# Aufgabe 1

## Gegeben

### Klasse

|  |  |
| --- | --- |
|  | public class PersonRepository { |
|  |  |
|  | private Properties databaseProperties; |
|  |  |
|  | public PersonRepository(final Properties databaseProperties) { |
|  | this.databaseProperties = databaseProperties; |
|  | } |
|  |  |
|  | public Optional<Person> findPersonByPersonNo(int personNo) |
|  | throws RepositoryException { |
|  | try (Connection connection = createConnection()) { |
|  | final String sql = "SELECT \* FROM PERSON WHERE PERSONNO = ?"; |
|  | final PreparedStatement statement = connection.prepareStatement(sql); |
|  | statement.setInt(1, personNo); |
|  | final ResultSet rs = statement.executeQuery(); |
|  | rs.next(); |
|  | final Person person = mapResultSetToPerson(rs); |
|  | return Optional.of(person); |
|  | } catch (SQLException e) { |
|  | throw new RepositoryException(e.getMessage(), e); |
|  | } |
|  | } |
|  |  |
|  | private Connection createConnection() throws SQLException { |
|  | final String url = databaseProperties.getProperty("url"); |
|  | final String user = databaseProperties.getProperty("user"); |
|  | final String password = databaseProperties.getProperty("password"); |
|  | return DriverManager.getConnection(url, user, password); |
|  | } |
|  |  |
|  | private Person mapToPerson(final ResultSet resultSet) |
|  | throws SQLException { |
|  | final Person person = new Person(); |
|  | person.setId(resultSet.getInt("ID")); |
|  | person.setPersonNo(resultSet.getInt("PERSONNO")); |
|  | person.setFirstname(resultSet.getString("FIRSTNAME")); |
|  | person.setLastname(resultSet.getString("LASTNAME")); |
|  | return person; |
|  | } |
|  | } |

### Testklasse

public class PersonRepositoryIT {

private PersonRepository repository;

@Before

public void setup() {

final String dbUrl = Paths.get("derbyEmbeddedDatabase")

.toAbsolutePath().toString();

final Properties databaseProperties = new Properties();

databaseProperties.setProperty("driver",

"org.apache.derby.jdbc.EmbeddedDriver");

databaseProperties.setProperty("url", "jdbc:derby:" + dbUrl);

databaseProperties.setProperty("user", "APP");

databaseProperties.setProperty("password", "APP");

repository = new PersonRepository(databaseProperties);

}

@Test

public void shouldBeNotFindPersonByPersonNo() throws RepositoryException {

// when

Optional<Person> optPerson = repository.findPersonByPersonNo(888888);

// then

assertThat(optPerson.isPresent(), is(false));

}

}

Weiter ist gegeben, dass die im Test verwendete Datenbank exisitiert. Ebenfalls existieren alle notwendigen Tabellen, Spalten, Berechtigungen und Testdaten.

## Frage

Wird dieser JUnit-Test "Grün"? Wenn nicht, was muss in der Klasse 'PersonRepository' geändert werden, damit der Test korrekt durchläuft?

|  |  |
| --- | --- |
| Zeilen-Nr. | Java-Code |
| 16 | if (!rs.next()) { |
| 16a | return Optional.empty(); |
| 16b | } |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |