serializable, java.lang.Comparable<ConstraintType>

```
public enum ConstraintType
extends java.lang.Enum<ConstraintType>
```

Provides all the constraint types which are necessary for the application
Author:
zessin

- ♦

Enum Constant Summary

Enum Constants

Enum Constant and Description

FOREIGN_KEY

PRIMARY_KEY

UNIQUE_KEY

Field Summary

Fields

Modifier and Type	Field and Description
private java.lang.String	name

Method Summary

All Methods Static Methods Instance Methods Concrete Methods

Modifier and Type	Method and Description
static ConstraintType	getConstraintTypeByName(java.lang.String name) Provides a ConstraintType based on its name
java.lang.String	getName()
boolean	isForeignKey() Tells if the ConstraintType is a foreign key
void	setName(java.lang.String name)
static ConstraintType	valueOf(java.lang.String name) Returns the enum constant of this type with the specified name.
static ConstraintType[]	values() Returns an array containing the constants of this enum type, in the order they are declared.

Methods inherited from class java.lang.Enum

clone, compareTo, equals, finalize, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

Methods inherited from class java.lang.Object

getClass, notify, notifyAll, wait, wait, wait

Enum Constant Detail

PRIMARY_KEY

public static final ConstraintType PRIMARY_KEY

FOREIGN_KEY

public static final ConstraintType FOREIGN_KEY

UNIQUE_KEY

public static final ConstraintType UNIQUE_KEY

♦ Field Detail

◊ name

```
private java.lang.String name
```

♦ Method Detail

◊ values

```
public static ConstraintType[] values()
```

Returns an array containing the constants of this enum type, in the order they are declared. This method may be used to iterate over the constants as follows:

```
for (ConstraintType c : ConstraintType.values())
    System.out.println(c);
```

Returns:
an array containing the constants of this enum type, in the order they are declared

◊ valueOf

```
public static ConstraintType valueOf(java.lang.String name)
```

Returns the enum constant of this type with the specified name. The string must match *exactly* an identifier used to declare an enum constant in this type. (Extraneous whitespace characters are not permitted.)
Parameters:
`name` - the name of the enum constant to be returned.
Returns:
the enum constant with the specified name
Throws:
`java.lang.IllegalArgumentException` - if this enum type has no constant with the specified name
`java.lang.NullPointerException` - if the argument is null

◊ getName

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

◊ setName

```
public void setName(java.lang.String name)
```

◊ isForeignKey

```
public boolean isForeignKey()
```

Tells if the ConstraintType is a foreign key
Returns:
true when the ConstraintType is a foreign key, false otherwise

◊ getConstraintTypeByName

```
public static ConstraintType getConstraintTypeByName(java.lang.String name)
```

Provides a ConstraintType based on its name
Parameters:
`name` - The name of the ConstraintType
Returns:
The correct ConstraintType for the name (null if not found)

Skip navigation links

- Overview
- Package
- Class
- Use
- Tree
- Deprecated
- Index
- Help

- Prev Class
- Next Class

- Frames
- No Frames

- All Classes

- Summary:
- Nested |
- Enum Constants |
- Field |
- Method

- Detail:
- Enum Constants |
- Field |
- Method

JavaScript is disabled on your browser.
Skip navigation links

- Overview
- Package
- Class
- Use
- Tree
- Deprecated
- Index
- Help

- Prev Class
- Next Class

- Frames
- No Frames

- All Classes

- Summary:
- Nested |
- Field |
- Constr |
- Method

- Detail:
- Field |
- Constr |
- Method

com.uem.dbstructure

Class DatabaseInfo

- java.lang.Object
- ♦ com.uem.dbstructure.DatabaseInfo

```
public class DatabaseInfo
extends java.lang.Object
```

Provides all the methods needed to load the metadata from a relational database
Author:
zessin

Field Summary

Fields

Modifier and Type	Field and Description
private java.util.List<Column>	columns
private java.util.List<Constraint>	constraints
private DatabaseConnection	databaseConnection

private java.sql.ResultSet	resultSet
private java.lang.String	schema
private java.util.List<Table>	tables

Constructor Summary

Constructors

Constructor and Description
DatabaseInfo() Initializes all the attributes and the schema based on the properties file

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method and
private Column	findColumnByTableAndColumnNames(java.la java.lang.String columnName) Finds a Column amongst all the loaded ones by its name a
private java.util.List<Constraint>	findForeignKeysByRelationshipTableName(Finds all the constraints related to a "many to many" relati
private Table	findTableByTableName(java.lang.String t Finds a Table amongst all the loaded ones by its name
java.util.List<Column>	getColumns()
java.util.List<Constraint>	getConstraints()
DatabaseConnection	getDatabaseConnection()
java.sql.ResultSet	getResultSet()
java.util.List<Table>	getTables()
private void	loadAllColumns() Queries and organizes all the columns metadata needed fo
private void	loadAllConstraints() Queries and organizes all the constraints metadata needed
private void	loadAllTables() Queries and organizes all the tables metadata needed for t
void	loadDatabaseInformation() Queries and organizes all the metadata needed for the appl
private void	loadRelationshipTables() Finds all the tables which are exclusively used for a "many
private void	loadTableColumns(Table table) Queries and organizes all the table's columns metadata nee
private void	loadTableConstraints(Table table)

	Queries and organizes all the table's constraints metadata
void	printLoadedData() Simple method for printing all the loaded metadata in Con
void	setResultSet(java.sql.ResultSet resultS

- ◊ **Methods inherited from class java.lang.Object**

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

- ♦ **Field Detail**

  - ◊ **tables**

    ```
    private final java.util.List<Table> tables
    ```

  - ◊ **columns**

    ```
    private final java.util.List<Column> columns
    ```

  - ◊ **constraints**

    ```
    private final java.util.List<Constraint> constraints
    ```

  - ◊ **databaseConnection**

    ```
    private final DatabaseConnection databaseConnection
    ```

  - ◊ **schema**

    ```
    private final java.lang.String schema
    ```

  - ◊ **resultSet**

    ```
    private java.sql.ResultSet resultSet
    ```

- ♦ **Constructor Detail**

  - ◊ **DatabaseInfo**

    ```
    public DatabaseInfo()
    ```

    Initializes all the attributes and the schema based on the properties file

- ♦ **Method Detail**

  - ◊ **loadDatabaseInformation**

    ```
    public void loadDatabaseInformation()
                                throws java.sql.SQLException
    ```

    Queries and organizes all the metadata needed for the application
    Throws:
    `java.sql.SQLException` - When the metadata couldn't be queried for some reason

  - ◊ **loadAllTables**

    ```
    private void loadAllTables()
                        throws java.sql.SQLException
    ```

Queries and organizes all the tables metadata needed for the application
Throws:
`java.sql.SQLException` - When the tables metadata couldn't be queried for some reason

◊ loadAllColumns

```
private void loadAllColumns()
                     throws java.sql.SQLException
```

Queries and organizes all the columns metadata needed for the application
Throws:
`java.sql.SQLException` - When the columns metadata couldn't be queried for some reason

◊ loadTableColumns

```
private void loadTableColumns(Table table)
                       throws java.sql.SQLException
```

Queries and organizes all the table's columns metadata needed for the application
Parameters:
`table` - The Table whose columns will be queried
Throws:
`java.sql.SQLException` - When the table's columns metadata couldn't be queried for some reason

◊ loadAllConstraints

```
private void loadAllConstraints()
                         throws java.sql.SQLException
```

Queries and organizes all the constraints metadata needed for the application
Throws:
`java.sql.SQLException` - When the constraints metadata couldn't be queried for some reason

◊ loadTableConstraints

```
private void loadTableConstraints(Table table)
                           throws java.sql.SQLException
```

Queries and organizes all the table's constraints metadata needed for the application
Parameters:
`table` - The Table whose constraints will be queried
Throws:
`java.sql.SQLException` - When the table's constraints metadata couldn't be queried for some reason

◊ loadRelationshipTables

```
private void loadRelationshipTables()
                             throws java.sql.SQLException
```

Finds all the tables which are exclusively used for a "many to many" relationship and mark them as so
Throws:
`java.sql.SQLException` - When the tables metadata couldn't be queried for some reason

◊ findTableByTableName

```
private Table findTableByTableName(java.lang.String tableName)
```

Finds a Table amongst all the loaded ones by its name
Parameters:
`tableName` - The name of the Table which will be searched
Returns:
The Table found (null if not found)

◊ findColumnByTableAndColumnNames

```
private Column findColumnByTableAndColumnNames(java.lang.String tableName,
                                               java.lang.String columnName)
```

Finds a Column amongst all the loaded ones by its name and its Table's name
Parameters:
`tableName` - The name of the Table which will be used in the search
`columnName` - The name of the Column which will be used in the search
Returns:
The Column found (null if not found)

◊ findForeignKeysByRelationshipTableName

```
private java.util.List<Constraint> findForeignKeysByRelationshipTableName(j
```

Finds all the constraints related to a "many to many" relationship table
Parameters:
`relationshipTableName` - The name of the "many to many" relationship table
Returns:
All the constraints found

◊ printLoadedData

```
public void printLoadedData()
```

Simple method for printing all the loaded metadata in Console

◊ getResultSet

```
public java.sql.ResultSet getResultSet()
```

◊ setResultSet

```
public void setResultSet(java.sql.ResultSet resultSet)
```

◊ getTables

```
public java.util.List<Table> getTables()
```

◊ getColumns

```
public java.util.List<Column> getColumns()
```

◊ getConstraints

```
public java.util.List<Constraint> getConstraints()
```

◊ getDatabaseConnection

```
public DatabaseConnection getDatabaseConnection()
```

Skip navigation links

- Overview
- Package
- Class
- Use
- Tree
- Deprecated
- Index
- Help

- Prev Class
- Next Class

- Frames
- No Frames

- All Classes

- Summary:
- Nested |
- Field |
- Constr |
- Method

- Detail:
- Field |
- Constr |
- Method

JavaScript is disabled on your browser.
Skip navigation links

- Overview
- Package
- Class
- Use
- Tree
- Deprecated
- Index
- Help

- Prev Class
- Next Class

- Frames
- No Frames

- All Classes

- Summary:
- Nested |
- Field |
- Constr |
- Method

- Detail:
- Field |
- Constr |
- Method

com.uem.dbstructure

Class Table

- java.lang.Object
- ♦ com.uem.dbstructure.Table

```
public class Table
extends java.lang.Object
```

Represents a table in the relational database
Author:
zessin

♦

Field Summary

Fields

Modifier and Type	Field and Description
private java.lang.String	name
private java.lang.Boolean	relationshipTable

Constructor Summary

Constructors

Constructor and Description
`Table(java.lang.String name)` Initializes the Table with a name

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method and Description
`boolean`	`equals(java.lang.Object other)`
`java.lang.String`	`getName()`
`java.lang.Boolean`	`isRelationshipTable()`
`void`	`setName(java.lang.String name)`
`void`	`setRelationshipTable(java.lang.Boolean relationsh`
`java.lang.String`	`toString()`

Methods inherited from class java.lang.Object

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

Field Detail

name

```
private java.lang.String name
```

relationshipTable

```
private java.lang.Boolean relationshipTable
```

Constructor Detail

Table

```
public Table(java.lang.String name)
```

Initializes the Table with a name
Parameters:
`name` - The name of the Table

Method Detail

◊ getName

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

◊ setName

```
public void setName(java.lang.String name)
```

◊ isRelationshipTable

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

◊ setRelationshipTable

```
public void setRelationshipTable(java.lang.Boolean relationshipTable)
```

◊ equals

```
public boolean equals(java.lang.Object other)
```

Overrides:
`equals` in class `java.lang.Object`

◊ toString

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

Overrides:
`toString` in class `java.lang.Object`

Skip navigation links

- Overview
- Package
- Class
- Use
- Tree
- Deprecated
- Index
- Help

- Prev Class
- Next Class

- Frames
- No Frames

- All Classes

- Summary:
- Nested |
- Field |
- Constr |
- Method

- Detail:
- Field |
- Constr |
- Method

JavaScript is disabled on your browser.
Skip navigation links

- Overview
- Package
- Class
- Use
- Tree
- Deprecated
- Index
- Help

- Prev Class
- Next Class

- Frames
- No Frames

- All Classes

- Summary:
- Nested |
- Field |
- Constr |
- Method

- Detail:
- Field |
- Constr |
- Method

com.uem.graph

Class Edge

- java.lang.Object
- ♦ com.uem.graph.Edge

```
public class Edge
extends java.lang.Object
```

Represents and Edge in the Graph
Author:
zessin

- ♦

Field Summary

Fields

Modifier and Type	Field and Description
private java.lang.String	name
private Vertex	v1
private Vertex	v2

Constructor Summary

Constructors

Constructor and Description
`Edge(java.lang.String name, Vertex v1, Vertex v2)` Initializes the Edge with a name and two vertices

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method and Description
`boolean`	`equals(java.lang.Object other)`
`java.lang.String`	`getName()`
`Vertex`	`getV1()`
`Vertex`	`getV2()`
`void`	`setName(java.lang.String name)`
`void`	`setV1(Vertex v1)`
`void`	`setV2(Vertex v2)`
`java.lang.String`	`toString()`

Methods inherited from class java.lang.Object

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

Field Detail

name

```
private java.lang.String name
```

v1

```
private Vertex v1
```

v2

```
private Vertex v2
```

Constructor Detail

Edge

```
public Edge(java.lang.String name,
            Vertex v1,
            Vertex v2)
```

Initializes the Edge with a name and two vertices

Parameters:
`name` - The Edge's name
`v1` - The source Vertex
`v2` - The destination Vertex

Method Detail

getName

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

setName

```
public void setName(java.lang.String name)
```

getV1

```
public Vertex getV1()
```

setV1

```
public void setV1(Vertex v1)
```

getV2

```
public Vertex getV2()
```

setV2

```
public void setV2(Vertex v2)
```

equals

```
public boolean equals(java.lang.Object other)
```

Overrides:
`equals` in class `java.lang.Object`

toString

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

Overrides:
`toString` in class `java.lang.Object`

Skip navigation links

- Overview
- Package
- Class
- Use
- Tree
- Deprecated
- Index
- Help

- Prev Class
- Next Class

- Frames
- No Frames

- All Classes

- Summary:
- Nested |
- Field |

- Constr |
- Method

- Detail:
- Field |
- Constr |
- Method

JavaScript is disabled on your browser.
Skip navigation links

- Overview
- Package
- Class
- Use
- Tree
- Deprecated
- Index
- Help

- Prev Class
- Next Class

- Frames
- No Frames

- All Classes

- Summary:
- Nested |
- Field |
- Constr |
- Method

- Detail:
- Field |
- Constr |
- Method

com.uem.graph

Class Graph

- java.lang.Object
- ♦ com.uem.graph.Graph

```
public class Graph
extends java.lang.Object
```

Represents the graph model for the relational model obtained with the metadata from the RDBMS
Author:
zessin

- ♦

Field Summary

Fields

Modifier and Type	Field and Description
`private java.util.Map<Vertex,java.util.List<Vertex>>`	`adjacencyList`
`private boolean`	`directed`
`private java.util.List<Edge>`	`edges`

private java.util.Map<Pair,Edge>	edgesFromVertices
private java.util.List<Vertex>	vertices

Constructor Summary

Constructors

Constructor and Description
Graph(boolean directed) Initializes the Graph telling if it's a directed one or not
Graph(DatabaseInfo databaseInfo, boolean directed) Initializes the Graph with some existing relational database information, previously obtained and organized, and telling if it's a directed one or not

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	
private void	addEdge(Edge edge) Adds an Edge in the graph
private void	addEdgeFromForeignKey Adds a new Edge based on an e
private void	addVertex(Vertex verte Adds a Vertex in the graph
private void	addVertexFromTable(Tab Adds a new Vertex based on an
private void	generateGraphFromDatab Generates a graph model based
java.util.Map<Vertex,java.util.List<Vertex>>	getAdjacencyList()
java.util.List<Edge>	getEdges()
java.util.Map<Pair,Edge>	getEdgesFromVertices()
private Vertex	getVertexByName(java. Finds a Vertex by its name
java.util.List<Vertex>	getVertices()
boolean	isDirected()
void	setAdjacencyList(java
void	setDirected(boolean di
void	setEdges(java.util.Lis
void	setEdgesFromVertices(
void	setVertices(java.util

◊ Methods inherited from class java.lang.Object

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

♦ Field Detail

◊ vertices

```
private java.util.List<Vertex> vertices
```

◊ edges

```
private java.util.List<Edge> edges
```

◊ adjacencyList

```
private java.util.Map<Vertex,java.util.List<Vertex>> adjacencyList
```

◊ edgesFromVertices

```
private java.util.Map<Pair,Edge> edgesFromVertices
```

◊ directed

```
private boolean directed
```

♦ Constructor Detail

◊ Graph

```
public Graph(boolean directed)
```

Initializes the Graph telling if it's a directed one or not
Parameters:
`directed` - Tells whether the Graph is directed or not

◊ Graph

```
public Graph(DatabaseInfo databaseInfo,
             boolean directed)
```

Initializes the Graph with some existing relational database information, previously obtained and organized, and telling if it's a directed one or not
Parameters:
`databaseInfo` - The relational database information
`directed` - Tells whether the Graph is directed or not

♦ Method Detail

◊ generateGraphFromDatabaseInfo

```
private void generateGraphFromDatabaseInfo(DatabaseInfo databaseInfo)
```

Generates a graph model based on the database information previously obtained and organized
Parameters:
`databaseInfo` - The relational database information

◊ addVertexFromTable

```
private void addVertexFromTable(Table table)
```

Adds a new Vertex based on an existing Table

Parameters:
`table` - The Table which will become a Vertex in the Graph

◊ **addEdgeFromForeignKey**

```
private void addEdgeFromForeignKey(Constraint foreignKey)
```

Adds a new Edge based on an existing foreign key Constraint
Parameters:
`foreignKey` - The foreign key Constraint which will become an Edge in the Graph

◊ **addVertex**

```
private void addVertex(Vertex vertex)
```

Adds a Vertex in the graph
Parameters:
`vertex` - The Vertex to be added in the Graph

◊ **addEdge**

```
private void addEdge(Edge edge)
```

Adds an Edge in the graph
Parameters:
`edge` - The Edge to be added in the Graph

◊ **getVertexByName**

```
private Vertex getVertexByName(java.lang.String vertexName)
```

Finds a Vertex by its name
Parameters:
`vertexName` - The name of the Vertex to be searched
Returns:
The Vertex found (null if not found)

◊ **getVertices**

```
public java.util.List<Vertex> getVertices()
```

◊ **setVertices**

```
public void setVertices(java.util.List<Vertex> vertices)
```

◊ **getEdges**

```
public java.util.List<Edge> getEdges()
```

◊ **setEdges**

```
public void setEdges(java.util.List<Edge> edges)
```

◊ **getAdjacencyList**

```
public java.util.Map<Vertex,java.util.List<Vertex>> getAdjacencyList()
```

◊ **setAdjacencyList**

```
public void setAdjacencyList(java.util.Map<Vertex,java.util.List<Vertex>> a
```

◊ **getEdgesFromVertices**

```
public java.util.Map<Pair,Edge> getEdgesFromVertices()
```

◊ **setEdgesFromVertices**

```
public void setEdgesFromVertices(java.util.Map<Pair,Edge> edgesFromVertices
```

◊ **isDirected**

```
public boolean isDirected()
```

◊ setDirected

```
public void setDirected(boolean directed)
```

Skip navigation links

- Overview
- Package
- Class
- Use
- Tree
- Deprecated
- Index
- Help

- Prev Class
- Next Class

- Frames
- No Frames

- All Classes

- Summary:
- Nested |
- Field |
- Constr |
- Method

- Detail:
- Field |
- Constr |
- Method

JavaScript is disabled on your browser.
Skip navigation links

- Overview
- Package
- Class
- Use
- Tree
- Deprecated
- Index
- Help

- Prev Class
- Next Class

- Frames
- No Frames

- All Classes

- Summary:
- Nested |
- Field |
- Constr |
- Method

- Detail:
- Field |
- Constr |
- Method

com.uem.graph

Class Pair

- java.lang.Object
- ♦ com.uem.graph.Pair

```
public class Pair
extends java.lang.Object
```

Represents a pair of vertices in the Graph. Used in the Graph class to easily find the edges between two vertices
Author:
zessin

- ♦

Field Summary

Fields

Modifier and Type	Field and Description
private Vertex	v1
private Vertex	v2

Constructor Summary

Constructors

Constructor and Description
`Pair(Vertex v1, Vertex v2)` Initializes the Pair with two vertices

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method and Description
`boolean`	`equals(java.lang.Object other)`
`Vertex`	`getV1()`
`Vertex`	`getV2()`
`void`	`setV1(Vertex v1)`
`void`	`setV2(Vertex v2)`
`java.lang.String`	`toString()`

Methods inherited from class java.lang.Object

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

Field Detail

v1

```
private Vertex v1
```

v2

```
private Vertex v2
```

Constructor Detail

Pair

```
public Pair(Vertex v1,
            Vertex v2)
```

Initializes the Pair with two vertices
Parameters:
`v1` - The source Vertex
`v2` - The destination Vertex

Method Detail

◊ getV1

```
public Vertex getV1()
```

◊ setV1

```
public void setV1(Vertex v1)
```

◊ getV2

```
public Vertex getV2()
```

◊ setV2

```
public void setV2(Vertex v2)
```

◊ equals

```
public boolean equals(java.lang.Object other)
```

Overrides:
`equals` in class `java.lang.Object`

◊ toString

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

Overrides:
`toString` in class `java.lang.Object`

Skip navigation links

- Overview
- Package
- Class
- Use
- Tree
- Deprecated
- Index
- Help

- Prev Class
- Next Class

- Frames
- No Frames

- All Classes

- Summary:
- Nested |
- Field |
- Constr |
- Method

- Detail:
- Field |
- Constr |
- Method

JavaScript is disabled on your browser.
Skip navigation links

- Overview
- Package
- Class
- Use
- Tree
- Deprecated
- Index
- Help

- Prev Class
- Next Class

- Frames
- No Frames

- All Classes

- Summary:
- Nested |
- Field |
- Constr |
- Method

- Detail:
- Field |
- Constr |
- Method

com.uem.graph

Class Vertex

- java.lang.Object
- ♦ com.uem.graph.Vertex

```
public class Vertex
extends java.lang.Object
```

Represents a Vertex in the Graph
Author:
zessin

♦

Field Summary

Fields

Modifier and Type	Field and Description
private java.lang.Long	degree
private java.lang.String	name

Constructor Summary

Constructors

Constructor and Description
`Vertex(java.lang.String name)` Initializes the Vertex with a name
`Vertex(Table table)` Initializes the Vertex using the Table's name

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method and Description
`boolean`	`equals(java.lang.Object other)`
`java.lang.Long`	`getDegree()`
`java.lang.String`	`getName()`
`void`	`increaseDegree()`
`boolean`	`isDegreePositive()`
`void`	`setDegree(java.lang.Long degree)`
`void`	`setName(java.lang.String name)`
`java.lang.String`	`toString()`

Methods inherited from class java.lang.Object

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

Field Detail

name

```
private java.lang.String name
```

degree

```
private java.lang.Long degree
```

Constructor Detail

Vertex

```
public Vertex(java.lang.String name)
```

Initializes the Vertex with a name
Parameters:
`name` - The name of the Vertex

◊ **Vertex**

```
public Vertex(Table table)
```

Initializes the Vertex using the Table's name
Parameters:
`table` - The Table which the Vertex will represent in the Graph

♦ Method Detail

◊ **getName**

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

◊ **setName**

```
public void setName(java.lang.String name)
```

◊ **getDegree**

```
public java.lang.Long getDegree()
```

◊ **setDegree**

```
public void setDegree(java.lang.Long degree)
```

◊ **increaseDegree**

```
public void increaseDegree()
```

◊ **isDegreePositive**

```
public boolean isDegreePositive()
```

◊ **equals**

```
public boolean equals(java.lang.Object other)
```

Overrides:
`equals` in class `java.lang.Object`

◊ **toString**

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

Overrides:
`toString` in class `java.lang.Object`

Skip navigation links

- Overview
- Package
- Class
- Use
- Tree
- Deprecated
- Index
- Help

- Prev Class
- Next Class

- Frames
- No Frames

- All Classes

- Summary:
- Nested |
- Field |
- Constr |
- Method

- Detail:
- Field |
- Constr |
- Method

JavaScript is disabled on your browser.
Skip navigation links

- Overview
- Package
- Class
- Use
- Tree
- Deprecated
- Index
- Help

- Prev Class
- Next Class

- Frames
- No Frames

- All Classes

- Summary:
- Nested |
- Field |
- Constr |
- Method

- Detail:
- Field |
- Constr |
- Method

com.uem.graphviz

Class Graphviz

- java.lang.Object
- ♦ com.uem.graphviz.Graphviz

```
public class Graphviz
extends java.lang.Object
```

Purpose: Graphviz Java API
Description:
With this Java class you can simply call dot from your Java programs.
Example usage:

```
Graphviz gv = new Graphviz();
gv.addln(gv.start_graph());
gv.addln("A - B;");
gv.addln("A - C;");
gv.addln(gv.end_graph());
System.out.println(gv.getDotSource());

String type = "gif";
String representationType = "dot";
File out = new File("out." + type); // out.gif in this example
gv.writeGraphToFile(gv.getGraph(gv.getDotSource(), type, representationType), out
```

Version:
v0.6, 2013/11/28 (November) -- Patch of Olivier Duplouy is added. Now you can specify the representation type of your graph: dot, neato, fdp, sfdp, twopi, circo, v0.5.1, 2013/03/18 (March) -- Patch of Juan Hoyos (Mac support), v0.5, 2012/04/24 (April) -- Patch of Abdur

Rahman (OS detection + start subgraph + read config file), v0.4, 2011/02/05 (February) -- Patch of Keheliya Gallaba is added. Now you can specify the type of the output file: gif, dot, fig, pdf, ps, svg, png, etc., v0.3, 2010/11/29 (November) -- Windows support + ability to read the graph from a text file, v0.2, 2010/07/22 (July) -- bug fix, v0.1, 2003/12/04 (December) -- first release

Author:

Laszlo Szathmary (jabba.laci@gmail.com)

Field Summary

Fields

Modifier and Type	Field and Description
`private int`	`currentDpiPos` Define the index in the image size array.
`private static java.lang.String`	`DOT` Where is your dot program located? It will be called externally.
`private int[]`	`dpiSizes` The image size in dpi.
`private java.lang.StringBuilder`	`graph` The source of the graph written in dot language.
`private static java.lang.String`	`TEMP_DIR` The dir.

Constructor Summary

Constructors

Constructor and Description
`Graphviz()` Constructor: creates a new Graphviz object that will contain a graph.

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method and Description
`void`	`add(java.lang.String line)` Adds a string to the graph's source (without newline).
`void`	`addln()` Adds a newline to the graph's source.
`void`	`addln(java.lang.String line)` Adds a string to the graph's source (with newline).
`void`	`clearGraph()`

void	decreaseDpi() Decrease the image size (dpi).
java.lang.String	end_subgraph() Returns a string that is used to end a graph.
java.lang.String	endGraph() Returns a string that is used to end a graph.
private byte[]	get_img_stream(java.io.File dot, java.lang.String type, java.lang.String representationType) It will call the external dot program, and return the image in binary format.
java.lang.String	getDotSource() Returns the graph's source description in dot language.
byte[]	getGraph(java.lang.String dot_source, java.lang.String type, java.lang.String representationType) Returns the graph as an image in binary format.
int	getImageDpi()
void	increaseDpi() Increase the image size (dpi).
void	readSource(java.lang.String input) Read a DOT graph from a text file.
java.lang.String	start_subgraph(int clusterid) Takes the cluster or subgraph id as input parameter and returns a string that is used to start a subgraph.
java.lang.String	startGraph() Returns a string that is used to start a graph.
private java.io.File	writeDotSourceToFile(java.lang.String str) Writes the source of the graph in a file, and returns the written file as a File object.
int	writeGraphToFile(byte[] img, java.io.File to) Writes the graph's image in a file.
int	writeGraphToFile(byte[] img, java.lang.String file) Writes the graph's image in a file.

◊

Methods inherited from class java.lang.Object

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

- ♦

Field Detail

◊ **TEMP_DIR**

```
private static java.lang.String TEMP_DIR
```

The dir. where temporary files will be created.

◊ **DOT**

```
private static java.lang.String DOT
```

Where is your dot program located? It will be called externally.

◊ **dpiSizes**

```
private final int[] dpiSizes
```

The image size in dpi. 96 dpi is normal size. Higher values are 10% higher each. Lower values 10% lower each. dpi patch by Peter Mueller

◊ **currentDpiPos**

```
private int currentDpiPos
```

Define the index in the image size array.

◊ **graph**

```
private java.lang.StringBuilder graph
```

The source of the graph written in dot language.

♦

Constructor Detail

◊ **Graphviz**

```
public Graphviz()
```

Constructor: creates a new Graphviz object that will contain a graph.

♦

Method Detail

◊ **increaseDpi**

```
public void increaseDpi()
```

Increase the image size (dpi).

◊ **decreaseDpi**

```
public void decreaseDpi()
```

Decrease the image size (dpi).

◊ **getImageDpi**

```
public int getImageDpi()
```

◊ **getDotSource**

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

Returns the graph's source description in dot language.
Returns:
Source of the graph in dot language.

◊ add

```
public void add(java.lang.String line)
```

Adds a string to the graph's source (without newline).

◊ addln

```
public void addln(java.lang.String line)
```

Adds a string to the graph's source (with newline).

◊ addln

```
public void addln()
```

Adds a newline to the graph's source.

◊ clearGraph

```
public void clearGraph()
```

◊ getGraph

```
public byte[] getGraph(java.lang.String dot_source,
                       java.lang.String type,
                       java.lang.String representationType)
```

Returns the graph as an image in binary format.
Parameters:
`dot_source` - Source of the graph to be drawn.
`type` - Type of the output image to be produced, e.g.: gif, dot, fig, pdf, ps, svg, png.
`representationType` - Type of how you want to represent the graph:
- dot
- neato
- fdp
- sfdp
- twopi
- circo

Returns:
A byte array containing the image of the graph.

◊ writeGraphToFile

```
public int writeGraphToFile(byte[] img,
                            java.lang.String file)
```

Writes the graph's image in a file.
Parameters:
`img` - A byte array containing the image of the graph.
`file` - Name of the file to where we want to write.
Returns:
Success: 1, Failure: -1

◊ writeGraphToFile

```
public int writeGraphToFile(byte[] img,
                            java.io.File to)
```

Writes the graph's image in a file.
Parameters:
`img` - A byte array containing the image of the graph.
`to` - A File object to where we want to write.
Returns:
Success: 1, Failure: -1

◊ get_img_stream

```
private byte[] get_img_stream(java.io.File dot,
                              java.lang.String type,
                              java.lang.String representationType)
```

It will call the external dot program, and return the image in binary format.
Parameters:
`dot` - Source of the graph (in dot language).
`type` - Type of the output image to be produced, e.g.: gif, dot, fig, pdf, ps, svg, png.
`representationType` - Type of how you want to represent the graph:
- dot
- neato
- fdp
- sfdp
- twopi
- circo

Returns:
The image of the graph in .gif format.

◊ writeDotSourceToFile

```
private java.io.File writeDotSourceToFile(java.lang.String str)
                                   throws java.io.IOException
```

Writes the source of the graph in a file, and returns the written file as a File object.
Parameters:
`str` - Source of the graph (in dot language).
Returns:
The file (as a File object) that contains the source of the graph.
Throws:
`java.io.IOException`

◊ startGraph

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

Returns a string that is used to start a graph.
Returns:
A string to open a graph.

◊ endGraph

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

Returns a string that is used to end a graph.
Returns:
A string to close a graph.

◊ start_subgraph

```
public java.lang.String start_subgraph(int clusterid)
```

Takes the cluster or subgraph id as input parameter and returns a string that is used to start a subgraph.
Returns:
A string to open a subgraph.

◊ end_subgraph

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

Returns a string that is used to end a graph.
Returns:
A string to close a graph.

◊ readSource

```
public void readSource(java.lang.String input)
```

Read a DOT graph from a text file.
Parameters:
`input` - Input text file containing the DOT graph source.

Skip navigation links

- Overview
- Package
- Class
- Use
- Tree
- Deprecated
- Index
- Help

- Prev Class
- Next Class

- Frames
- No Frames

- All Classes

- Summary:
- Nested |
- Field |
- Constr |
- Method

- Detail:
- Field |
- Constr |
- Method

JavaScript is disabled on your browser.
Skip navigation links

- Overview
- Package
- Class
- Use
- Tree
- Deprecated
- Index
- Help

- Prev Class
- Next Class

- Frames
- No Frames

- All Classes

- Summary:
- Nested |
- Field |
- Constr |
- Method

- Detail:
- Field |
- Constr |
- Method

com.uem.graphviz

Class GraphvizCodeGenerator

- java.lang.Object
- ♦ com.uem.graphviz.GraphvizCodeGenerator

```
public class GraphvizCodeGenerator
extends java.lang.Object
```

This class contains the logic responsible for generating the code that represents the Graph model, which will be sent to Graphviz
Author:
zessin

- ♦

Field Summary

Fields

Modifier and Type	Field and Description
(package private) Graph	graph
(package private) Graphviz	gv

Constructor Summary

Constructors

Constructor and Description
GraphvizCodeGenerator(Graph graph) Initializes the class with an existing Graph model

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method and Description
void	generateCode() Generates the Graphviz code itself
private java.lang.String	printNeutralDegreeVertices() Searches for all the vertices with neutral degree and return them
private java.lang.String	printPositiveDegreeVertices() Searches for all the vertices with positive degree and return them
void	writeGraphFile() After the code is generated, this method calls the Graphviz API for generating the image of the graph

Methods inherited from class java.lang.Object

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

Field Detail

graph

```
final Graph graph
```

gv

```
final Graphviz gv
```

Constructor Detail

GraphvizCodeGenerator

```
public GraphvizCodeGenerator(Graph graph)
```

Initializes the class with an existing Graph model
Parameters:
graph - The graph for whom the code will be generated

♦ Method Detail

◊ generateCode

```
public void generateCode()
```

Generates the Graphviz code itself

◊ writeGraphFile

```
public void writeGraphFile()
```

After the code is generated, this method calls the Graphviz API for generating the image of the graph

◊ printPositiveDegreeVertices

```
private java.lang.String printPositiveDegreeVertices()
```

Searches for all the vertices with positive degree and return them
Returns:
A string with all the positive degree vertices (empty string if no vertices found)

◊ printNeutralDegreeVertices

```
private java.lang.String printNeutralDegreeVertices()
```

Searches for all the vertices with neutral degree and return them
Returns:
A string with all the neutral degree vertices (empty string if no vertices found)

Skip navigation links

- Overview
- Package
- Class
- Use
- Tree
- Deprecated
- Index
- Help

- Prev Class
- Next Class

- Frames
- No Frames

- All Classes

- Summary:
- Nested |
- Field |
- Constr |
- Method

- Detail:
- Field |
- Constr |
- Method

JavaScript is disabled on your browser.
Skip navigation links

- Overview
- Package
- Class
- Use
- Tree
- Deprecated
- Index
- Help

- Prev Class
- Next Class

- Frames
- No Frames

- All Classes

- Summary:
- Nested |
- Field |
- Constr |
- Method

- Detail:
- Field |
- Constr |
- Method

com.uem.main

Class Application

- java.lang.Object
- ♦ com.uem.main.Application

```
public class Application
extends java.lang.Object
```

The Main class of the application. Responsible for initializing all the process.
Author:
zessin

Constructor Summary

Constructors

Constructor and Description
`Application()`

Method Summary

All Methods Static Methods Concrete Methods

Modifier and Type	Method and Description
static void	main(java.lang.String[] args)

- ◊ **Methods inherited from class java.lang.Object**

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

- ♦ **Constructor Detail**

  - ◊ **Application**

    public Application()

- ♦ **Method Detail**

  - ◊ **main**

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

Skip navigation links

- Overview
- Package
- Class
- Use
- Tree
- Deprecated
- Index
- Help

- Prev Class
- Next Class

- Frames
- No Frames

- All Classes

- Summary:
- Nested |
- Field |
- Constr |
- Method

- Detail:
- Field |
- Constr |
- Method

JavaScript is disabled on your browser.
Skip navigation links

- Overview
- Package
- Class
- Use
- Tree
- Deprecated
- Index
- Help

- Prev Class
- Next Class

- Frames
- No Frames

- All Classes

- Summary:
- Nested |
- Field |
- Constr |
- Method

- Detail:
- Field |
- Constr |
- Method

com.uem.main

Class RelationalToGraph

- java.lang.Object
- ♦ com.uem.main.RelationalToGraph

```
public class RelationalToGraph
extends java.lang.Object
```

Responsible for calling all the necessary methods in an organized way, printing some useful information in the console
Author:
zessin

- ♦

Field Summary

Fields

Modifier and Type	Field and Description
private DatabaseInfo	databaseInfo

Constructor Summary

Constructors

Constructor and Description
`RelationalToGraph()` Initializes the class

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method and Description
`void`	`execute()` Executes all the necessary methods for the application to work

Methods inherited from class java.lang.Object

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

Field Detail

databaseInfo

```
private final DatabaseInfo databaseInfo
```

Constructor Detail

RelationalToGraph

```
public RelationalToGraph()
```

Initializes the class

Method Detail

execute

```
public void execute()
```

Executes all the necessary methods for the application to work

- Help

- Prev Class
- Next Class

- Frames
- No Frames

- All Classes

- Summary:
- Nested |
- Field |
- Constr |
- Method

- Detail:
- Field |
- Constr |
- Method

JavaScript is disabled on your browser.
Skip navigation links

- Overview
- Package
- Class
- Use
- Tree
- Deprecated
- Index
- Help

- Prev Class
- Next Class

- Frames
- No Frames

- All Classes

- Summary:
- Nested |
- Field |
- Constr |
- Method

- Detail:
- Field |
- Constr |
- Method

com.uem.util

Class ApplicationLogger

- java.lang.Object
- ♦ com.uem.util.ApplicationLogger

```
public class ApplicationLogger
extends java.lang.Object
```

Utility class which provides the logging methods for the application
Author:
zessin

Field Summary

Fields

Modifier and Type	Field and Description
private static ApplicationLogger	applicationLogger
static java.lang.String	LOG_FILE_NAME
private java.util.logging.Logger	logger

Constructor Summary

Constructors

Modifier	Constructor and Description
private	ApplicationLogger() Initializes the class with the desired format for the log file

Method Summary

All Methods Static Methods Concrete Methods

Modifier and Type	Method and Description
static void	error(java.lang.String message) Logs a message with a SEVERE Level
static void	info(java.lang.String message) Logs a message with an INFO Level
private static void	log(java.util.logging.Level level, java.lang.String message) Singleton method which logs the desired message using the active instance of the ApplicationLogger class
static void	warning(java.lang.String message) Logs a message with a WARNING Level

Methods inherited from class java.lang.Object

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

Field Detail

LOG_FILE_NAME

```
public static final java.lang.String LOG_FILE_NAME
```

See Also:
Constant Field Values

applicationLogger

```
private static ApplicationLogger applicationLogger
```

logger

```
private java.util.logging.Logger logger
```

Constructor Detail

◊ **ApplicationLogger**

```
private ApplicationLogger()
```

Initializes the class with the desired format for the log file

♦ Method Detail

◊ **info**

```
public static void info(java.lang.String message)
```

Logs a message with an INFO Level
Parameters:
`message` - The message to be logged

◊ **warning**

```
public static void warning(java.lang.String message)
```

Logs a message with a WARNING Level
Parameters:
`message` - The message to be logged

◊ **error**

```
public static void error(java.lang.String message)
```

Logs a message with a SEVERE Level
Parameters:
`message` - The message to be logged

◊ **log**

```
private static void log(java.util.logging.Level level,
                        java.lang.String message)
```

Singleton method which logs the desired message using the active instance of the ApplicationLogger class
Parameters:
`level` - Level of the log message
`message` - The log message itself

Skip navigation links

- Overview
- Package
- Class
- Use
- Tree
- Deprecated
- Index
- Help

- Prev Class
- Next Class

- Frames
- No Frames

- All Classes

- Summary:

- Nested |
- Field |
- Constr |
- Method

- Detail:
- Field |
- Constr |
- Method

JavaScript is disabled on your browser.
Skip navigation links

- Overview
- Package
- Class
- Use
- Tree
- Deprecated
- Index
- Help

- Prev Class
- Next Class

- Frames
- No Frames

- All Classes

- Summary:
- Nested |
- Field |
- Constr |
- Method

- Detail:
- Field |
- Constr |
- Method

com.uem.util

Class PropertiesHelper

- java.lang.Object
- ♦ com.uem.util.PropertiesHelper

```
public class PropertiesHelper
extends java.lang.Object
```

Utility class which provides the methods for querying the values of the properties in the application.properties file
Author:
zessin

♦

Field Summary

Fields

Modifier and Type	Field and Description
private static java.lang.String	PROP_COLUMNS_VIEW
private static java.lang.String	PROP_CONSTRAINTS_VIEW
private static java.lang.String	PROP_DATABASE_PASSWORD

private static java.lang.String	PROP_DATABASE_SCHEMA
private static java.lang.String	PROP_DATABASE_TYPE
private static java.lang.String	PROP_DATABASE_URL
private static java.lang.String	PROP_DATABASE_USERNAME
private static java.lang.String	PROP_DOT_PATH
private static java.lang.String	PROP_OUTPUT_PATH
private static java.lang.String	PROP_TABLES_VIEW
private static java.lang.String	PROPERTIES_DIR_NAME
private static java.lang.String	PROPERTIES_FILE_NAME

Constructor Summary

Constructors

Constructor and Description
PropertiesHelper()

Method Summary

All Methods Static Methods Concrete Methods

Modifier and Type	Method and Description
private static void	createDefaultPropertiesFile(java.io.File prop java.util.Properties properties) Creates a default application.properties file
static java.lang.String	getColumnsView() Finds the property which represents the name of the column's view
static java.lang.String	getConstraintsView() Finds the property which represents the name of the constraint's vie
static java.lang.String	getDatabasePassword() Finds the property which represents the database password
static java.lang.String	getDatabaseSchema() Finds the property which represents the database schema
static DatabaseType	getDatabaseType() Finds the property which represents the database type
static java.lang.String	getDatabaseUrl() Finds the property which represents the database URL
static java.lang.String	getDatabaseUsername() Finds the property which represents the database username
static java.lang.String	getDotPath() Finds the property which represents the dot file path

static java.lang.String	getOutputPath() Finds the property which represents the output path for the generate graph image
private static java.io.File	getPropertiesDirectory() Provides the directory in which the application.properties file shall located
private static java.util.Properties	getPropertiesFile() Provides the Properties file for the application.
private static java.lang.String	getPropertyValue(java.lang.String propertyNam Finds the desired property in the application.properties file
static java.lang.String	getTablesView() Finds the property which represents the name of the table's view

◊

Methods inherited from class java.lang.Object

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

- ♦

Field Detail

◊ **PROPERTIES_DIR_NAME**

```
private static final java.lang.String PROPERTIES_DIR_NAME
```

See Also:
Constant Field Values

◊ **PROPERTIES_FILE_NAME**

```
private static final java.lang.String PROPERTIES_FILE_NAME
```

See Also:
Constant Field Values

◊ **PROP_DATABASE_TYPE**

```
private static final java.lang.String PROP_DATABASE_TYPE
```

See Also:
Constant Field Values

◊ **PROP_DATABASE_URL**

```
private static final java.lang.String PROP_DATABASE_URL
```

See Also:
Constant Field Values

◊ **PROP_DATABASE_SCHEMA**

```
private static final java.lang.String PROP_DATABASE_SCHEMA
```

See Also:
Constant Field Values

◊ **PROP_DATABASE_USERNAME**

```
private static final java.lang.String PROP_DATABASE_USERNAME
```

See Also:

Constant Field Values

◊ **PROP_DATABASE_PASSWORD**

```
private static final java.lang.String PROP_DATABASE_PASSWORD
```

See Also:
Constant Field Values

◊ **PROP_TABLES_VIEW**

```
private static final java.lang.String PROP_TABLES_VIEW
```

See Also:
Constant Field Values

◊ **PROP_COLUMNS_VIEW**

```
private static final java.lang.String PROP_COLUMNS_VIEW
```

See Also:
Constant Field Values

◊ **PROP_CONSTRAINTS_VIEW**

```
private static final java.lang.String PROP_CONSTRAINTS_VIEW
```

See Also:
Constant Field Values

◊ **PROP_OUTPUT_PATH**

```
private static final java.lang.String PROP_OUTPUT_PATH
```

See Also:
Constant Field Values

◊ **PROP_DOT_PATH**

```
private static final java.lang.String PROP_DOT_PATH
```

See Also:
Constant Field Values

♦

Constructor Detail

◊ **PropertiesHelper**

```
public PropertiesHelper()
```

♦

Method Detail

◊ **getDatabaseType**

```
public static DatabaseType getDatabaseType()
```

Finds the property which represents the database type
Returns:
The value of the property found

◊ **getDatabaseUrl**

```
public static java.lang.String getDatabaseUrl()
```

Finds the property which represents the database URL
Returns:
The value of the property found

◊ getDatabaseSchema

```
public static java.lang.String getDatabaseSchema()
```

Finds the property which represents the database schema
Returns:
The value of the property found

◊ getDatabaseUsername

```
public static java.lang.String getDatabaseUsername()
```

Finds the property which represents the database username
Returns:
The value of the property found

◊ getDatabasePassword

```
public static java.lang.String getDatabasePassword()
```

Finds the property which represents the database password
Returns:
The value of the property found

◊ getTablesView

```
public static java.lang.String getTablesView()
```

Finds the property which represents the name of the table's view
Returns:
The value of the property found

◊ getColumnsView

```
public static java.lang.String getColumnsView()
```

Finds the property which represents the name of the column's view
Returns:
The value of the property found

◊ getConstraintsView

```
public static java.lang.String getConstraintsView()
```

Finds the property which represents the name of the constraint's view
Returns:
The value of the property found

◊ getOutputPath

```
public static java.lang.String getOutputPath()
```

Finds the property which represents the output path for the generated graph image
Returns:
The value of the property found

◊ getDotPath

```
public static java.lang.String getDotPath()
```

Finds the property which represents the dot file path
Returns:
The value of the property found

◊ getPropertyValue

```
private static java.lang.String getPropertyValue(java.lang.String propertyN
```

Finds the desired property in the application.properties file
Parameters:

propertyName - The name of the property being searched

Returns:

The value of the property found

◊ getPropertiesFile

```
private static java.util.Properties getPropertiesFile()
```

Provides the Properties file for the application. Creates a default one if none was found

Returns:

The Properties file found or created

◊ getPropertiesDirectory

```
private static java.io.File getPropertiesDirectory()
```

Provides the directory in which the application.properties file shall be located

Returns:

The File representing the correct directory

◊ createDefaultPropertiesFile

```
private static void createDefaultPropertiesFile(java.io.File propertiesFile
                                                java.util.Properties proper
                                          throws java.io.IOException
```

Creates a default application.properties file

Parameters:

propertiesFile - The File which shall be created

properties - The Properties which shall be put in the File created

Throws:

java.io.IOException - When the File couldn't be created for some reason