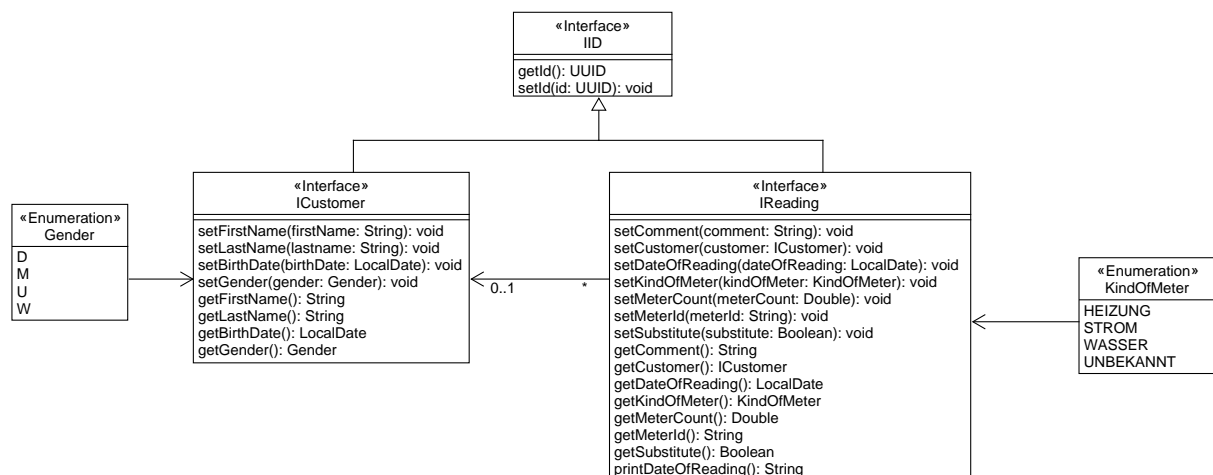


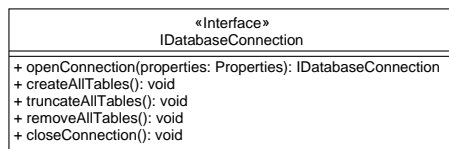
## Sprint 1: Datenbankanbindung der digitalen Hausverwaltung

**Abgabe:** Die vollständige Lösung zu dieser Aufgabe ist bis zum ???.2024 in Ihr Repository einzupflegen.

In diesem Sprint geht es darum, für die Speicherung der Daten einer digitalen Hausverwaltung eine entsprechende Datenbank bereitzustellen und mit Java entsprechende CRUD-Funktionalität umzusetzen. Die entsprechenden Datenhaltungsklassen für die Kunden- und Ablesungsdaten, die in der Datenbank gespeichert werden sollen, sollen folgende Interfaces implementieren:



Für die Datenbankverbindung soll dabei folgendes Interface umgesetzt werden:



Die Informationen zur Datenbankverbindung (URL, Benutzer, Passwort) sollen in einer Properties-Datei gespeichert werden. Diese soll folgenden Aufbau besitzen:

```
[Systemnutzer].db.url=...
[Systemnutzer].db.user=...
[Systemnutzer].db.pw=...
```

Hinweis: [Systemnutzer] soll dabei durch den aktuellen Systemnutzer ersetzt werden. Sie können den Systemnutzer in Java mit `System.getProperty("user.name")` auslesen.

Diese Datei soll sich entweder im Home-Verzeichnis des Systemnutzers (auslesbar mit `System.getProperty("user.home")`) oder im Classpath des Projekts (z.B. unter `src/main/resources`) befinden.

**Für den erfolgreichen Abschluss dieses Sprints sind folgende Punkte zu realisieren.**

- Erstellen Sie ein entsprechendes Repository auf GitHub (**Name: {Klasse}-{Gruppe}**) und fügen Sie alle Teammitglieder zu diesem Repository hinzu. Fügen sie außerdem den Lehrer-Account mit der E-Mailadresse **fael2@bsinfo.eu** hinzu, so dass dieser Account Ihr Repository jederzeit vollumfänglich anschauen bzw. auschecken kann.
- Pflegen Sie die vorgegebenen Interfaces in ein Maven-Projekt ein und pflegen Sie dieses Projekt in Ihr Git Repository ein.
- Die Dokumentation Ihres Projektfortschritts, insbesondere wer im Team welche Aufgaben umsetzt, muss über **Issues** auf GitHub erfolgen. Erstellen Sie dazu für diesen Sprint einen Meilenstein „**Sprint 1: Datenbankankbindung**“ und ordnen Sie alle Issues für diesen Sprint diesem Meilenstein zu.
- Erstellen Sie eine entsprechende Datenbank zur Speicherung der Kunden- bzw. Ablesungsdaten. Dabei ist folgendes DBMS zu verwenden: MariaDB (mindestens Version 11.3)
- Realisieren Sie anschließend den CRUD-Zugriff auf die Kunden- bzw. Ablesungsdaten in der Datenbank. Verwenden Sie dabei für den Zugriff auf die Datenbank das Framework JDBC.  
Achten Sie für eine erfolgreiche Implementierung auf folgende Punkte:
  - Ihre Anwendung darf zu keiner Zeit durch Exceptions abstürzen.
  - Zu keiner Zeit dürfen inkonsistente Daten in der Datenbank vorliegen.
  - Wird ein Kunde gelöscht, sollen dessen Ablesungen trotzdem gespeichert bleiben. Die Kundenreferenz soll bei den entsprechenden Ablesungen auf null gesetzt werden.
  - Es darf keine Ablesung ohne Kunde in die Datenbank eingefügt werden.
  - Wird versucht eine Ablesung mit einem Kunden, der nicht in der Datenbank existiert, einzufügen, so soll der Kunde in der Datenbank angelegt werden und anschließend die Ablesung eingefügt werden.
- Erstellen Sie für Ihre Anwendung mithilfe von JUnit passende **Integrationstests**, so dass Sie für dieses Projekt eine Zweigabdeckung von mind. 80% erreichen. Hinweis: Die JUnit-Tests selbst sollen für die Zweigabdeckung nicht berücksichtigt werden. Automatisch generierter Code (z.B. durch Eclipse) darf für die Zweigabdeckung ignoriert werden.  
Die Tests sollen mit dem Befehl „mvn test“ ausgeführt werden können. Außerdem soll eine Übersicht über die Testabdeckung als HTML-Seite möglich sein, die mit dem Befehl „mvn clean install site -P test“ erstellt werden kann. (siehe Skript zu Maven)
- Jedes Teammitglied muss dabei zusätzlich die Skripte zu den folgenden Themen bearbeitet haben: JDBC, Maven-Projekte, Arbeiten mit git und Testen mit JUnit.