

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			

ORM

## Class mEntity

java.lang.Object  
ORM.mEntity

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

### ORM - Object Relational Mapping

Entidade de persistência dinâmica de objetos baseada na existência da definição de mapeamento da classe.

Qualquer classe que possua uma pré-definição de estrutura de mapeamento junto ao banco de dados relacional poderá ser acessada através de uma entidade dinâmica (vide ORM.mClassCDef para mais informações sobre o controle e registro de definição dos mapeamentos das classes).

Aplicações utilizando esta estrutura de entidade poderão acessar dinamicamente os dados persistentes no banco de dados relacional via JPA (Java Persistence API). O mapeamento das classes é realizado dinamicamente junto ao "class loader" e as tabelas são reconhecidas diretamente no JPQL (Java Persistence Query Language). Todas as propriedades são automaticamente mapeadas para os atributos correspondentes, permitindo ainda o acesso às entidades relacionadas (one-to-one, one-to-many e many-to-one). Os dados podem ser acessados no formato M (tecnologia M - mumps (default)), formato java ou formato externo de apresentação (vide mSystem.NLS para definições de idioma e de formatos de apresentação de datas, horários e números).

Para uma maior flexibilidade de uso e agilidade na conversão do sistema AlphaLinc, esta estrutura possui métodos legados para manipulação de dados no formato de global com mapeamento de campos utilizando um delimitador. A utilização destes métodos legados não é recomendada em novas implementações, pois oneram o desempenho e prejudicam a legibilidade, bem como, a manutenibilidade do código.

### Author:

Innovatium Systems (mosselaar)

## Constructor Summary

### Constructors

#### Constructor and Description

**mEntity**(java.lang.String className)

Cria uma instância (ou registro) de uma entidade conforme definição da classe compilada (a classe deve existir na estrutura de mapeamento de classes compiladas)

## Method Summary

### Methods

#### Modifier and Type

#### Method and Description

static boolean

**exists**(java.lang.Object... idValues)

Verifica existência de registro da entidade

static boolean

**existsId**(java.lang.String idKey)

Verifica existência de registro da entidade

static <b>mEntity</b>	<b>first</b> (java.lang.Object... idValues) Abre o primeiro registro da entidade considerando os valores informados da chave primária
static <b>mEntity</b>	<b>firstId</b> (java.lang.String idKey) Abre o primeiro registro da entidade considerando os valores informados da chave primária
java.lang.Object	<b>get</b> (java.lang.String propertyName) Obtém valor da propriedade da entidade (no formato java)
java.lang.String	<b>getDataDelimiter</b> () Obtém valor do delimitador do registro (método legado - não utilizar em novas implementações)
java.lang.Object	<b>getDataPiece</b> (int index) Obtém valor de uma propriedade do registro conforme índice posicional (no formato M) (método legado - não utilizar em novas implementações)
java.lang.Object	<b>getDataPiece</b> (int from, int to) Obtém valor(es) de propriedade(s) do registro conforme intervalo posicional (no formato M) (método legado - não utilizar em novas implementações)
java.lang.String	<b>getDataRecord</b> () Obtém dados do registro em formato string (valores delimitados) (método legado - não utilizar em novas implementações)
java.lang.String	<b>getDateDisplay</b> (java.lang.String propertyName, int dateformat) Obtém valor da propriedade de data no formato externo de apresentação
java.lang.String	<b>getTimeDisplay</b> (java.lang.String propertyName, int precision) Obtém valor da propriedade de data/hora no formato externo de apresentação
java.lang.String	<b>getTimeDisplay</b> (java.lang.String propertyName, int precision, int dateformat, int timeformat) Obtém valor da propriedade de data/hora no formato externo de apresentação
java.lang.String	<b>getDisplay</b> (java.lang.String propertyName) Obtém valor da propriedade da entidade (no formato externo de apresentação)
java.lang.Object[]	<b>getId</b> () Obtém valores da chave primária
java.lang.String	<b>getIdKey</b> () Obtém string com a chave primária (valores delimitados por   )
java.lang.Object	<b>getM</b> (java.lang.String propertyName) Obtém valor da propriedade da entidade (no formato M)
java.lang.String	<b>getNumericDisplay</b> (java.lang.String propertyName, int decimals) Obtém valor da propriedade numérica no formato externo de apresentação
java.lang.String	<b>getNumericDisplay</b> (java.lang.String propertyName, int decimals, java.lang.String format) Obtém valor da propriedade numérica no formato externo de apresentação
static java.lang.String[]	<b>getPropertyIdList</b> () Obtém relação de propriedades da chave primária da entidade
int	<b>getPropertyIndex</b> (java.lang.String propertyName) Obtém índice posicional de uma propriedade da entidade
static java.lang.String[]	<b>getPropertyList</b> () Obtém relação de todas as propriedades da entidade
java.lang.String	<b>propertyName</b> (int index) Obtém nome da propriedade de um índice posicional
<b>mEntity</b>	<b>getReference</b> (java.lang.String propertyName) Obtém referência de entidade conforme chave estrangeira da propriedade

<b>mEntityCollection</b>	<b>getRelationship</b> (java.lang.String targetClassName) Obtém coleção de entidades relacionadas conforme classe de destino com chave estrangeira
<b>mEntityCollection</b>	<b>getRelationship</b> (java.lang.String targetClassName, java.lang.String targetPropertyName) Obtém coleção de entidades relacionadas conforme classe e propriedade de destino com chave estrangeira (caso classe de destino possua múltiplos relacionamentos)
java.lang.String	<b>getTimeDisplay</b> (java.lang.String propertyName, int precision) Obtém valor da propriedade de hora no formato externo de apresentação
java.lang.String	<b>getTimeDisplay</b> (java.lang.String propertyName, int precision, int timeformat) Obtém valor da propriedade de hora no formato externo de apresentação
void	<b>initialize</b> (java.lang.String... classNames) Inicializa definição das entidades para as classes informadas (as classes devem existir na estrutura de mapeamento de classes compiladas)
static <b>mEntity</b>	<b>last</b> (java.lang.Object... idValues) Abre o último registro da entidade considerando os valores informados da chave primária
static <b>mEntity</b>	<b>lastId</b> (java.lang.String idKey) Abre o último registro da entidade considerando os valores informados da chave primária
boolean	<b>lock</b> () Realiza um lock exclusivo do registro (instância) corrente (select for update)
static boolean	<b>lock</b> (java.lang.Object... idValues) Realiza um lock exclusivo do registro (instância) selecionado (select for update)
static boolean	<b>lockId</b> (java.lang.String idKey) Realiza um lock exclusivo do registro (instância) selecionado (select for update)
boolean	<b>lockShared</b> () Realiza um lock compartilhado do registro (instância) corrente (shared mode)
static boolean	<b>lockShared</b> (java.lang.Object... idValues) Realiza um lock compartilhado do registro (instância) selecionado (shared mode)
static boolean	<b>lockSharedId</b> (java.lang.String idKey) Realiza um lock compartilhado do registro (instância) selecionado (shared mode)
<b>mEntity</b>	<b>next</b> () Avança e abre um registro da entidade a partir da instância corrente
static <b>mEntity</b>	<b>next</b> (java.lang.Object... idValues) Avança e abre um registro da entidade
static <b>mEntity</b>	<b>nextId</b> (java.lang.String idKey) Avança e abre um registro da entidade
boolean	<b>open</b> (java.lang.Object... idValues) Abre um registro da entidade
boolean	<b>openId</b> (java.lang.String idKey) Abre um registro da entidade
<b>mEntity</b>	<b>previous</b> () Retrocede e abre um registro da entidade a partir da instância corrente
static <b>mEntity</b>	<b>previous</b> (java.lang.Object... idValues) Retrocede e abre um registro da entidade
static <b>mEntity</b>	<b>previousId</b> (java.lang.String idKey) Retrocede e abre um registro da entidade
boolean	<b>reload</b> () Recarrega o registro (instância) corrente

boolean	<b>remove ()</b> Remove o registro (instância) corrente (as entidades filhas são excluídas automaticamente) (todas entidades que possuem a chave desta entidade em sua chave primária são consideradas como filhas e possuem a cláusula para exclusão em cascata)
static boolean	<b>remove</b> (java.lang.Object... idValues) Remove o registro (instância) selecionado (as entidades filhas são excluídas automaticamente) (todas entidades que possuem a chave desta entidade em sua chave primária são consideradas como filhas e possuem a cláusula para exclusão em cascata)
static boolean	<b>removeId</b> (java.lang.String idKey) Remove o registro (instância) selecionado (as entidades filhas são excluídas automaticamente) (todas entidades que possuem a chave desta entidade em sua chave primária são consideradas como filhas e possuem a cláusula para exclusão em cascata)
void	<b>save ()</b> Salva o registro (instância) corrente (sem salvar as entidades relacionadas)
void	<b>save</b> (boolean related) Salva o registro (instância) corrente
void	<b>set</b> (java.lang.String propertyName, java.lang.Object value) Atribui valor da propriedade da entidade (no formato java)
void	<b>setDataDelimiter</b> (java.lang.String dataDelimiter) Inicializa valor do delimitador do registro (método legado - não utilizar em novas implementações)
void	<b>setDataPiece</b> (int from, int to, java.lang.Object dataPiece) Atribui valor(es) de propriedade(s) do registro conforme intervalo posicional (no formato M) (método legado - não utilizar em novas implementações)
void	<b>setDataPiece</b> (int index, java.lang.Object dataPiece) Atribui valor de uma propriedade do registro conforme índice posicional (no formato M) (método legado - não utilizar em novas implementações)
void	<b>setDataRecord</b> (java.lang.String dataRecord) Atribui registro em formato string (valores delimitados) (método legado - não utilizar em novas implementações)
void	<b>setDateDisplay</b> (java.lang.String propertyName, java.lang.String value, int dateformat) Atribui valor da propriedade de data no formato externo de apresentação
void	<b>setDateTimeDisplay</b> (java.lang.String propertyName, java.lang.String value, int dateformat, int timeformat) Atribui valor da propriedade de data no formato externo de apresentação
void	<b>setDisplay</b> (java.lang.String propertyName, java.lang.String value) Atribui valor da propriedade da entidade (no formato externo de apresentação)
void	<b>setM</b> (java.lang.String propertyName, java.lang.Object value) Atribui valor da propriedade da entidade (no formato M)
void	<b>setNumericDisplay</b> (java.lang.String propertyName, java.lang.String value, java.lang.String format) Atribui valor da propriedade numérica no formato externo de apresentação
void	<b>setReference</b> (java.lang.String propertyName, mEntity entityRef) Atribui referência de entidade conforme chave estrangeira da propriedade

void	<b>setTimeDisplay</b> (java.lang.String propertyName, java.lang.String value, int timeformat) Atribui valor da propriedade de data no formato externo de apresentação
------	--

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

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

## Constructor Detail

### mEntity

```
public mEntity(java.lang.String className)
```

Cria uma instância (ou registro) de uma entidade conforme definição da classe compilada (a classe deve existir na estrutura de mapeamento de classes compiladas)

**Parameters:**

className - nome da classe compilada

## Method Detail

### initialize

```
public void initialize(java.lang.String... classNames)
```

Inicializa definição das entidades para as classes informadas (as classes devem existir na estrutura de mapeamento de classes compiladas)

**Parameters:**

classNames - nome da(s) classe(s) compilada(s)

### open

```
public boolean open(java.lang.Object... idValues)
```

Abre um registro da entidade

**Parameters:**

idValues - valores da chave primária do registro

**Returns:**

booleano indicando se o registro foi aberto

### openId

```
public boolean openId(java.lang.String idKey)
```

Abre um registro da entidade

**Parameters:**

`idKey` - id do registro (string com a chave primária)

**Returns:**

booleano indicando se o registro foi aberto

**next**

```
public mEntity next()
```

Avança e abre um registro da entidade a partir da instância corrente

**Returns:**

registro da entidade encontrada (nulo caso não for encontrado o registro seguinte)

**next**

```
public static mEntity next(java.lang.Object... idValues)
```

Avança e abre um registro da entidade

**Parameters:**

`idValues` - valores da chave primária do registro

**Returns:**

registro da entidade encontrada (nulo caso não for encontrado o registro seguinte)

**nextId**

```
public static mEntity nextId(java.lang.String idKey)
```

Avança e abre um registro da entidade

**Parameters:**

`idKey` - id do registro (string com a chave primária)

**Returns:**

registro da entidade encontrada (nulo caso não for encontrado o registro seguinte)

**previous**

```
public mEntity previous()
```

Retrocede e abre um registro da entidade a partir da instância corrente

**Returns:**

registro da entidade encontrada (nulo caso não for encontrado o registro seguinte)

**previous**

```
public static mEntity previous(java.lang.Object... idValues)
```

Retrocede e abre um registro da entidade

**Parameters:**

idValues - valores da chave primária do registro

**Returns:**

registro da entidade encontrada (nulo caso não for encontrado o registro seguinte)

**previousId**

```
public static mEntity previousId(java.lang.String idKey)
```

Retrocede e abre um registro da entidade

**Parameters:**

idKey - id do registro (string com a chave primária)

**Returns:**

registro da entidade encontrada (nulo caso não for encontrado o registro seguinte)

**first**

```
public static mEntity first(java.lang.Object... idValues)
```

Abre o primeiro registro da entidade considerando os valores informados da chave primária

**Parameters:**

idValues - valores da chave primária do registro

**Returns:**

registro da entidade encontrada (nulo caso não for encontrado o registro seguinte)

**firstId**

```
public static mEntity firstId(java.lang.String idKey)
```

Abre o primeiro registro da entidade considerando os valores informados da chave primária

**Parameters:**

idKey - id do registro (string com a chave primária)

**Returns:**

registro da entidade encontrada (nulo caso não for encontrado o registro seguinte)

**last**

```
public static mEntity last(java.lang.Object... idValues)
```

Abre o último registro da entidade considerando os valores informados da chave primária

**Parameters:**

`idValues` - valores da chave primária do registro

**Returns:**

registro da entidade encontrada (nulo caso não for encontrado o registro seguinte)

**lastId**

```
public static mEntity lastId(java.lang.String idKey)
```

Abre o último registro da entidade considerando os valores informados da chave primária

**Parameters:**

`idKey` - id do registro (string com a chave primária)

**Returns:**

registro da entidade encontrada (nulo caso não for encontrado o registro seguinte)

**exists**

```
public static boolean exists(java.lang.Object... idValues)
```

Verifica existência de registro da entidade

**Parameters:**

`idValues` - valores da chave primária do registro

**Returns:**

booleano indicando existência do registro

**existsId**

```
public static boolean existsId(java.lang.String idKey)
```

Verifica existência de registro da entidade

**Parameters:**

`idKey` - id do registro (string com a chave primária)

**Returns:**

booleano indicando existência do registro

**save**

```
public void save(boolean related)
```



Salva o registro (instância) corrente

**Parameters:**

`related` - booleano indicando a necessidade de salvar as entidades relacionadas (todas as referências e relacionamentos desta instância são consideradas como entidades relacionadas)

**save**

```
public void save()
```

Salva o registro (instância) corrente (sem salvar as entidades relacionadas)

**remove**

```
public boolean remove()
```

Remove o registro (instância) corrente (as entidades filhas são excluídas automaticamente) (todas entidades que possuem a chave desta entidade em sua chave primária são consideradas como filhas e possuem a cláusula para exclusão em cascata)

**Returns:**

booleano indicando se o registro foi removido (excluído fisicamente)

**remove**

```
public static boolean remove(java.lang.Object... idValues)
```

Remove o registro (instância) selecionado (as entidades filhas são excluídas automaticamente) (todas entidades que possuem a chave desta entidade em sua chave primária são consideradas como filhas e possuem a cláusula para exclusão em cascata)

**Parameters:**

`idValues` - valores da chave primária do registro

**Returns:**

booleano indicando se o registro foi removido (excluído fisicamente)

**removeId**

```
public static boolean removeId(java.lang.String idKey)
```

Remove o registro (instância) selecionado (as entidades filhas são excluídas automaticamente) (todas entidades que possuem a chave desta entidade em sua chave primária são consideradas como filhas e possuem a cláusula para exclusão em cascata)

**Parameters:**

`idKey` - id do registro (string com a chave primária)

**Returns:**

booleano indicando se o registro foi removido (excluído fisicamente)

**reload**

```
public boolean reload()
```

Recarrega o registro (instância) corrente

**Returns:**

booleano indicando se o registro foi recarregado (somente registros que não existe na base não são carregados, demais erros geram uma exceção)

**lock**

```
public boolean lock()
```

Realiza um lock exclusivo do registro (instância) corrente (select for update)

**Returns:**

booleano indicando se o registro foi locado

**lock**

```
public static boolean lock(java.lang.Object... idValues)
```

Realiza um lock exclusivo do registro (instância) selecionado (select for update)

**Parameters:**

`idValues` - valores da chave primária do registro

**Returns:**

booleano indicando se o registro foi locado

**lockId**

```
public static boolean lockId(java.lang.String idKey)
```

Realiza um lock exclusivo do registro (instância) selecionado (select for update)

**Parameters:**

`idKey` - id do registro (string com a chave primária)

**Returns:**

booleano indicando se o registro foi locado

**lockShared**

```
public boolean lockShared()
```

Realiza um lock compartilhado do registro (instância) corrente (shared mode)

**Returns:**

booleano indicando se o registro foi locado

**lockShared**

```
public static boolean lockShared(java.lang.Object... idValues)
```

Realiza um lock compartilhado do registro (instância) selecionado (shared mode)

**Parameters:**

idValues - valores da chave primária do registro

**Returns:**

booleano indicando se o registro foi locado

**lockSharedId**

```
public static boolean lockSharedId(java.lang.String idKey)
```

Realiza um lock compartilhado do registro (instância) selecionado (shared mode)

**Parameters:**

idKey - id do registro (string com a chave primária)

**Returns:**

booleano indicando se o registro foi locado

**get**

```
public java.lang.Object get(java.lang.String propertyName)
```

Obtém valor da propriedade da entidade (no formato java)

**Parameters:**

propertyName - nome da propriedade da entidade

**Returns:**

valor da propriedade da entidade

**set**

```
public void set(java.lang.String propertyName,  
               java.lang.Object value)
```

Atribui valor da propriedade da entidade (no formato java)

**Parameters:**

propertyName - nome da propriedade da entidade

value - valor da propriedade da entidade

**getM**

```
public java.lang.Object getM(java.lang.String propertyName)
```

Obtém valor da propriedade da entidade (no formato M)

**Parameters:**

propertyName - nome da propriedade da entidade

**Returns:**

valor da propriedade da entidade

**setM**

```
public void setM(java.lang.String propertyName,  
                java.lang.Object value)
```

Atribui valor da propriedade da entidade (no formato M)

**Parameters:**

propertyName - nome da propriedade da entidade

value - valor da propriedade da entidade

**getDisplay**

```
public java.lang.String getDisplay(java.lang.String propertyName)
```

Obtém valor da propriedade da entidade (no formato externo de apresentação)

**Parameters:**

propertyName - nome da propriedade da entidade

**Returns:**

valor da propriedade da entidade

**getNumericDisplay**

```
public java.lang.String getNumericDisplay(java.lang.String propertyName,  
                                          int decimals)
```

Obtém valor da propriedade numérica no formato externo de apresentação

**Parameters:**

propertyName - nome da propriedade da entidade

decimals - número de casas decimais

**Returns:**

valor formatado da propriedade

**getNumericDisplay**

```
public java.lang.String getNumericDisplay(java.lang.String propertyName,  
                                         int decimals,  
                                         java.lang.String format)
```

Obtém valor da propriedade numérica no formato externo de apresentação

**Parameters:**

`propertyName` - nome da propriedade da entidade

`decimals` - número de casas decimais

`format` - formato de decimais e separador de milhar (primeiro caracter determina o separador de decimais e o segundo caracter determina o separador de milhar - exemplos: ".", " => 9,999,999.99 / ", " => 9.999.999,99)

**Returns:**

valor formatado da propriedade

**getDateDisplay**

```
public java.lang.String getDateDisplay(java.lang.String propertyName,  
                                       int dateformat)
```

Obtém valor da propriedade de data no formato externo de apresentação

**Parameters:**

`propertyName` - nome da propriedade da entidade

`dateformat` - código de formatação da data (vide opções em \$zdate)

**Returns:**

valor formatado da propriedade

**getTimeDisplay**

```
public java.lang.String getTimeDisplay(java.lang.String propertyName,  
                                       int precision)
```

Obtém valor da propriedade de hora no formato externo de apresentação

**Parameters:**

`propertyName` - nome da propriedade da entidade

`precision` - precisão de fração de segundos

**Returns:**

valor formatado da propriedade

**getTimeDisplay**

```
public java.lang.String getTimeDisplay(java.lang.String propertyName,  
                                       int precision,  
                                       int timeformat)
```

Obtém valor da propriedade de hora no formato externo de apresentação

**Parameters:**

`propertyName` - nome da propriedade da entidade

`precision` - precisão de fração de segundos

`timeformat` - código de formatação da hora (vide opções em `$ztime`)

**Returns:**

valor formatado da propriedade

### getTimeDisplay

```
public java.lang.String getTimeDisplay(java.lang.String propertyName,  
                                       int precision)
```

Obtém valor da propriedade de data/hora no formato externo de apresentação

**Parameters:**

`propertyName` - nome da propriedade da entidade

`precision` - precisão de fração de segundos

**Returns:**

valor formatado da propriedade

### getTimeDisplay

```
public java.lang.String getTimeDisplay(java.lang.String propertyName,  
                                       int precision,  
                                       int dateformat,  
                                       int timeformat)
```

Obtém valor da propriedade de data/hora no formato externo de apresentação

**Parameters:**

`propertyName` - nome da propriedade da entidade

`precision` - precisão de fração de segundos

`dateformat` - código de formatação da data (vide opções em `$zdate`)

`timeformat` - código de formatação da hora (vide opções em `$ztime`)

**Returns:**

valor formatado da propriedade

### setDisplay

```
public void setDisplay(java.lang.String propertyName,  
                      java.lang.String value)
```

Atribui valor da propriedade da entidade (no formato externo de apresentação)

**Parameters:**

propertyName - nome da propriedade da entidade

value - valor da propriedade da entidade

### setNumericDisplay

```
public void setNumericDisplay(java.lang.String propertyName,  
                             java.lang.String value,  
                             java.lang.String format)
```

Atribui valor da propriedade numérica no formato externo de apresentação

#### Parameters:

propertyName - nome da propriedade da entidade

format - formato de decimais e separador de milhar (primeiro caracter determina o separador de decimais e o segundo caracter determina o separador de milhar - exemplos: ".", => 9,999,999.99 / ",", => 9.999.999,99)

value - valor formatado da propriedade

### setDateDisplay

```
public void setDateDisplay(java.lang.String propertyName,  
                          java.lang.String value,  
                          int dateformat)
```

Atribui valor da propriedade de data no formato externo de apresentação

#### Parameters:

propertyName - nome da propriedade da entidade

dateformat - código de formatação da data (vide opções em \$zdate)

value - valor formatado da propriedade

### setTimeDisplay

```
public void setTimeDisplay(java.lang.String propertyName,  
                          java.lang.String value,  
                          int timeformat)
```

Atribui valor da propriedade de data no formato externo de apresentação

#### Parameters:

propertyName - nome da propriedade da entidade

timeformat - código de formatação da hora (vide opções em \$ztime)

value - valor formatado da propriedade

### setDateTimeDisplay

```
public void setDateTimeDisplay(java.lang.String propertyName,  
                               java.lang.String value,  
                               int dateformat,  
                               int timeformat)
```

Atribui valor da propriedade de data no formato externo de apresentação

**Parameters:**

propertyName - nome da propriedade da entidade

dateformat - código de formatação da data (vide opções em \$zdate)

timeformat - código de formatação da hora (vide opções em \$ztime)

value - valor formatado da propriedade

### getReference

```
public mEntity getReference(java.lang.String propertyName)
```

Obtém referência de entidade conforme chave estrangeira da propriedade

**Parameters:**

propertyName - nome da propriedade da entidade

**Returns:**

instância da entidade referenciada (registro)

### setReference

```
public void setReference(java.lang.String propertyName,  
                          mEntity entityRef)
```

Atribui referência de entidade conforme chave estrangeira da propriedade

**Parameters:**

propertyName - nome da propriedade da entidade

entityRef - instância da entidade referenciada (registro)

### getRelationship

```
public mEntityCollection getRelationship(java.lang.String targetClassName)
```

Obtém coleção de entidades relacionadas conforme classe de destino com chave estrangeira

**Parameters:**

targetClassName - nome da classe de destino

**Returns:**

coleção de entidades relacionadas

### getRelationship



```
public mEntityCollection getRelationship(java.lang.String targetClassName,  
                                       java.lang.String targetPropertyName)
```

Obtém coleção de entidades relacionadas conforme classe e propriedade de destino com chave estrangeira (caso classe de destino possua múltiplos relacionamentos)

**Parameters:**

targetClassName - nome da classe de destino

targetPropertyName - nome da propriedade de destino com chave estrangeira

**Returns:**

coleção de entidades relacionadas

### getId

```
public java.lang.Object[] getId()
```

Obtém valores da chave primária

**Returns:**

array de valores da chave primária

### getIdKey

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

Obtém string com a chave primária (valores delimitados por ||)

**Returns:**

idKey do registro

### getPropertyIdList

```
public static java.lang.String[] getPropertyIdList()
```

Obtém relação de propriedades da chave primária da entidade

**Returns:**

array de nomes das propriedades da chave primária

### getPropertyList

```
public static java.lang.String[] getPropertyList()
```

Obtém relação de todas as propriedades da entidade

**Returns:**

array de nomes das propriedades da entidade

**getPropertyName**

```
public java.lang.String getPropertyName(int index)
```

Obtém nome da propriedade de um índice posicional

**Parameters:**

`index` - índice posicional da propriedade

**Returns:**

nome da propriedade

**getPropertyIndex**

```
public int getPropertyIndex(java.lang.String propertyName)
```

Obtém índice posicional de uma propriedade da entidade

**Parameters:**

`propertyName` - nome da propriedade

**Returns:**

índice posicional da propriedade

**getDataRecord**

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

Obtém dados do registro em formato string (valores delimitados) (método legado - não utilizar em novas implementações)

**Returns:**

registro em formato string

**setDataRecord**

```
public void setDataRecord(java.lang.String dataRecord)
```

Atribui registro em formato string (valores delimitados) (método legado - não utilizar em novas implementações)

**Parameters:**

`dataRecord` - registro em formato string

**getDataDelimiter**

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

Obtém valor do delimitador do registro (método legado - não utilizar em novas implementações)

**Returns:**

valor do delimitador

**setDataDelimiter**

```
public void setDataDelimiter(java.lang.String dataDelimiter)
```

Inicializa valor do delimitador do registro (método legado - não utilizar em novas implementações)

**Parameters:**

`dataDelimiter` - valor do delimitador

**getDataPiece**

```
public java.lang.Object getDataPiece(int index)
```

Obtém valor de uma propriedade do registro conforme índice posicional (no formato M) (método legado - não utilizar em novas implementações)

**Parameters:**

`index` - índice posicional

**Returns:**

valor da propriedade

**getDataPiece**

```
public java.lang.Object getDataPiece(int from,  
                                     int to)
```

Obtém valor(es) de propriedade(s) do registro conforme intervalo posicional (no formato M) (método legado - não utilizar em novas implementações)

**Parameters:**

`from` - índice posicional de início

`to` - índice posicional de término

**Returns:**

valor(es) da(s) propriedade(s) (string com delimitador caso houver mais de um valor)

**setDataPiece**

```
public void setDataPiece(int index,  
                         java.lang.Object dataPiece)
```

Atribui valor de uma propriedade do registro conforme índice posicional (no formato M) (método legado - não utilizar em novas implementações)

**Parameters:**

`index` - índice posicional

`dataPiece` - valor da propriedade

**setDataPiece**

```
public void setDataPiece(int from,
                        int to,
                        java.lang.Object dataPiece)
```

Atribui valor(es) de propriedade(s) do registro conforme intervalo posicional (no formato M) (método legado - não utilizar em novas implementações)

**Parameters:**

`from` - índice posicional de início

`to` - índice posicional de término

`dataPiece` - valor(es) da(s) propriedade(s) (string com delimitador caso houver mais de um valor)

[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](#)