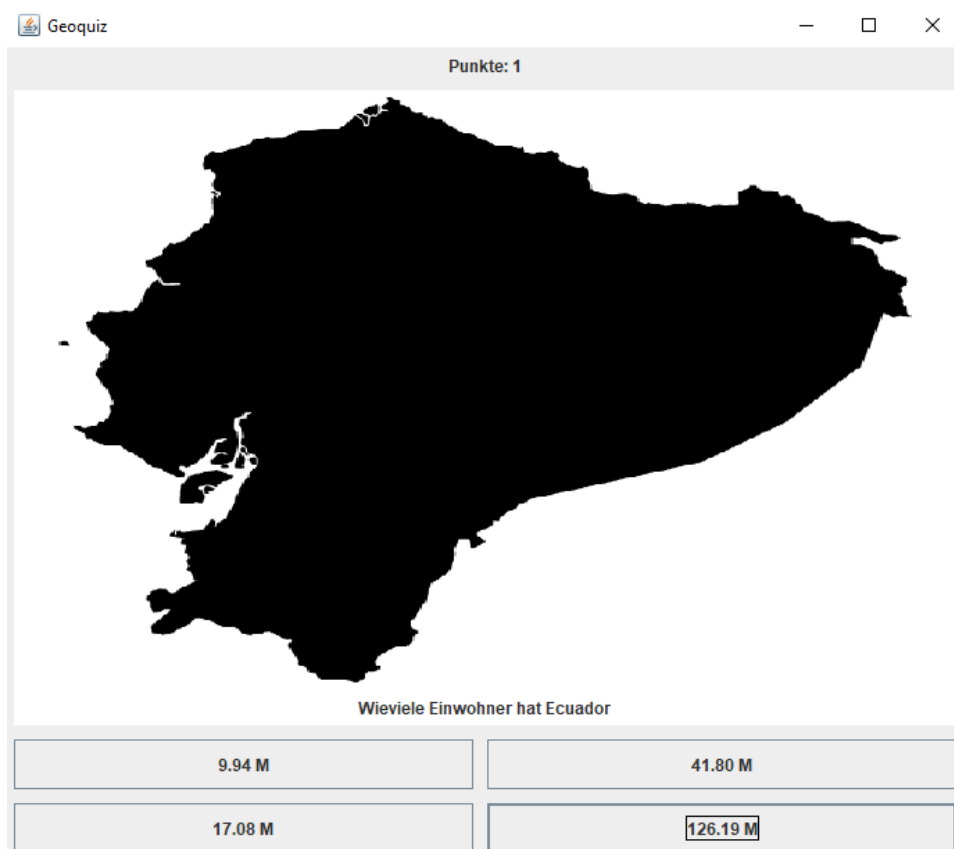


OOP2 PRÜFUNG FS 2021

Bedingungen:

- Erlaubte Hilfsmittel: Unterrichtsunterlagen, Java Buch und Übungen.
- Die Prüfung ist schrittweise, gemäss Aufgabenstellung lokal auf Ihrem Computer zu lösen. Kopieren sie zu diesem Zwecke den gesamten Ordner *P_FS21_Geoquiz_Vorlage* auf Ihre lokale Harddisk und importieren sie das Projekt in Eclipse.
- Am Ende der Prüfung ist der Ordner *src* umbenannt in *NameVorname* gezippt per Email abzugeben.
- Setzen sie als erstes Ihren Namen und Vornamen in die Dateien, die Sie bearbeiten.
- Gegenseitiges Abschreiben in irgendeiner Form führt zur Note 1!
- Folgend sie bei der Wahl von Variablen und der User-Interface Positionierung exakt den Angaben in der Aufgabenstellung.

Beschreibung:



Ziel dieser Prüfung ist es, ein Geoquiz zu programmieren. Die Quizfragen werden aus einer csv-Datei zeilenweise eingelesen. Es gibt drei unterschiedliche Fragetypen, die von der abstrakten Klasse **QuizQuestion** abgeleitet werden. Die Klasse **OutlineQuestion** fragt nach Umrissen von Ländern, **PopulationQuestion** nach Bevölkerungszahlen und **FlagQuestion** nach der Nationalflagge. Die Fragetypen werden zufällig ausgewählt.

Durch Auswahl eines der vier möglichen Lösungsbuttons wird bei richtiger Lösung ein Punkt erzielt. Die Optionen der Antworten werden als Text im Button angezeigt. Das Geoquiz basiert auf dem

MVC Entwurfsmuster. Das Model sorgt für die Logik der Fragen und Antwortgenerierung und lädt die Daten zu den Ländern aus der Datei, die dann in der Klasse **Country** verwaltet werden. Die View beinhaltet alle JLabels, JPanels und JButtons zur Darstellung der Landkarten, Flaggen, Fragen, Antworten und Punktzahl. Die genaue Spezifikation kann der Beilage entnommen werden. Der Controller als Bindeglied zwischen View und Model ruft die notwendigen Model Methoden auf. Die detaillierte Programmstruktur kann dem Klassendiagramm in der Beilage entnommen werden.

Die Aufgabenstellung führt Sie schrittweise zum Ziel. Folgen Sie daher beim Lösen der Prüfung der Aufgabenstellung.

Achten Sie darauf, dass Ihr Code kompilierbar bleibt!

Viel Erfolg!

Aufgabe 1: Klasse View Methode *init*

Die View soll im Gridbag-Layout realisiert werden. Neben den JButtons und dem JLabel für die Punkte ist ein JPanel **panelQuestion** eingebettet, das wiederum im Gridbag-Layout das ImagePanel **imageQuestion** und das JLabel **labelQuestion** hält. Implementieren Sie die Methode ***init***, um das GUI zu erstellen. Layout und Verhalten bei Grössenänderung kann der Beilage entnommen werden. Setzen Sie für die JPanels den passenden LayoutManager.

Sie sollten dann das Layout mit den Default-Labels sehen. Das ImagePanel ist noch leer und wird erst in der ***update*** Methode sichtbar.

Beim Click der Buttons soll jeweils ***actionPerformed*** von **View** aufgerufen werden.

Aufgabe 2: Klasse View - Methode *update* und *actionPerformed*

Die ***actionPerformed*** Methode leitet die Events an den Controller weiter. Die Methode ***update*** holt aus dem Model die aktualisierten Fragen, Bilder und Punktestand. Implementieren Sie die Methoden gemäss Kommentar im Code.

Aufgabe 3: Klasse Controller

Die Methode ***setView*** kümmert sich neben der obligatorischen Bekanntmachung der view im **Controller** um das Laden der Länder und Generierung der Frage. Implementieren Sie die Klasse **Controller** gemäss Kommentare im Code.

Aufgabe 4: Klasse Model Methode *loadCountries*

In der Klasse Model sind die Methoden zur Generierung und Prüfung der Frage. Ferner wird mit der Methode ***loadCountries*** das csv-file mit den Ländern und deren Attribute geladen. Die csv-Datei mit den Ländern befindet sich im Ordner *res/* zu dessen Pfad Sie mittels **`"/res/"`** gelangen.

Implementieren Sie zunächst die Methode ***loadCountries***. Der Speicher ist passend für die Anzahl Länder allokiert. Pro Zeile sind der Name und Attribute komma-separiert abgespeichert:

Land,Code,Population,Area

z.B.

Switzerland,CH,8513227,41284.0

Lesen Sie die Datei zeilenweise als String ein und schreiben Sie die Attribute in die passenden Country Attribute des allokierten countries[233] Arrays.

Testen können Sie die Methode mit dem mitgelieferten Unit-Test **CountryTest**.

Aufgabe 5: Klasse Model

Implementieren Sie die restlichen Methoden *generateQuestion* und *answered* der Klasse Model nach Kommentaren im Code.

Hinweis:

Die Methode *RandomUtil.randomDistinct*(int amount, int minInclusiveValue, int maxExclusiveValue) liefert ein int Array der Länge amount mit Zufallszahlen im Intervall [minInclusiveValue, maxExclusiveValue[.

Aufgabe 6: Klasse PopulationQuestion

In der Klasse **PopulationQuestion** gibt es die Methode *formatPop(int population)*, die genutzt werden soll, um die grossen Zahlen lesbarer zu machen.

Ist:

| Wieviele Einwohner hat Guinea-Bissau | |
|--------------------------------------|--------------------------------------|
| <input type="text" value="1411"/> | 41801533 |
| 1874309 | <input type="text" value="3726549"/> |

Sorgen Sie dafür, dass die Zahlen in Einheiten M für grösser 1Million Einwohner, K für grösser 1000 Einwohner mit jeweils 2 Nachkommastelle als String zurückgegeben werden. Unterhalb 1000 Einwohner soll die Zahl wie bisher direkt ohne Umrechnung zurückgegeben werden.

Soll:

| Wieviele Einwohner hat Antarctica | |
|-----------------------------------|--------------------------------------|
| 1.11 k | <input type="text" value="38.43 M"/> |
| 96.76 k | 1.39 M |