|  |  |
| --- | --- |
| **ID** | 24 |
| **Verfahren** | Parametrisierter JUnit-Test |
| **Klasse** | HSQLDBManager |
| **Methoden** | init()  Alle vorhandenen Tabellen werden gelöscht und neu erstellt |
| **Vorbedingung** | Datenbank ist aktiv |
| **Eingaben** | - |
| **Erwartetes Ergebnis** | Tabelleninhalt von der Tabelle *GENERATIONS* kann abgerufen werden |
| **Ergebnis** | Tabelleninhalt von der Tabelle *GENERATIONS* kann abgerufen werden  **Test erfolgreich** |

|  |  |
| --- | --- |
| **ID** | 25 |
| **Verfahren** | Parametrisierter JUnit-Test |
| **Klasse** | HSQLDBManager |
| **Methoden** | update(String sqlStatement)  Ein gegebenes SQL-Statement wird ausgeführt |
| **Vorbedingung** | Datenbank ist aktiv |
| **Eingaben** | "INSERT INTO GENERATIONS ( id , generation, route , distance ) " + "VALUES ( "+ 1 + "," + 1 + "," + "'0,1,0'" + "," + 2 + " ) " |
| **Erwartetes Ergebnis** | Tabelle *Generations* hat eine Zeile |
| **Ergebnis** | Tabelle *Generations* hat eine Zeile  **Test erfolgreich** |

|  |  |
| --- | --- |
| **ID** | 26 |
| **Verfahren** | Parametrisierter JUnit-Test |
| **Klasse** | HSQLDBManager |
| **Methoden** | select(String sqlStatement): ResultSet  Ein gegebenes SQL-Statement wird ausgeführt und das Resultat zurückgegeben |
| **Vorbedingung** | Datenbank ist aktiv  Tabelle *Generations* hat eine Zeile mit vordefinierten Daten |
| **Eingaben** | "SELECT \* FROM GENERATIONS" |
| **Erwartetes Ergebnis** | Integer in der Spalte *generation* ist 1  String in der Spalte *route* ist 0,1,0  Integer in der Spalte *distance* ist 2 |
| **Ergebnis** | Integer in der Spalte *generation* ist 1  String in der Spalte *route* ist 0,1,0  Integer in der Spalte *distance* ist 2  **Test erfolgreich** |

|  |  |
| --- | --- |
| **ID** | 27 |
| **Verfahren** | Parametrisierter JUnit-Test |
| **Klasse** | HSQLDBManager |
| **Methoden** | getTable(String table): ResultSet  Eine komplette Tabelle wird abgerufen und zurückgegeben |
| **Vorbedingung** | Datenbank ist aktiv |
| **Eingaben** | "GENERATIONS" |
| **Erwartetes Ergebnis** | Tabelle *GENERATIONS* hat 4 Spalten |
| **Ergebnis** | Tabelle *GENERATIONS* hat 4 Spalten  **Test erfolgreich** |

|  |  |
| --- | --- |
| **ID** | 28 |
| **Verfahren** | Parametrisierter JUnit-Test |
| **Klasse** | HSQLDBManager |
| **Methoden** | dropTable()  Alle vorhandenen Tabellen werden gelöscht |
| **Vorbedingung** | Datenbank ist aktiv  Tabelle *GENERATIONS* ist vorhanden |
| **Eingaben** | - |
| **Erwartetes Ergebnis** | Tabelle *GENERATIONS* ist nicht mehr vorhanden (NullPointerException muss geworfen werden) |
| **Ergebnis** | NullPointerException wird geworfen  **Test erfolgreich** |

|  |  |
| --- | --- |
| **ID** | 29 |
| **Verfahren** | Parametrisierter JUnit-Test |
| **Klasse** | HSQLDBManager |
| **Methoden** | createTable()  Es werden alle Tabellen neu erstellt |
| **Vorbedingung** | Datenbank ist aktiv |
| **Eingaben** | - |
| **Erwartetes Ergebnis** | Tabelle *GENERATIONS* hat 4 Spalten  Tabelle *HISTORY* hat 4 Spalten  Tabelle *TEST* existiert nicht |
| **Ergebnis** | Tabelle *GENERATIONS* hat 4 Spalten  Tabelle *HISTORY* hat 4 Spalten  Tabelle *TEST* existiert nicht  **Test erfolgreich** |

|  |  |
| --- | --- |
| **ID** | 30 |
| **Verfahren** | Parametrisierter JUnit-Test |
| **Klasse** | HSQLDBManager |
| **Methoden** | updateTable(String table, int generation, String route, int distance)  Auf eine gegebene Tabelle wird ein INSERT mit den gegebenen Daten ausgeführt |
| **Vorbedingung** | Datenbank ist aktiv |
| **Eingaben** | "GENERATIONS", 1, „0,1,0“, 2 |
| **Erwartetes Ergebnis** | Tabelle *GENERATIONS* hat eine Zeile |
| **Ergebnis** | Tabelle *GENERATIONS* hat eine Zeile  **Test erfolgreich** |