Geoinformationsdaten

Die **Mondial-Datenbank** enthält geographische Informationen über Städte, Länder, Berge, Flüsse und vieles mehr. Dieses Projekt soll die Datenbank bedienerfreundlich aufbereiteen.

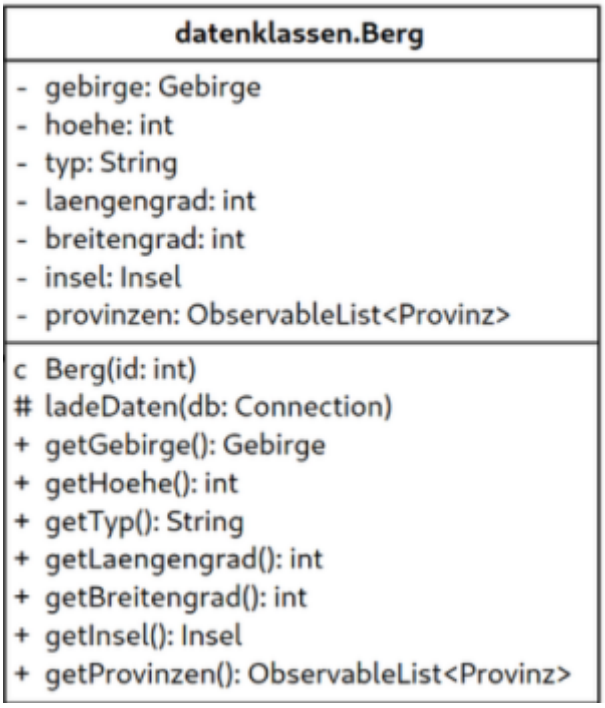
Dazu entwickeln wir eine Anwendung, mit der man die Informationen zu geographischen Objekten anzeigen lassen kann und die es ermöglicht, zu verknüpften Objekten zu springen.

Das BlueJ-Projekt enthält eine Vorlage.

Passe die Verbindungsinformationen in der Klasse `Helper` im Paket `datenklassen` an, so dass eine Verbindung zur SQLite-Datenbank `mondial.db` hergestellt werden kann, die sich bereits im Unterordner `datenklassen` befindet.

Die Datenbank selbst und die Beziehungen zwischen den Tabellen werden durch das Diagramm in der Datei `Mondial.html` im Hauptverzeichnis des Projekts beschrieben.

Jedes Objekt in der Datenbank wird durch ein entsprechendes Java-Objekt repräsentiert- im folgenden Beispiel der Typ Berg. Das Java-Objekt enthält alle Informationen zum Datenbank-Objekt und bietet Methoden zum Auslesen an.



Alle Klassen erben von der abstrakten Klasse `MondialObjekt`. Der Konstruktor jeder Unterklasse von `MondialObjekt` erhält als einzigen Parameter die ID des Objekts als int. Alle weiteren Informationen zum Objekt werden in der Methode `ladeDaten(Connection db)` geladen.

Der gegebene Programmrumpf kümmert sich bereits darum, dass die Objekte aus der Datenbank geladen werden.

Deine Aufgabe ist es, die Implementation der Methode `ladeDaten(…)` zu vervollständigen. Sorge dafür, dass alle Attribute des Java-Objekts mit den entsprechenden Werten aus der Datenbank belegt werden. Du kannst in der Klasse `Berg` nachschauen, wie das aussehen kann, diese ist bereits fertig

implementiert.

Zusammenarbeit mit Git

Da es sich bei dieser Aufgabe um ein relativ aufwendiges Projekt handelt, bietet es sich an, dass mehrere Personen im Team zu arbeiten.

Verwende dazu die Teasmarbeitsfunktion von BlueJ:

- Lade die Projektvorlage herunter. Entferne, wenn du das Repo geklont hast den Unterordner `.git`, wenn du das Projekt aus einem zipo-Archiv entpackt hast, musst du nichts weiter tun.
- Öffne das Projekt in BlueJ, kompliliete alle Dateien, auch im Unterordner.
- "Teile" das Projekt an ein Repository, auf das alle Teammitglider schreiben können.
- Die anderen Teammitglieder können sich das Projekt dort als "Arbeitskopie" wieder abholen.
- Stimmt euch ab, wer welche Objekttypen bearbeitet. Nimm dann nur an den Klassen Änderungen vor, die "die gehören".
- Erstzelle in der Teamwork-Funktion von BlueJ Commits und teile sie auf das gemeinsame Repo, um die Ergebnisse zusammenzuführen.

Hinweise

Die einzelnen Objekttypen unterscheiden sich in ihrem Aufwand bzw. Schwierigkeitsgrad. Wählen Sie einen (oder mehrere) Objekttyp, der deiner Arbeitsgeschwindigkeit entspricht.

- Hoher Aufwand: Land, Fluss, Provinz, Stadt, See
- Mittlerer Aufwand: Meer, Insel, Wueste, Organisation, Gebirge
- Geringer Aufwand: Kontinent, Sprache, Religion, Volksgruppe, Inselgruppe

Ergänzung

Nun soll noch eine Volltextsuche durchgeführt werden, mit der man z.B. alle Objekte finden kann, deren Namen die Zeichenkette "ing" enthält. Implementiere dazu die Methode `sucheObjekte()` in der Klasse `Helper` im Unterordner `datenklassen`.