## 2 类级别操作

**2.2创建代理类功能实现**

2.2.1 createDeputyClass

该函数的主要功能是创建并插入新的classTableItem和switchingTableItem。

根据输入的参数创建一个代理类，并在类表和交换表中分别添加相应的信息。具体步骤如下：

（1）生成新的classId，即类ID。

（2）遍历selectResult，根据其中的信息创建ClassTableItem对象，并将其添加到ClassTableList中。

（3）根据selectResult中的信息，创建SwitchingTableItem对象，并将其添加到SwitchingTableList中。

（4）返回生成的类ID。

在该函数中，主要用到了MemConnect类中的静态方法，如getClassTable()、getClassTableList()、getSwitchingTableList()等，用于获取类表、类表列表和交换表列表等信息。同时，为了获取selectResult中的属性ID、属性名、属性类型和类名等信息，还用到了SelectResult类的getAttrid()、getAttrname()、getType()、getClassName()等方法。

2.2.2 createDeputyTableItem

该函数的功能是创建并插入新的deputyTableItem。

通过传入的className数组、deputyType和deputyId参数，创建一个DeputyTableItem对象，并将其添加到MemConnect的deputyTableList中。具体实现过程如下：

（1）使用Arrays.stream()将className数组转换为流，然后使用Collectors.toCollection()方法将其收集到HashSet中，以去除重复元素。

（2）遍历去重后的className集合，对于每个className，通过memConnect.getClassId()方法获取其对应的classId。

（3）使用classId、deputyId和将deputyType转换为字符串数组的新数组，创建一个DeputyTableItem对象。

（4）最后，通过MemConnect.getDeputyTableList().add()方法将DeputyTableItem对象添加到deputyTableList中。

2.2.3 createBiPointerTableItem

该函数主要功能是创建并插入新的biPointerTableItem。

根据传入的SelectResult对象和deputyId，创建并插入BiPointerTableItem对象到BiPointerTable列表中。具体步骤如下：

（1）创建InsertImpl实例用于执行插入操作。

（2）调用Arrays.asList()方法将selectResult的属性名由字符串数组转换为列表形式。

（3）遍历selectResult中的元组。

（4）对于每个元组：使用insert.execute()方法插入代理类对象，并获取插入后的代理对象ID。调用getOriginClass()方法去除重复元素后将结果保存在origin集合中。

（5）然后遍历origin集合，对于每个元素：调用memConnect.getClassId()方法获取源类ID。根据元组信息创建BiPointerTableItem对象。通过MemConnect.getBiPointerTableList().add()将BiPointerTableItem插入到BiPointerTable列表中。