

Sehr wichtig: Die Online-Befragung zur gewählten alternativen Veranstaltungsform ist online. Bitte ausfüllen: <https://forms.gle/inYQDjQyABVkuVcp9> (Die eventuell von Google geforderte Anmeldung wäre nur notwendig, wenn Sie die Befragung zwischenspeichern wollten.)

Aufgabe 26 (3 Punkte)

Schreiben Sie ein Java-Programm mit dem alle in Mondial mit Längen- und Breitengrad eingetragenen Städte eines Landes in einer Karte eingezeichnet werden. In dem von der Veranstaltungsseite erhältlichen Projekt ist bereits eine Eingabezeile realisiert, in die der Name eines Landes eintragen und Return gedrückt werden kann. Ihre Aufgabe besteht darin, eine Klasse zu schreiben, die das Interface `karte.Karte` realisiert. Achten Sie darauf,

- dass alle Städte angezeigt werden mit ihrer Position,
- dass das Programm bei nicht existierenden Ländern oder Ländern bei denen die Gradangaben bei allen Städten fehlen, nicht abstürzt,
- dass es eine sinnvolle Ausgabe gibt, wenn nur eine Stadt eingetragen ist.
- dass der Platz zur Ausgabe ausgenutzt wird, also Länder so vollständig den Platz der Anzeige nutzen (auf Breite und Höhe verteilen).

Zum Testen eignen sich u.a.

Andorra und Germany. Die rechte Seite zeigt eine Beispielausgabe für Australia. Ausgaben dürfen sich überlappen. Das Projekt nutzt vereinfachend keine Module.

Die Karte wird mit Hilfe der Klasse `Interaktionsbrett` angezeigt, über die man sich auf der Seite <http://kleuker.iui.hs-osnabrueck.de/querschnittlich/Interaktionsbrett/index.html> informieren kann. Falls es Startprobleme mit dem Projekt gibt, bitte Ende des Aufgabenblatts beachten.



Aufgabe 27 (5 Punkte)

- a) Legen Sie eine Tabelle der folgenden Form in einer neuen Datenbank an.

```
CREATE TABLE Studierend(  
  Matrnr INTEGER,  
  Name VARCHAR(16),  
  Semester VARCHAR(6),  
  CONSTRAINT PK_Student  
  PRIMARY KEY(matrnr)  
);
```

- b) Schreiben Sie basierend auf dem Teilprojekt von der Veranstaltungswebseite ein Java-Programm in der Klasse `dialog.Studierendenbearbeitung`, mit dem man

- das Programm beenden
- Studierende ergänzen (scheitert, wenn Matrikelnummer vorhanden)
- alle Studierenden ausgeben
- die Namen der Studierenden ändern
- alle Studierenden löschen

kann. Der folgende Nutzungsdialog soll z. B. möglich sein, Eingaben sind umrandet.
Das System soll melden, ob die Änderung durchgeführt wurde oder nicht.

Was wollen Sie?
(0) Programm beenden
(1) neuen Studi hinzufuegen
(2) alle Studis zeigen
(3) Namen eines Studi aendern
(4) alle Studis loeschen
 2
42: Ute (WiSe19)
43: Uwe (WiSe19)
44: Urs (SoSe20)

Was wollen Sie?
(0) Programm beenden
(1) neuen Studi hinzufuegen
(2) alle Studis zeigen
(3) Namen eines Studi aendern
(4) alle Studis loeschen
 4

Was wollen Sie?
(0) Programm beenden
(1) neuen Studi hinzufuegen
(2) alle Studis zeigen
(3) Namen eines Studi aendern
(4) alle Studis loeschen
 2

Was wollen Sie?
(0) Programm beenden
(1) neuen Studi hinzufuegen
(2) alle Studis zeigen
(3) Namen eines Studi aendern
(4) alle Studis loeschen
 1
Name:
Matrikelnummer:
Semester:
erfolgreich hinzugefuegt

Was wollen Sie?
(0) Programm beenden
(1) neuen Studi hinzufuegen
(2) alle Studis zeigen
(3) Namen eines Studi aendern
(4) alle Studis loeschen
 2
72: Marion (WiSe19)

Was wollen Sie?
(0) Programm beenden

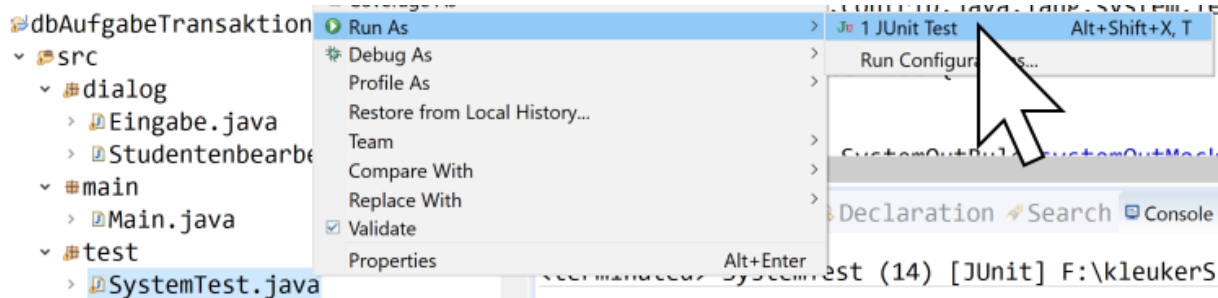
(1) neuen Studi hinzufuegen
(2) alle Studis zeigen
(3) Namen eines Studi aendern
(4) alle Studis loeschen
 3
Matrikelnummer:
neuer Name:
nicht geaendert

Was wollen Sie?
(0) Programm beenden
(1) neuen Studi hinzufuegen
(2) alle Studis zeigen
(3) Namen eines Studi aendern
(4) alle Studis loeschen
 3
Matrikelnummer:
neuer Name:
erfolgreich geaendert

Was wollen Sie?
(0) Programm beenden
(1) neuen Studi hinzufuegen
(2) alle Studis zeigen
(3) Namen eines Studi aendern
(4) alle Studis loeschen
 2
72: Susanne (WiSe19)

Was wollen Sie?
(0) Programm beenden
(1) neuen Studi hinzufuegen
(2) alle Studis zeigen
(3) Namen eines Studi aendern
(4) alle Studis loeschen
 1
Name:
Matrikelnummer:
Semester:
nicht hinzugefuegt

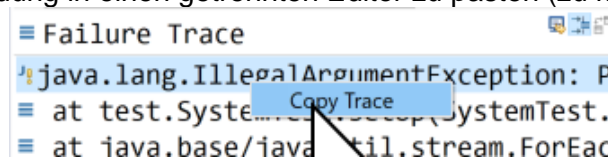
Was wollen Sie?
(0) Programm beenden
(1) neuen Studi hinzufuegen
(2) alle Studis zeigen
(3) Namen eines Studi aendern
(4) alle Studis loeschen
 0



Da es hier um die Funktionsweise geht, können Sie auf ergonomische Prüfungen von Fehleingaben verzichten und bei einer SQLException diese einfach fangen, das Programm soll aber nicht abstürzen. Erzeugen Sie *genau* die im Beispieldialog gezeigten Ausgaben, z. B. Matrikelnr: `_Name_(Semester)`, da es sonst Probleme mit den Tests geben kann.

Nutzen Sie zum Testen des Programms die Klasse `SystemTest` (die unabhängig von den folgenden Teilaufgaben nutzbar sein sollte). Nutzen Sie zunächst die Standardtransaktionseinstellungen von Java (Erinnerung: `autoCommit` ist `true`).

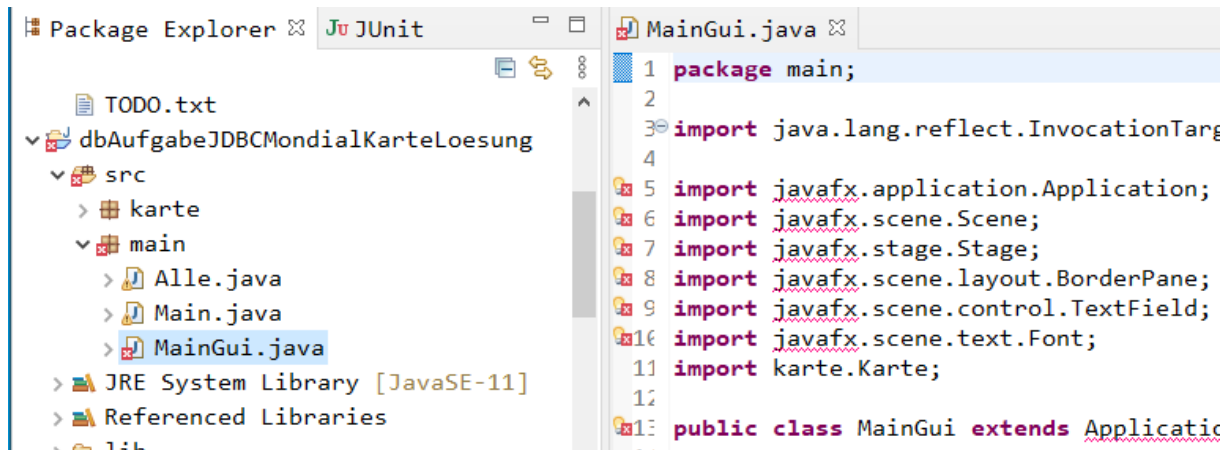
Hinweis: Um sich JUnit-Meldungen genauer anzusehen, ist es sinnvoll, einen Rechtsklick auf dem Failure Trace zu machen, „Copy Trace“ auszuwählen und die genaue Fehlermeldung in einen getrennten Editor zu pasten (zu kopieren, `Strg+V`).



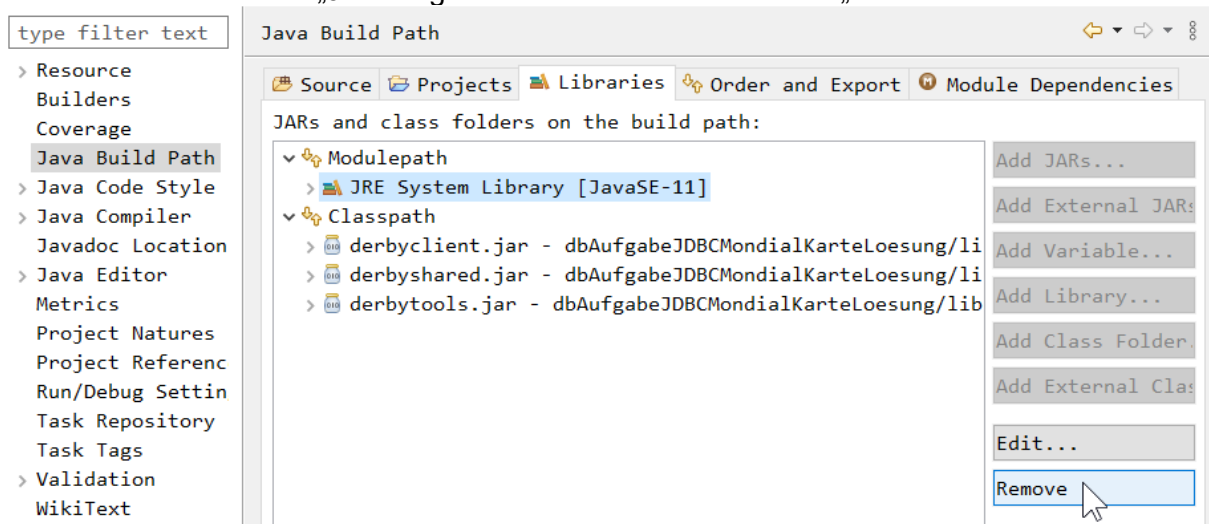
- c) Führen Sie Ihr Programm gleichzeitig mehrfach in verschiedenen Eclipse-Fenstern aus und notieren Sie, was passiert, wenn Sie sich bei der ersten Nutzung und dann bei der zweiten Nutzung sich alle Studierenden zuerst anschauen (Menüpunkt 2) und danach fast gleichzeitig den Namen einer studierenden Person ändern (erster Nutzung beginnt Änderung, zweite Nutzung beginnt Änderung, erste Nutzung schließt Änderung ab, zweite Nutzung schließt Änderung ab)?
- d) Ändern Sie Ihr Programm so ab, dass Sie für das `Connection`-Objekt `con` unmittelbar nach seiner Erzeugung die Methode `this.con.setAutoCommit(false)` und erst bei der Beendigung des Programms die Methode `this.con.commit()` aufrufen. Damit finden hier nur zum Experimentieren und für die Praxis sehr ungewöhnlich alle Schritte im Programm in einer einzigen Transaktion statt. Führen Sie die Analyse aus c) für folgende Situationen durch, wenn unmittelbar nach `this.con.setAutoCommit(false)` jeweils folgende Zeile ergänzt wird. Beachten Sie, dass vor einem neuen Experiment alle Programme wieder geschlossen sind und das Programm neu aufgerufen wird.
- `this.con.setTransactionIsolation(Connection.TRANSACTION_READ_COMMITTED);`
 - `this.con.setTransactionIsolation(Connection.TRANSACTION_REPEATABLE_READ);`
 - `this.con.setTransactionIsolation(Connection.TRANSACTION_SERIALIZABLE);`
- Dokumentieren Sie jeweils Ihre Beobachtungen. Falls es so aussieht, dass das System nicht reagiert (warum?), warten Sie jeweils eine Minute.

mögliche Problembehebung:

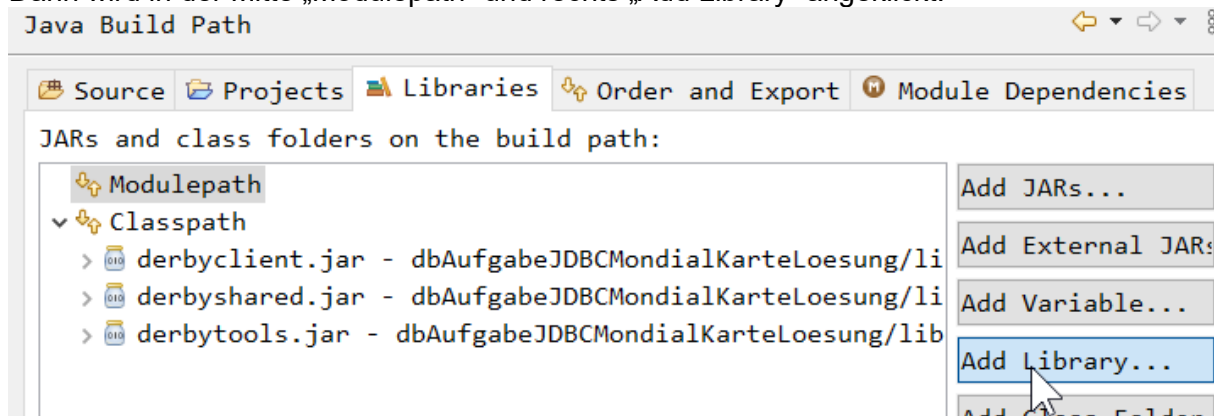
Bei der Nutzung der KleukerSEU ist die Oberflächenbibliothek JavaFX (OpenJFX) bereits mit eingebunden. Gegebenenfalls kann es trotzdem passieren, dass sie nicht gefunden wird, was zu folgenden Fehlermeldungen in `MainGui` führt.



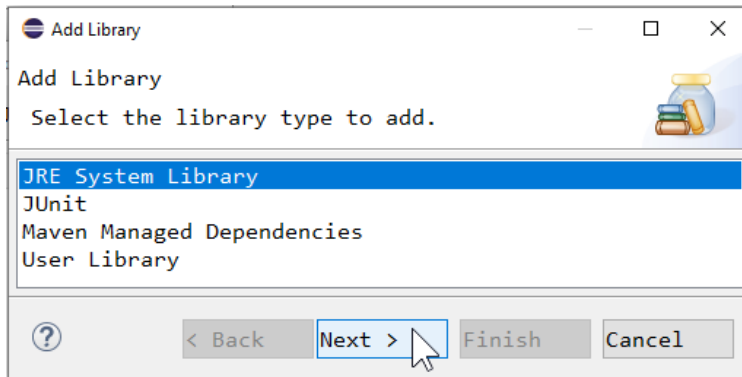
Dies deutet an, dass eine andere Java-Version als erwartet gerade genutzt wird. Diese Version muss jetzt neu eingestellt werden. Dazu wird nach einem Rechtsklick auf dem Projekt unten „Properties“ ausgewählt, rechts auf „Java Build Path“ und oben auf „Libraries“ geklickt. Zunächst wird dann auf „JRE...“ geklickt und dieses rechts über „Remove“ entfernt.



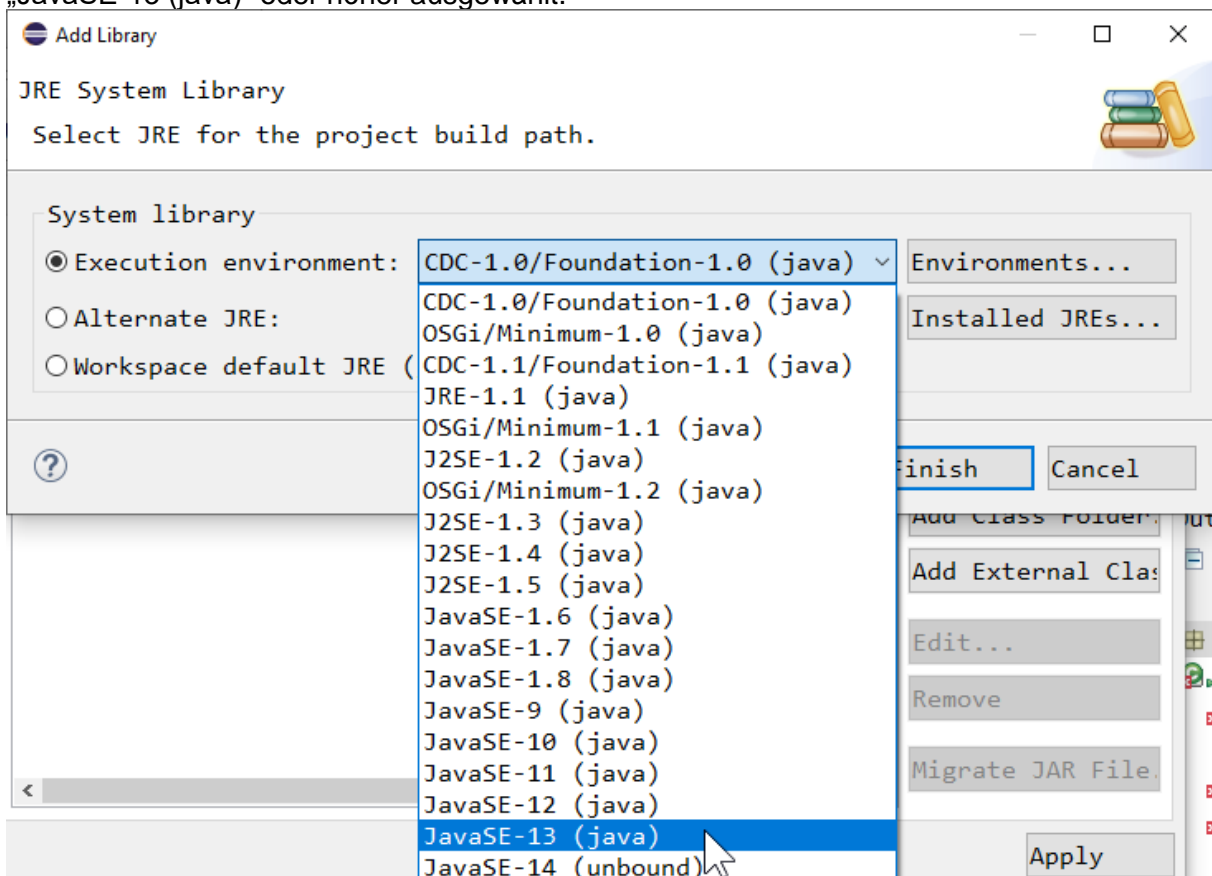
Dann wird in der Mitte „Modulepath“ und rechts „Add Library“ angeklickt.



Es wird oben „JRE System Library“ ausgewählt und „Next >“ geklickt.



Es wird oben „Execution environment“ und dann in der Drop-Down-Box rechts daneben „JavaSE-13 (java)“ oder höher ausgewählt.



Die Aktion wird mit „Finish“ und „Apply and Close“ abgeschlossen. Danach sollten alle Fehlermeldungen verschwunden sein.

Sollte es weiterhin Probleme in der KleukerSEU geben, melden Sie sich bei Prof. Dr. Kleuker (E-Mail, Vorlesungssprechstunde).