

White Box Testfälle

Version 1.0

zuletzt geändert am 05.06.04

1 Überblick

Es sind für drei wichtige Klassen des Projekts White Box Testfälle zu erstellen. Für alle nicht trivialen Methoden der einzelnen Klassen werden Kontrollflussdiagramme erstellt. Basierend auf diesen Diagrammen können dann die entsprechenden White Box Testfälle erstellt.

Der White Box Test (Strukturtest) ist ein logisch-orientierter Test. Die Testdaten werden mit Kenntnis der Programmlogik definiert. Angestrebt wird ein vollständiger Pfadtest. White Box Testen konzentriert sich auf konkreten Code innerhalb einer Klasse bzw. Methode. Der Zugang zum Quellcode ist also unumgänglich. Durch gezielte Tests muss das Funktionieren von Konstruktionselementen (Schleifen, Verzweigungen, usw.) überprüft werden. In einem Kontrollflussdiagramm werden in diesem Dokument alle Ausführungspfade der einzelnen Methoden folgender drei Klassen beschrieben:

- Klasse PostgresqlProcessor aus dem Package Processors
- Klasse SearchObject aus dem Package db
- Klasse Proxy aus dem Package db

Anhand von so genannten Überdeckungsgraden werden in weiterer Folge die Testfälle erstellt. Die Testfälle werden also im Vergleich zu Black Box Testfällen noch um eine Spalte mit den Überdeckungsgraden erweitert. Durch sie kann eingesehen werden, welche Ausführungspfade des Codes durch die Testfälle abgedeckt werden. Sie bestimmt auch maßgeblich die Anzahl solcher Testfälle. Alternativ zum Kontrollflussdiagramm können auch Sequenzdiagramme, also dynamische Diagramme, benutzt werden um Testfälle zu erstellen.

2 Beispiele aus dem Projekt Ticket-Line

In diesem Absatz möchte ich die Wahl der zu testenden Klassen begründen. Bei allen drei Klassen handelt es sich um sehr generische Klassen. Also Klassen, welche oft verwendete Methoden zur Verfügung stellen und somit eine Art Fundament für das Projekt darstellen. Das Funktionieren solcher Klasse ist somit nicht nur quantitativ sondern auch qualitativ sicherzustellen. PostgresqlProcessor bietet eine Schnittstelle zu eine Postgresqldatenbank an. Ohne sie wäre die Realisierung des Ticketlineprojekts gar nicht möglich. SearchObject ermöglicht es eine Suchabfrage zu erstellen. Die Proxyklasse repräsentiert die Implementierung der Proxyschicht. Diese dient der Abstraktion von Entitäten über Interfaces.

3 Beschreibung der Kontrollflussdiagramme

Kontrollflussdiagramme bestehen aus folgenden grundlegenden Elementen:

- **Anweisungen**, welche aus Rechtecke mit abgerundeten Kanten bestehen
- Alle **Folgeanweisungen** werden durch **Pfeile** mit der Pfeilspitze in Richtung der Folgeanweisung mit der vorhergehenden Anweisung verbunden.

4 Beschreibung der Testfälletabellen

Die Tabellen bestehen aus folgenden Spalten:

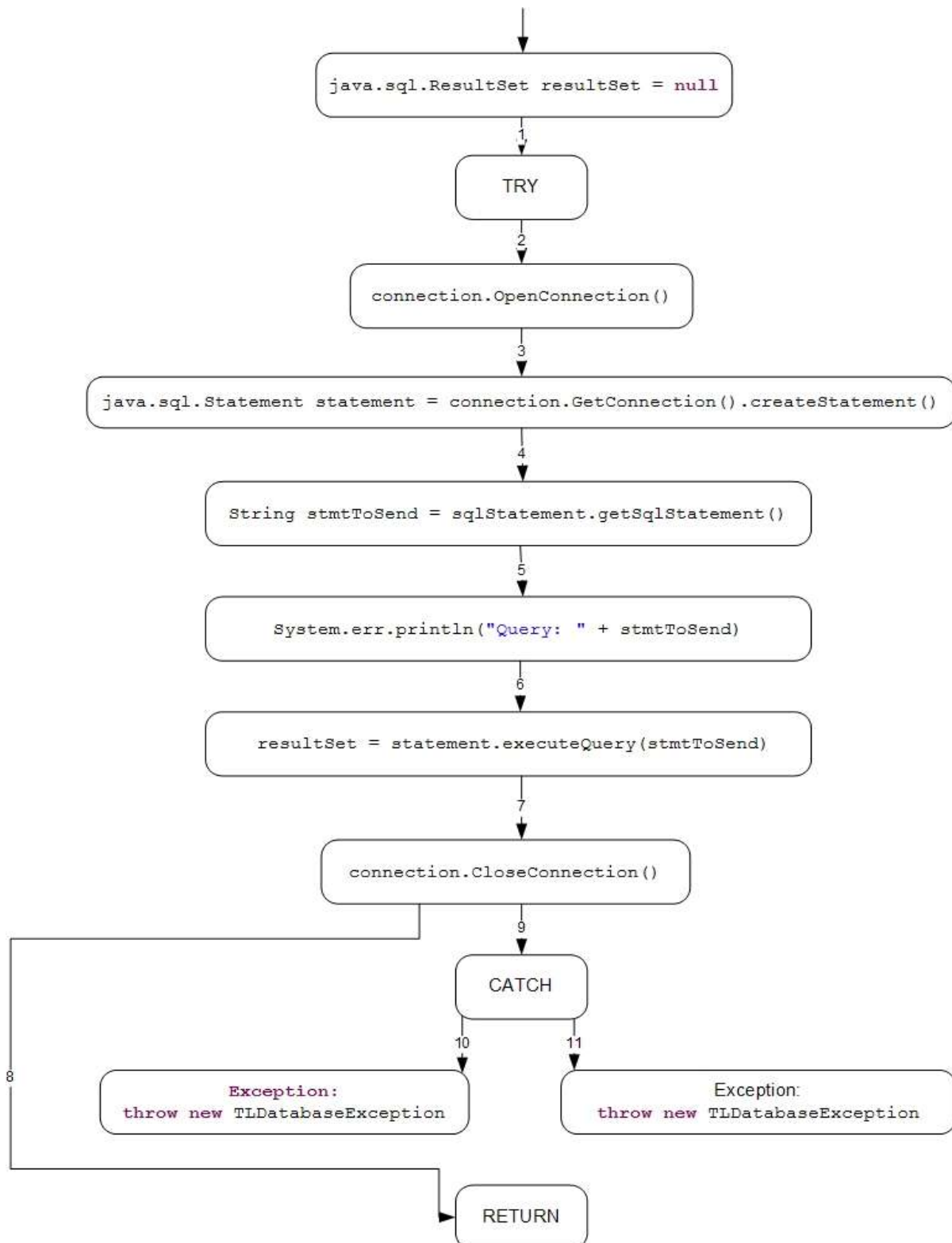
- **Nr.** gibt die Nummer des Testfalls an.
- **Klasse** teilt eine Testfall als NF (Normalfall) , SF (Sonderfall) und FF (Fehlerfall)
- **Beschreibung des Testfalls**
- **Erwartetes Ergebnis**
- **Eingabedaten**
- **Testergebnis**

5 Kontrollflussdiagramme

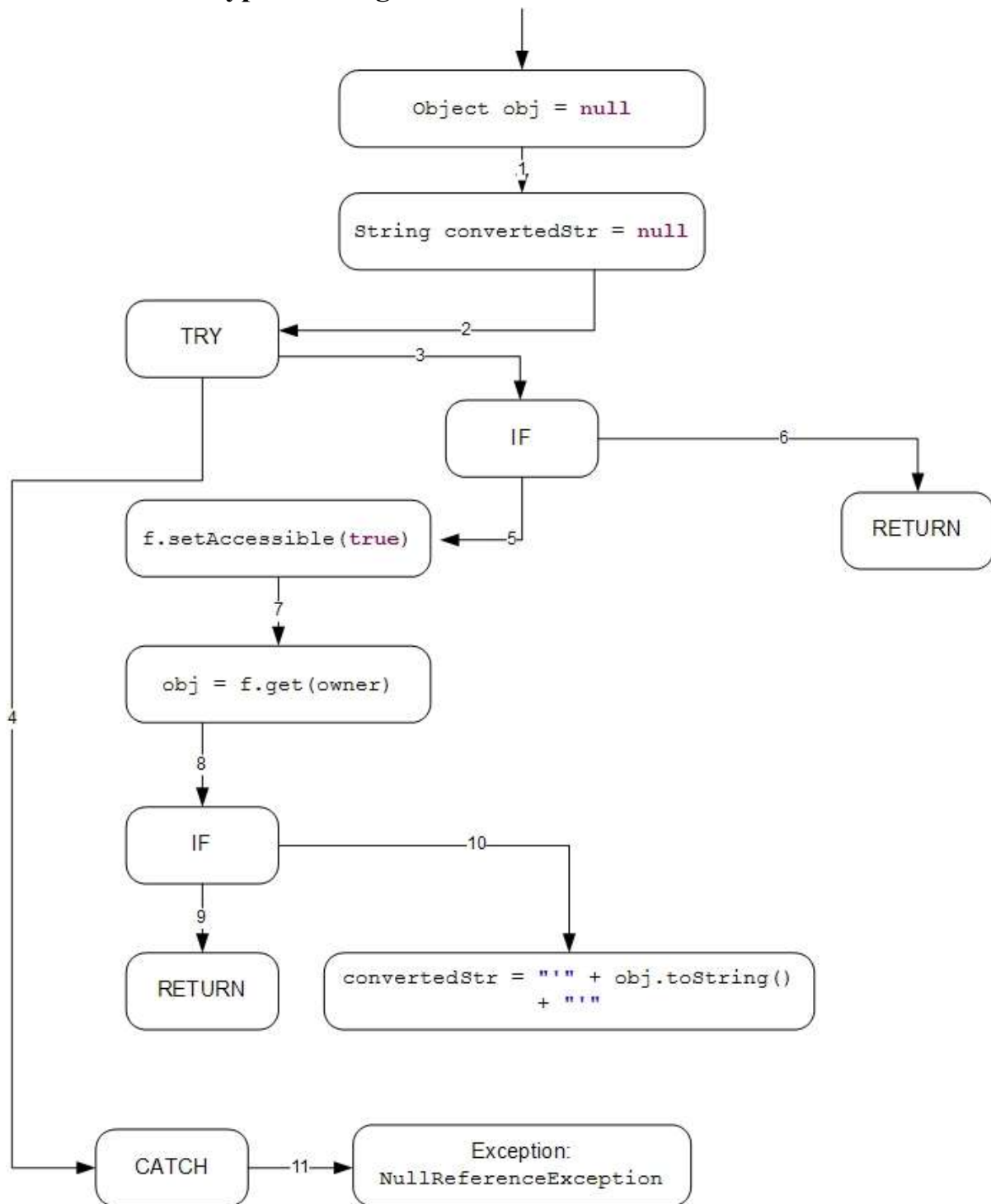
In diesem Abschnitt folgen nun die Kontrollflussdiagramme. Die entsprechenden Testfälle finden sich in der nächsten Hauptgliederung → 6 und den korrespondierenden Unterpunkten.

5.1 Klasse PostgresqlProcessor

5.1.1 Methode CommitSelect

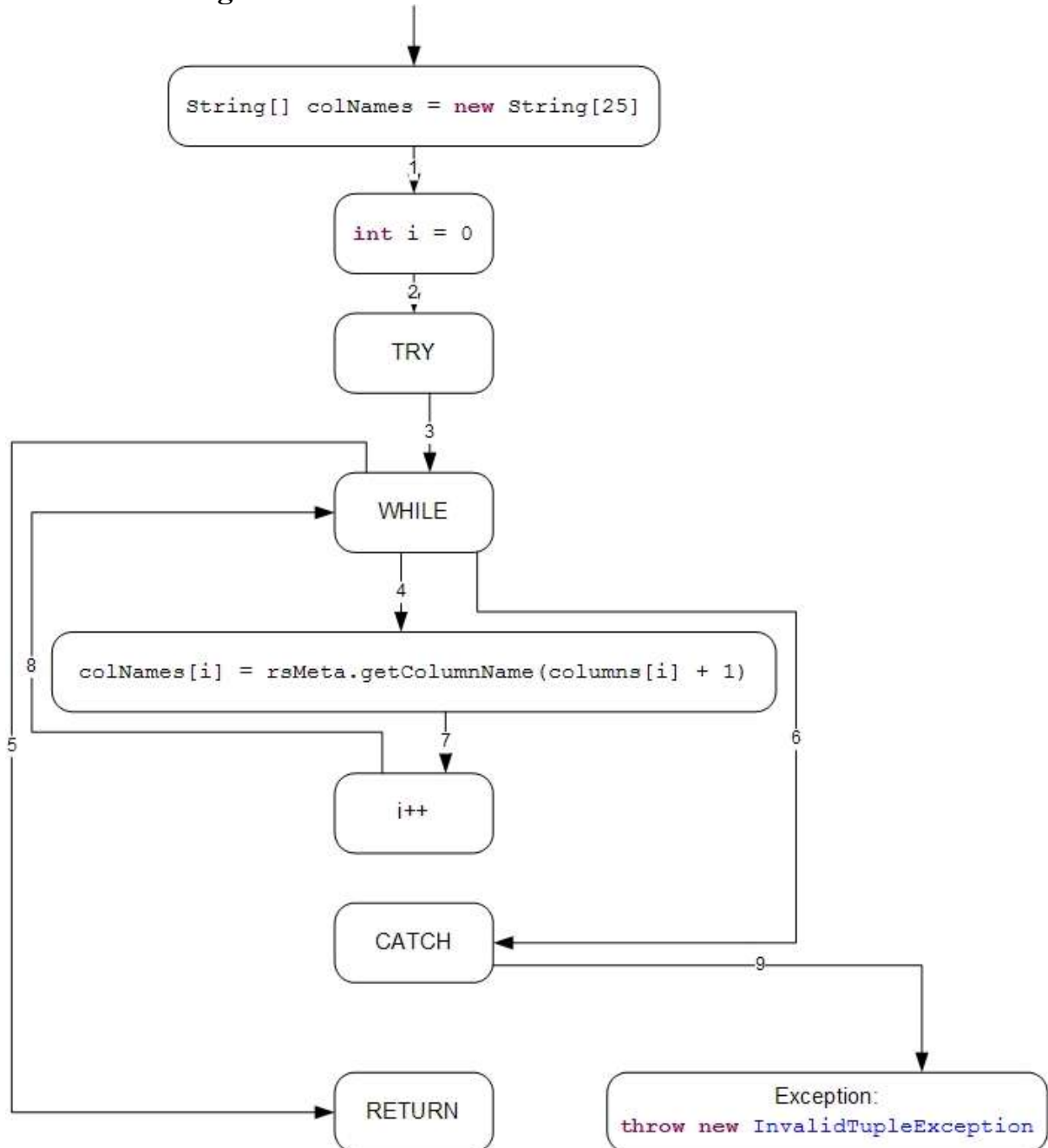


5.1.2 Methode TypeToString



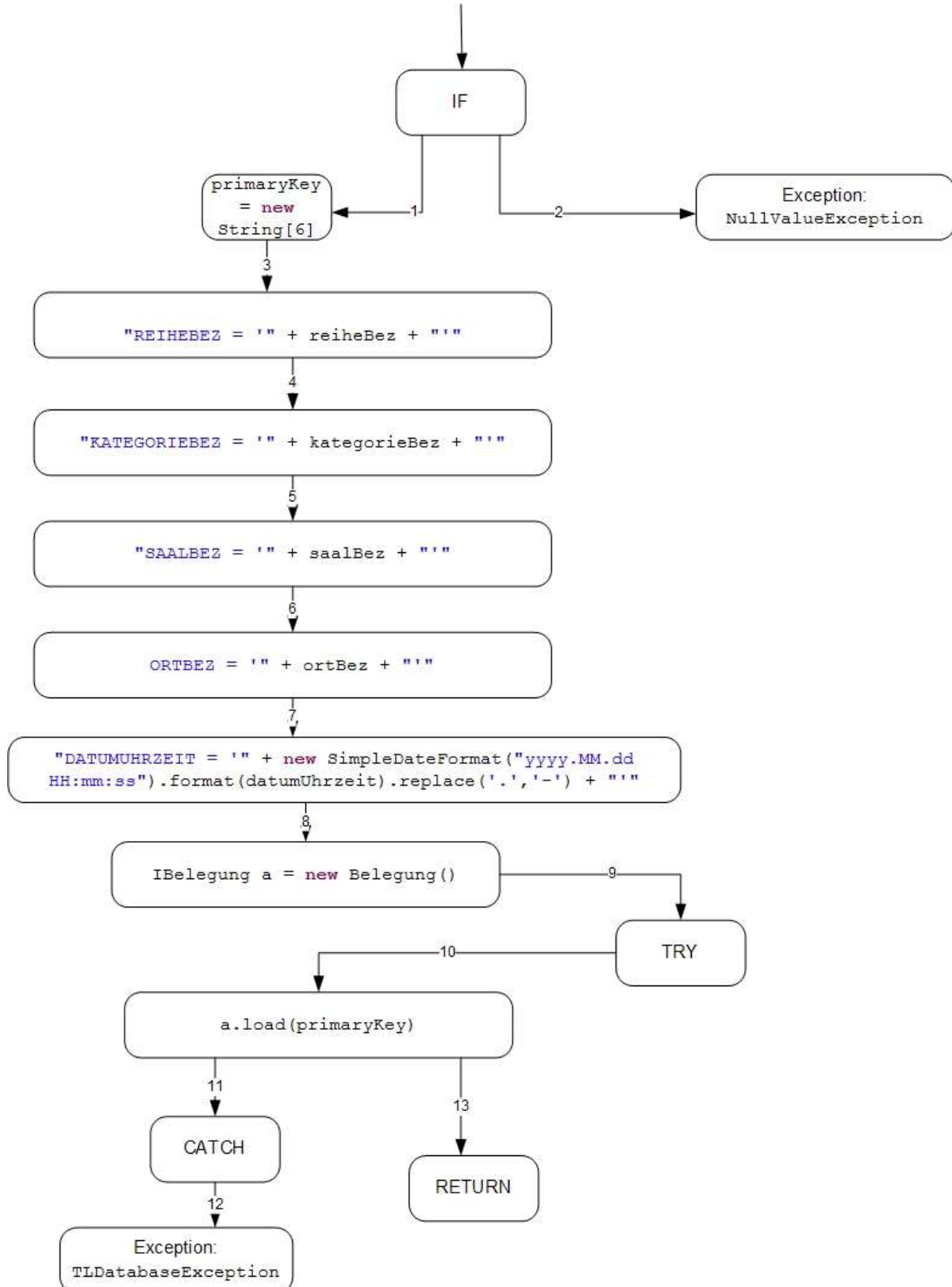
5.2 Klasse SearchObject

5.2.1 Methode getSearchResult

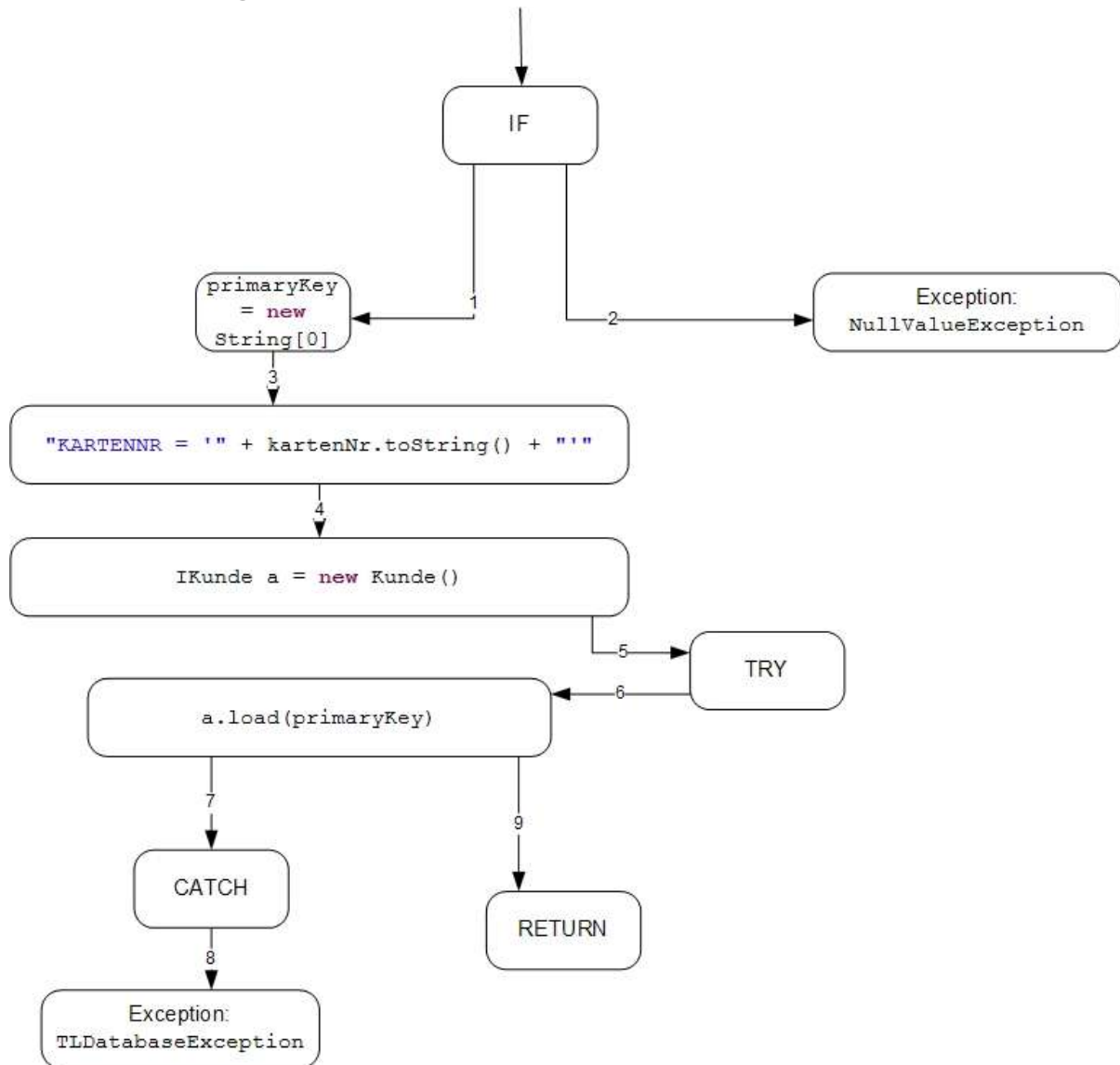


5.3 Klasse Proxy

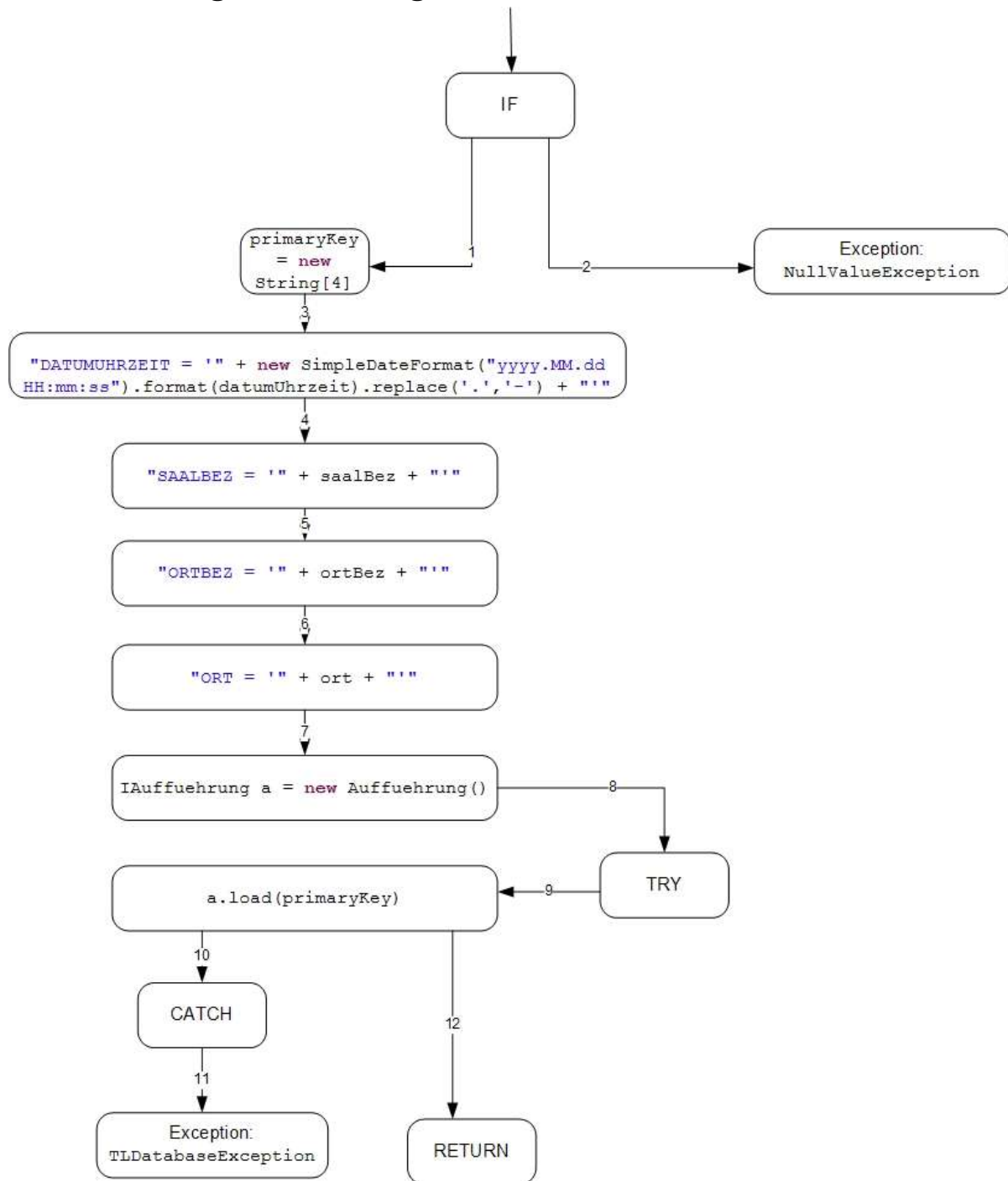
5.3.1 Methode getBelegung



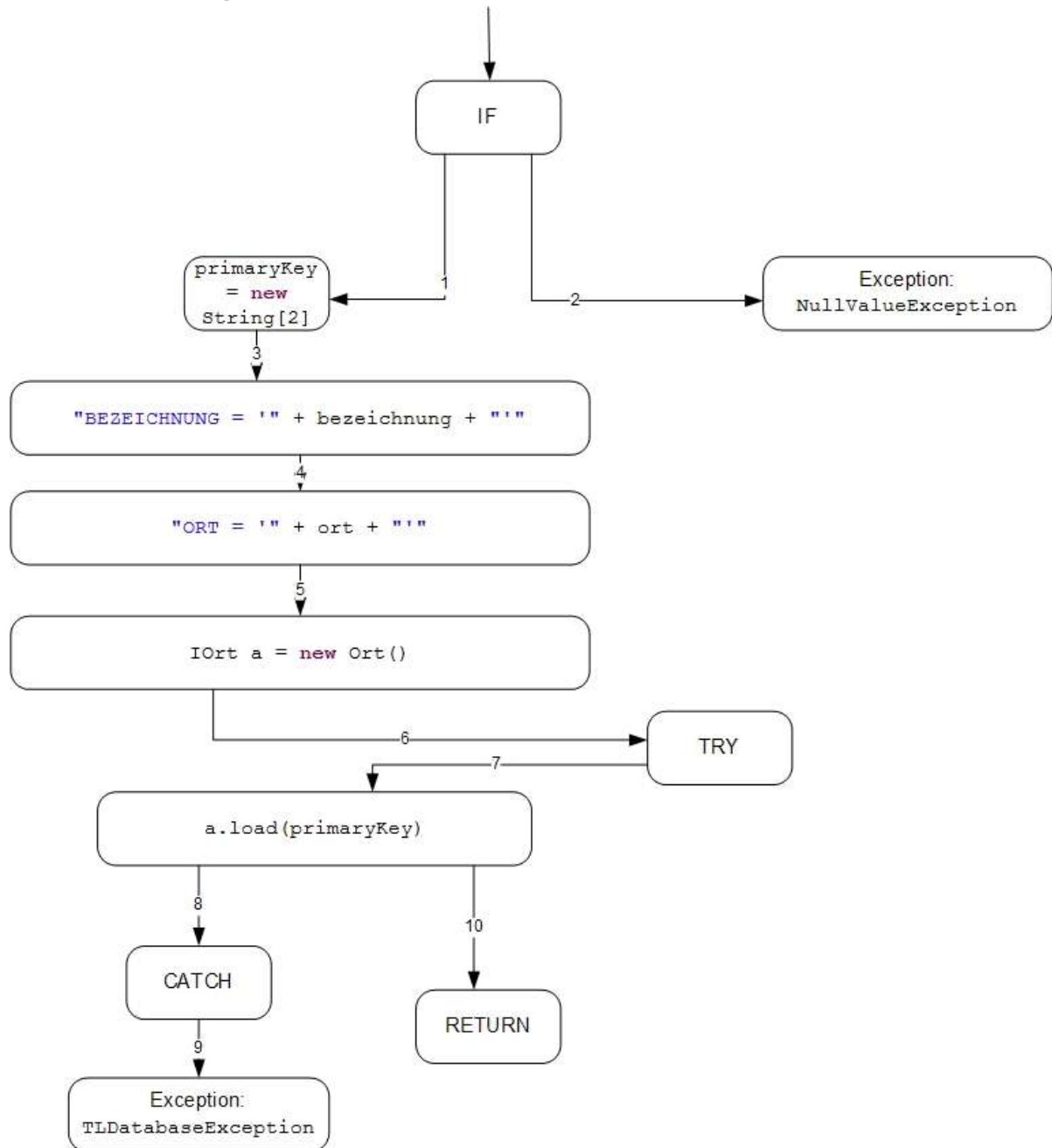
5.3.2 Methode getKunde



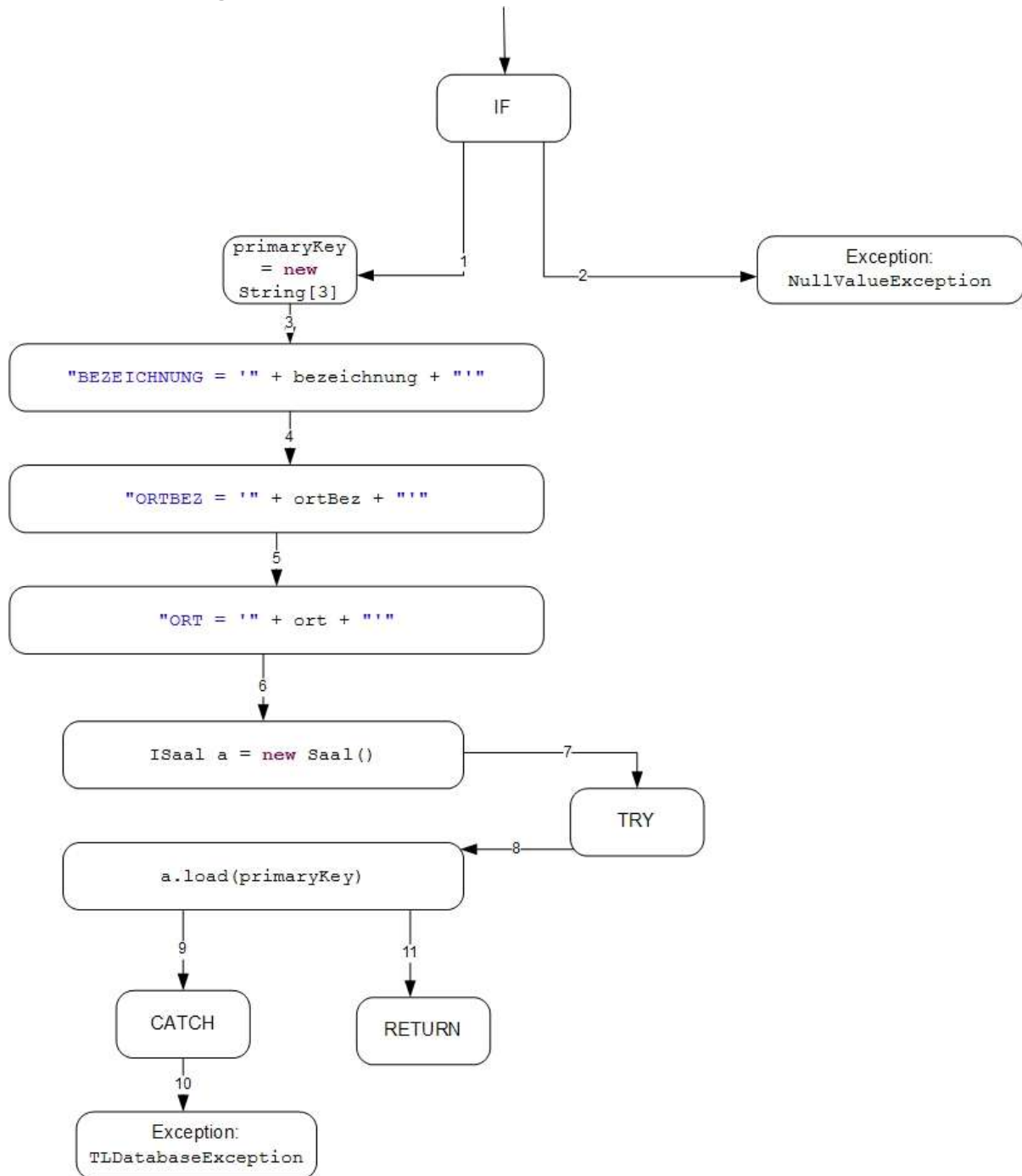
5.3.3 Methode getAuffuehrung



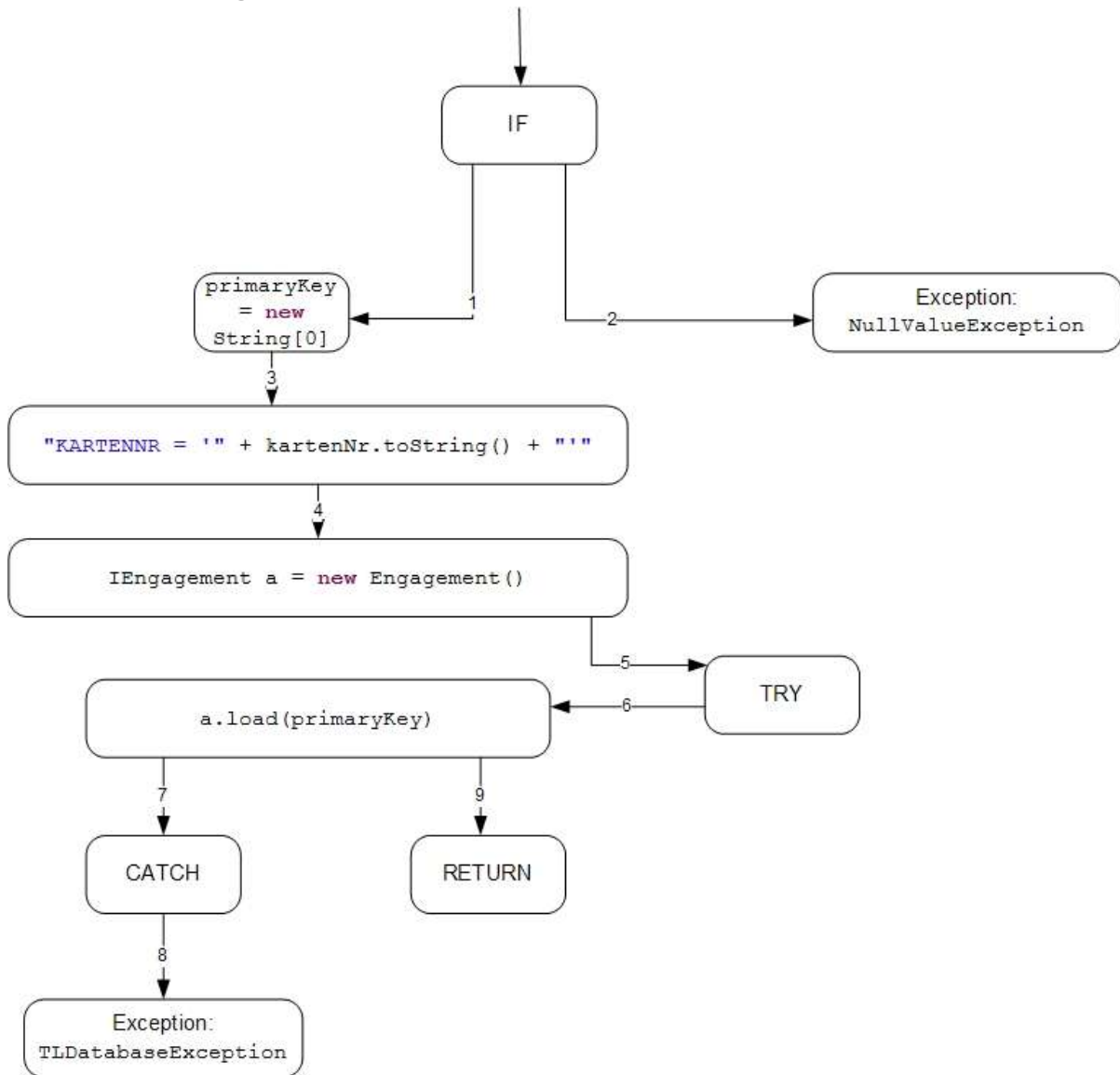
5.3.4 Methode getOrt



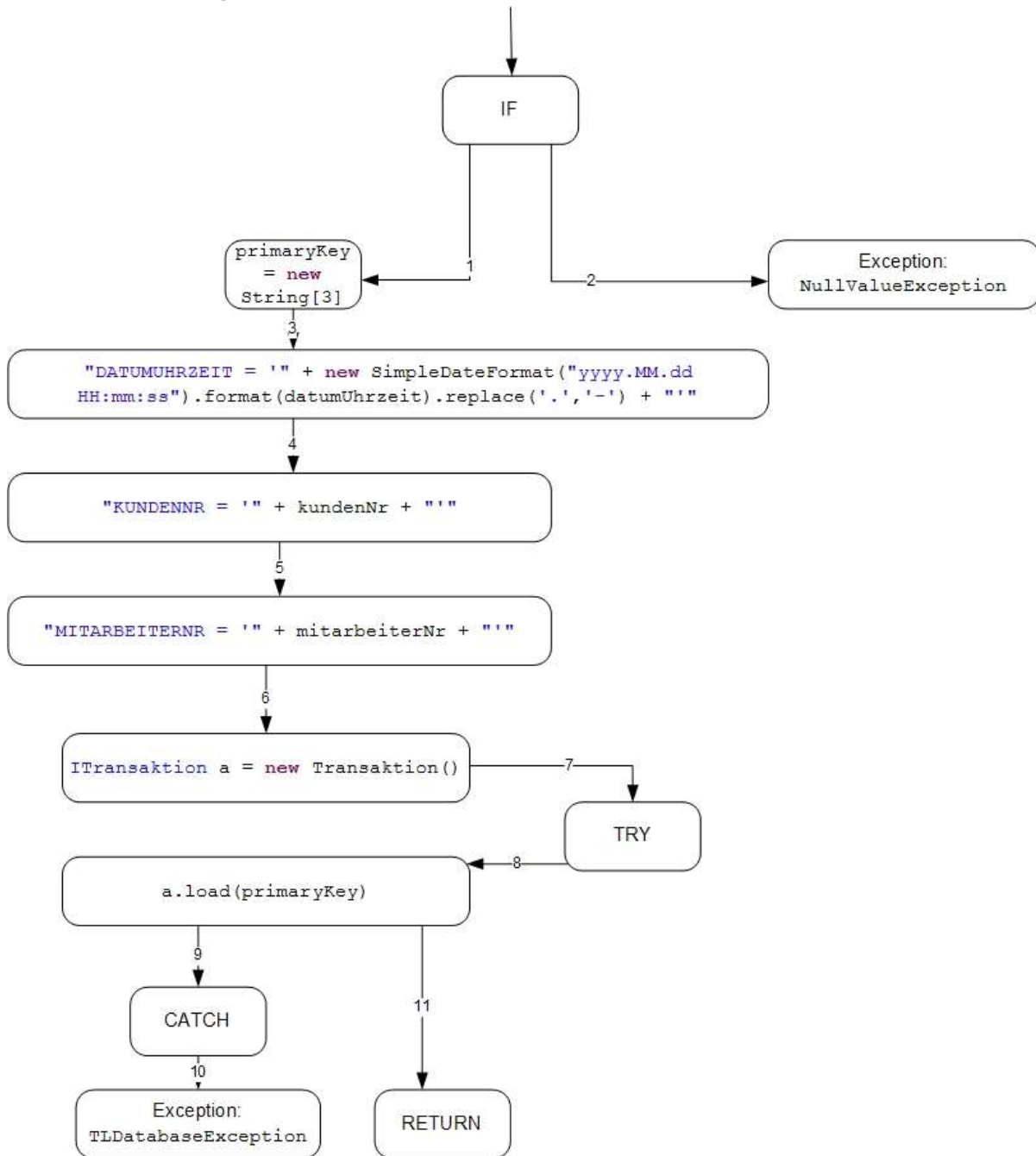
5.3.5 Methode getSaal



5.3.6 Methode getMitarbeiter



5.3.7 Methode getTransaktion



6 Testfälle

In diesem Abschnitt werden die Testfälle anhand der Kontrollflussdiagramme erstellt. Diese sind in Form von Tabellen, welche denen von Black Box Testen ähneln, dargestellt. Die entsprechenden Kontrollflussdiagramme finden sich unter den korrespondierenden Punkten der Hauptgliederung → 5.

6.1 Klasse PostgresqlProcessor

6.1.1 Methode CommitSelect

Nr.	Klasse	Beschreibung des Testfalls	Erwartetes Ergebnis	Eingabedaten	Testergebnis
1	FF	Ein Attribut des übergebenen SQL Statements ist in der Entität nicht vorhanden	Fehlermeldung	sqlStatement=''	'
Überprüfte Kanten: 1,2,3,4,5,6					
2	NF	Gültiges INSERT Statement	Daten werden gespeichert	INSERT mit einem Datensatz der noch nicht in der Datenbank vorhanden ist.	
Überprüfte Kanten: 1,2,3,4,5,6,7,8					
3	NF	Gültiges SELECT Statement	Daten werden geladen und im Resultset zurückgeliefert	SELECT mit einem Datensatz welcher in der Datenbank vorhanden ist.	
Überprüfte Kanten: 1,2,3,4,5,6,7,8					
4	FF	Verbindung zur Datenbank fehlerhaft oder nicht vorhanden	Fehlermeldung		
Überprüfte Kanten: 1,2,12,10					

6.1.2 Methode ToString

Nr.	Klasse	Beschreibung des Testfalls	Erwartetes Ergebnis	Eingabedaten	Testergebnis
1	FF	Fehler bei der Umwandlung in einen Strings	Fehlermeldung		
Überprüfte Kanten: 1,2,4,11					
2	SF	Nur das Datum wird gefragt	Nur das Datum wird in DB gespeichert	Nur das Datum wird ausgewählt.	
Überprüfte Kanten: 1,2,3,6					
3	NF	Kombination gültiger Werte	Umwandlung in einen String erfolgt		
Überprüfte Kanten: 1,2,3,5,7,8,10					

6.2 Klasse SearchObject

6.2.1 Methode getSearchResult

Nr.	Klasse	Beschreibung des Testfalls	Erwartetes Ergebnis	Eingabedaten	Testergebnis
1	FF	Übergabe eines invaliden Tupels	Fehlermeldung	Ein leeres Tupel übergeben.	
Überprüfte Kanten: 1,2,3,6,8					
3	NF	Kombination gültiger Werte	Umwandlung in einen String erfolgt		
Überprüfte Kanten: 1,2,3,4,7,8,5					

6.3 Klasse Proxy

6.3.1 Methode getBelegung

Nr.	Klasse	Beschreibung des Testfalls	Erwartetes Ergebnis	Eingabedaten	Testergebnis
1	NF	Es wurde kein Objekt gefunden, das den Suchkriterien entspricht	Hinweismeldung	Ein nicht vorhandenes Objekt übergeben	
Überprüfte Kanten: 1,3,4,5,6,7,8,9,10-,11,12					
2	FF	Die Belegung kann nicht gesucht werden, da ein übergebener Wert leer ist	Fehlermeldung		
Überprüfte Kanten: 2					
3	NF	Kombination gültiger Werte	Belegungsobjekt wird geliefert		
Überprüfte Kanten: 1,3,4,5,6,7,8,9,10-,13					

6.3.2 Methode getKunde

Nr.	Klasse	Beschreibung des Testfalls	Erwartetes Ergebnis	Eingabedaten	Testergebnis
1	NF	Es wurde kein Objekt gefunden, das den Suchkriterien entspricht	Hinweismeldung	Ein nicht vorhandenes Objekt übergeben	
Überprüfte Kanten: 1,3,4,5,6,7,8					
2	FF	Der Kunde kann nicht gesucht werden, da ein übergebener Wert leer ist	Fehlermeldung		
Überprüfte Kanten: 2					
3	NF	Kombination gültiger Werte	Kundeobjekt wird geliefert		
Überprüfte Kanten: 1,3,4,5,6,9					

6.3.3 Methode getAuffuehrung

Nr.	Klasse	Beschreibung des Testfalls	Erwartetes Ergebnis	Eingabedaten	Testergebnis
1	NF	Es wurde kein Objekt gefunden, das den Suchkriterien entspricht	Hinweismeldung	Ein nicht vorhandenes Objekt übergeben	
Überprüfte Kanten: 1,3,4,5,6,7,8,9,10,11					
2	FF	Die Aufführung kann nicht gesucht werden, da ein übergebener Wert leer ist	Fehlermeldung		
Überprüfte Kanten: 2					
3	NF	Kombination gültiger Werte	Aufführungsobjekt wird geliefert		
Überprüfte Kanten: 1,3,4,5,6,9,12					

6.3.4 Methode getOrt

Nr.	Klasse	Beschreibung des Testfalls	Erwartetes Ergebnis	Eingabedaten	Testergebnis
1	NF	Es wurde kein Objekt gefunden, das den Suchkriterien entspricht	Hinweismeldung	Ein nicht vorhandenes Objekt übergeben	
Überprüfte Kanten: 1,3,4,5,6,7,8,9					
2	FF	Der Ort kann nicht gesucht werden, da ein übergebener Wert leer ist	Fehlermeldung		
Überprüfte Kanten: 2					
3	NF	Kombination gültiger Werte	Ortobjekt wird geliefert		
Überprüfte Kanten: 1,3,4,5,6,9,10					

6.3.5 Methode getSaal

Nr.	Klasse	Beschreibung des Testfalls	Erwartetes Ergebnis	Eingabedaten	Testergebnis
1	NF	Es wurde kein Objekt gefunden, das den Suchkriterien entspricht	Hinweismeldung	Ein nicht vorhandenes Objekt übergeben	
Überprüfte Kanten: 1,3,4,5,6,7,8,9,10					
2	FF	Der Saal kann nicht gesucht werden, da ein übergebener Wert leer ist	Fehlermeldung		
Überprüfte Kanten: 2					
3	NF	Kombination gültiger Werte	Saalobjekt wird geliefert		
Überprüfte Kanten: 1,3,4,5,6,7,8,11					

6.3.6 Methode getMitarbeiter

Nr.	Klasse	Beschreibung des Testfalls	Erwartetes Ergebnis	Eingabedaten	Testergebnis
1	NF	Es wurde kein Objekt gefunden, das den Suchkriterien entspricht	Hinweismeldung	Ein nicht vorhandenes Objekt übergeben	
Überprüfte Kanten: 1,3,4,5,6,7,8					
2	FF	Der Mitarbeiter kann nicht gesucht werden, da ein übergebener Wert leer ist	Fehlermeldung		
Überprüfte Kanten: 2					
3	NF	Kombination gültiger Werte	Mitarbeiterobjekt wird geliefert		
Überprüfte Kanten: 1,3,4,5,6,9					

6.3.7 Methode getTransaktion

Nr.	Klasse	Beschreibung des Testfalls	Erwartetes Ergebnis	Eingabedaten	Testergebnis
1	NF	Es wurde kein Objekt gefunden, das den Suchkriterien entspricht	Hinweismeldung	Ein vorhandenes Objekt übergeben	nicht
Überprüfte Kanten: 1,3,4,5,6,7,8,9,10					
2	FF	Die Transaktion kann nicht gesucht werden, da ein übergebener Wert leer ist	Fehlermeldung		
Überprüfte Kanten: 2					
3	NF	Kombination gültiger Werte	Transaktionsobjekt wird geliefert		
Überprüfte Kanten: 1,3,4,5,6,7,8,11					

7 Literatur

Zuser, Wolfgang: *Software Engineering mit UML und dem Unified Process*; München: Paerson Education Deutschland. (insbesondere Kapitel 7 Arbeitsschnitt Test)