

Prüfung Programming, Automation, Computation

Programming: Visualisierung mit Google Code Playground (5 Punkte)

Wir haben im Unterricht verschiedene Visualisierungen im Google Code Playground angeschaut. Betrachten wir hier eine Visualisierung von Daten aus einer externen Datenquelle. Wir verwenden die Tabelle mit den Anzahl Milliardären pro Land, die wir im Unterricht verwendet haben.

Gegeben sind folgende Daten (URL der Daten <https://docs.google.com/spreadsheet/pub?key=0AjfyeQOvnd95dGNRXzVtUDlWU2d5NkJNaWxlTlowdEE&output=csv>):

	Jahr 2004	Jahr 2005	Jahr 2006	Jahr 2007
United States	277	341	371	415
Germany	52	57	55	55
Russia	25	27	33	53
India	9	12	23	36
United Kingdom	17	24	24	29
Turkey	6	8	21	25
Japan	22	24	27	24
Canada	17	17	22	23
Hong Kong, China	14	15	17	21
Brazil	6	8	16	20

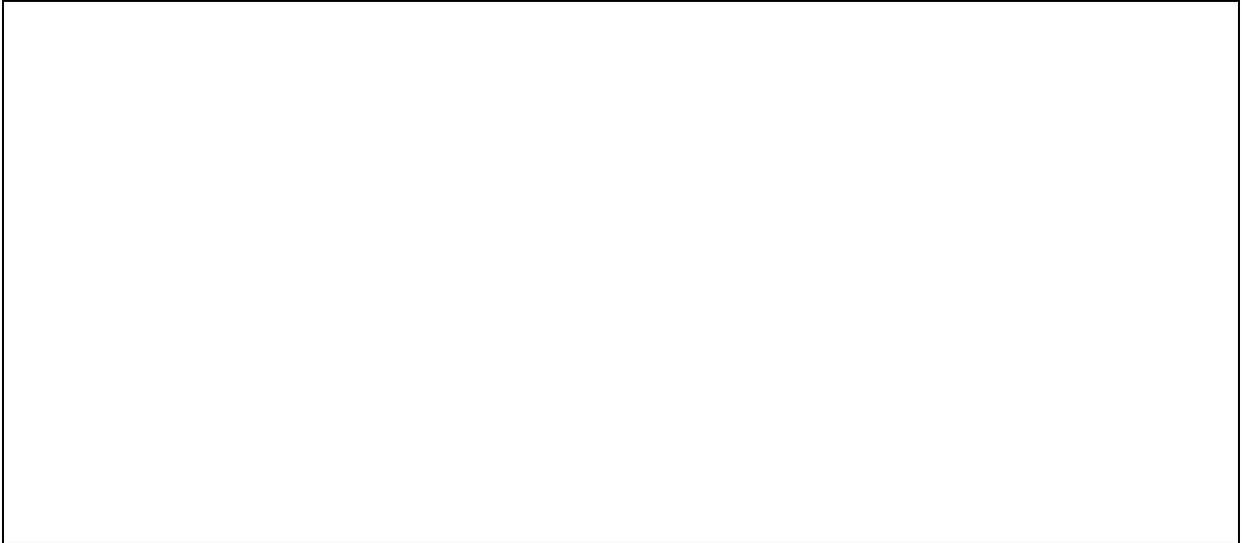
Betrachten wir dazu folgendes Programm:

```
function drawVisualization() {
  var query = new google.visualization.Query('https://docs.google.com/spreadsheet/pub?key=0AjfyeQOvnd95dGNRXzVtUDlWU2d5NkJNaWxlTlowdEE&output=csv');
  query.setQuery('SELECT A,E WHERE (E > 25) AND (E < 100) ORDER BY E DESC');
  query.send(handleQueryResponse);
}

function handleQueryResponse(response) {
  var data = response.getDataTable();
  visualization = new google.visualization.BarChart(document.getElementById(vis));
  visualization.draw(data, {legend: 'bottom'});
}
```

Aufgabe 1: Programmausführung simulieren

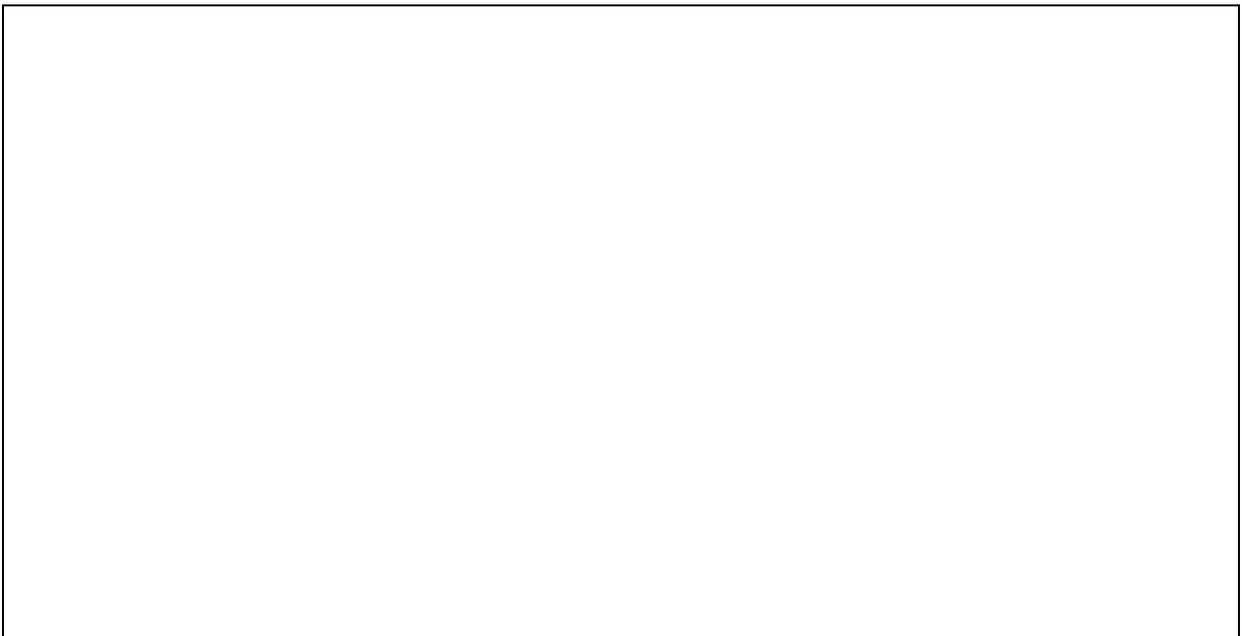
Skizzieren Sie, wie das durch dieses Programm erstellte Balkendiagramm aussieht. Das Diagramm soll qualitativ stimmen; die exakten Grössenverhältnisse sind nicht ausschlaggebend. Markieren Sie zudem in der Tabelle auf der vorangehenden Seite, welche Zahlen Sie verwenden:



Aufgabe 2: Schnittstellen erklären

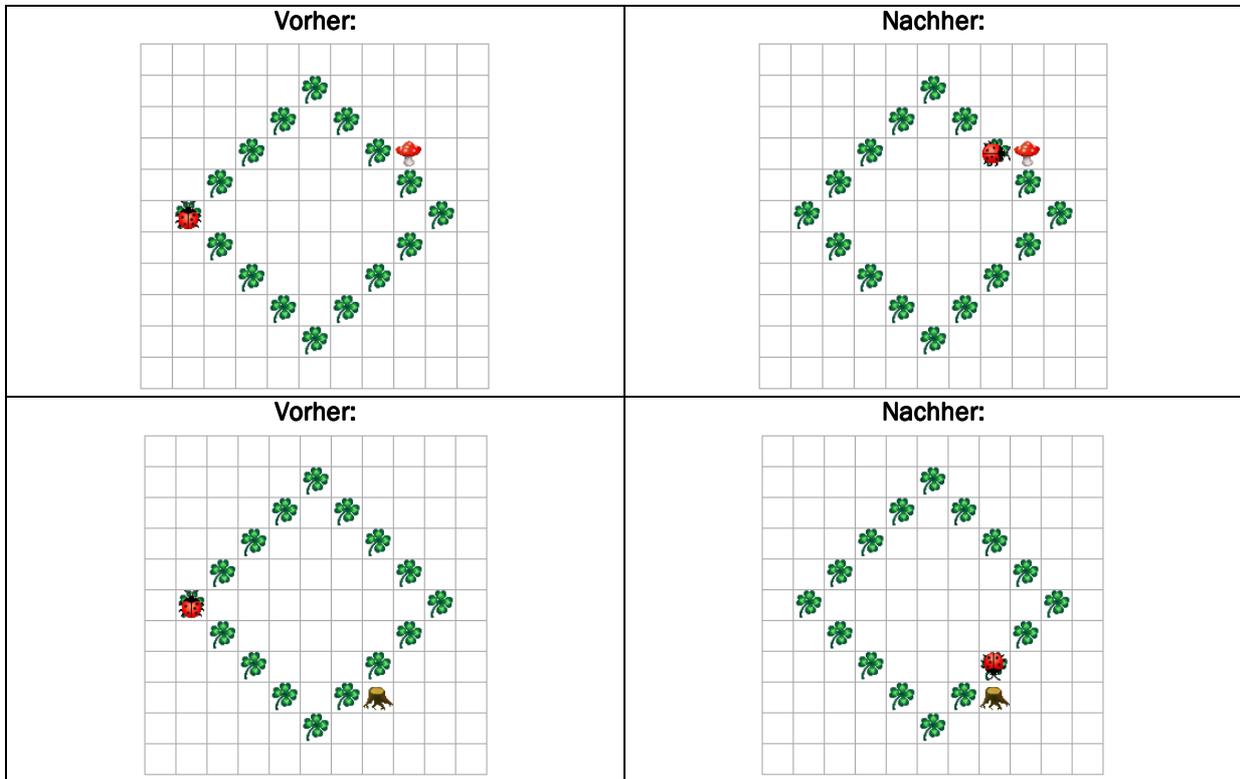
Wir haben im Unterricht das Konzept der Schnittstellen kennen gelernt. Das Alltagsbeispiel waren Büchsen und Büchsenöffner; in Google Code Playground haben wir Google Maps angeschaut und gesehen, wie über eine kleine, wohldefinierte Schnittstelle Google uns mächtige Funktionalität zur Verfügung stellt.

Erklären Sie in wenigen Sätzen einem Kollegen, der Google Code Playground und das Konzept der Schnittstellen nicht kennt, wie es möglich ist, dass das Programm mit wenigen Zeilen Daten holen und visualisieren kann. Was muss das Programm angeben; was macht Google dann daraus? Sie können durchaus eine Analogie zu Büchsen oder anderen Alltagsbeispielen machen.



Automation: Schreiben Sie einen Automaten für Kara (5 Punkte)

Schreiben Sie einen Automaten, der Kara **auf einem Kleeblatt-Quadrat** (wie in der Abbildung links gezeigt) laufen lässt. Kara soll aufhören, wenn er auf einem Kleeblatt steht und zudem vor ihm ein Pilz oder ein Baum steht (oder endlos laufen, falls es weder Baum noch Pilz hat).



Hinweise

- Kara startet immer auf einem Kleeblatt, wie links dargestellt.
- Kara ist es gleichgültig, wie gross das Quadrat ist.
- Die **Pilze und Bäume** sind **nur direkt angrenzend an den Kleeblättern**, wie in den Abbildungen dargestellt. Kara braucht sich also zum Beispiel keine Gedanken zu machen, ob etwa auf einem Kleeblatt ein Pilz steht.

Aufgabe 1: Schreiben Sie Ihren Automaten in die Zustandstabelle unten

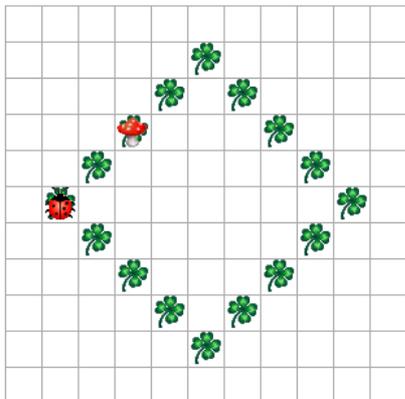
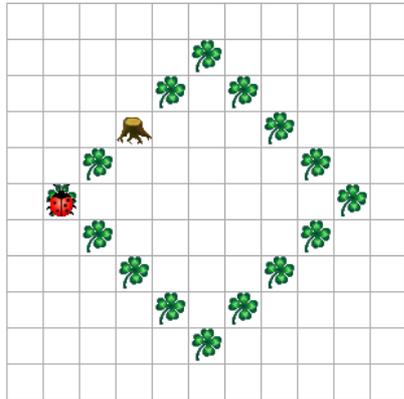
- Ein Zustand mit den drei angegebenen Sensoren reicht für eine Lösung aus.
- Machen Sie kenntlich, welche Sensorwerte Sie für einen Übergang erwarten.
- Geben Sie die Befehle an, die Kara ausführen soll.
- Notieren Sie den jeweils nächsten Zustand.
- Sollte es zu viele Zeilen haben, können Sie die nicht-benötigten Zeilen ebenfalls streichen.

			Kara macht:	Nächster Zustand:
<input checked="" type="checkbox"/>	<input type="checkbox"/> yes <input type="checkbox"/> or <input type="checkbox"/> no	<input type="checkbox"/> yes <input type="checkbox"/> or <input type="checkbox"/> no	<input type="checkbox"/> yes <input type="checkbox"/> or <input type="checkbox"/> no	<input type="text"/>
<input checked="" type="checkbox"/>	<input type="checkbox"/> yes <input type="checkbox"/> or <input type="checkbox"/> no	<input type="checkbox"/> yes <input type="checkbox"/> or <input type="checkbox"/> no	<input type="checkbox"/> yes <input type="checkbox"/> or <input type="checkbox"/> no	<input type="text"/>
<input checked="" type="checkbox"/>	<input type="checkbox"/> yes <input type="checkbox"/> or <input type="checkbox"/> no	<input type="checkbox"/> yes <input type="checkbox"/> or <input type="checkbox"/> no	<input type="checkbox"/> yes <input type="checkbox"/> or <input type="checkbox"/> no	<input type="text"/>
<input checked="" type="checkbox"/>	<input type="checkbox"/> yes <input type="checkbox"/> or <input type="checkbox"/> no	<input type="checkbox"/> yes <input type="checkbox"/> or <input type="checkbox"/> no	<input type="checkbox"/> yes <input type="checkbox"/> or <input type="checkbox"/> no	<input type="text"/>
<input checked="" type="checkbox"/>	<input type="checkbox"/> yes <input type="checkbox"/> or <input type="checkbox"/> no	<input type="checkbox"/> yes <input type="checkbox"/> or <input type="checkbox"/> no	<input type="checkbox"/> yes <input type="checkbox"/> or <input type="checkbox"/> no	<input type="text"/>
<input checked="" type="checkbox"/>	<input type="checkbox"/> yes <input type="checkbox"/> or <input type="checkbox"/> no	<input type="checkbox"/> yes <input type="checkbox"/> or <input type="checkbox"/> no	<input type="checkbox"/> yes <input type="checkbox"/> or <input type="checkbox"/> no	<input type="text"/>
<input checked="" type="checkbox"/>	<input type="checkbox"/> yes <input type="checkbox"/> or <input type="checkbox"/> no	<input type="checkbox"/> yes <input type="checkbox"/> or <input type="checkbox"/> no	<input type="checkbox"/> yes <input type="checkbox"/> or <input type="checkbox"/> no	<input type="text"/>
<input checked="" type="checkbox"/>	<input type="checkbox"/> yes <input type="checkbox"/> or <input type="checkbox"/> no	<input type="checkbox"/> yes <input type="checkbox"/> or <input type="checkbox"/> no	<input type="checkbox"/> yes <input type="checkbox"/> or <input type="checkbox"/> no	<input type="text"/>

Aufgabe 2: Analysieren Sie Ihr Programm

Analysieren Sie Ihr Programm: Was passiert, wenn die Welt nicht wie vorgegeben aussieht?

Beschreiben Sie kurz, was bei der Programmausführung geschieht, und begründen Sie Ihre Antwort kurz.



Automation: Modellieren mit Automaten (5 Punkte)

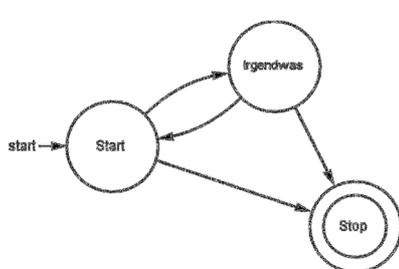
Ein Telekommunikationsunternehmen braucht Ihre Unterstützung. Der Prozess für die Bestellung eines neuen Festnetzanschlusses liegt nur in textueller Form vor. Ihre Aufgabe: Den Prozess als endlichen Automaten zu visualisieren. Der Prozess ist wie folgt beschrieben:

Wenn ein Kunde einen Festnetzanschluss bestellen möchte, muss er zunächst die Adresse eingeben, wo der Anschluss gewünscht ist. Anschliessend wählt der Kunde zwischen folgenden Produkten: Analoges Anschluss (ein einzelner Anschluss) oder digitaler Anschluss (ISDN, mehrere Anschlüsse). Gleichzeitig kann der Kunde angeben, ob er zusätzlich Internet bestellen möchte. Dann gelten folgende Regeln:

- Wenn der Kunde einen analogen Anschluss ohne Internet bestellt, so muss er den Anschluss selber installieren. Er muss dazu das Datum wählen, an dem der Anschluss aktiviert wird.
- Wenn der Kunden einen analogen Anschluss und Internet bestellt, so kann er wählen, ob jemand vom Telekommunikationsunternehmen das Internet installieren soll. Installiert der Internet Kunde selbst, muss er das Datum wählen, an dem Anschluss und Internet aktiviert werden; lässt er jemanden kommen, muss er kein Datum wählen.
- Wenn der Kunde einen digitalen Anschluss bestellt, muss er zwingend einen Elektriker für die Installation bestellen. Der Elektriker installiert den Anschluss und allenfalls Internet. Um Aktivierungsdatum kümmert sich der Elektriker.

Wenn die obigen Fragen geklärt sind, muss der Kunde entweder eine Telefonnummer wählen (falls er einen analogen Anschluss bestellt) oder mehrere Telefonnummern (falls er einen digitalen Anschluss bestellt). Anschliessend kommt in beiden Fällen eine Zusammenfassung mit allen Angaben, die der Kunde im Prozess gemacht hat.

Modellieren Sie die beschriebenen Abläufe als endlichen Automaten:

<ul style="list-style-type: none">▪ Geben Sie Start- und Endzustand an.▪ Überlegen Sie sich, welche Zustände es braucht. Geben Sie den Zuständen sprechende Bezeichnungen, damit klar wird, welchem Schritt im Ablauf der Zustand entspricht.▪ Die Übergänge brauchen Sie nicht zu beschriften.	<p>Die Abbildung zeigt ein Beispiel, wie ein Automat gezeichnet wird:</p>  <pre>graph LR; Start((Start)) --> Irgendwas((Irgendwas)); Irgendwas --> Start; Irgendwas --> Stop(((Stop))); start((start)) --> Start;</pre>
---	---

Zeichnen Sie hier Ihren Automaten für die Abläufe im Kundendienst:

A large, empty rectangular box with a thin black border, intended for drawing an automaton (state machine) for customer service processes. The box is currently blank.

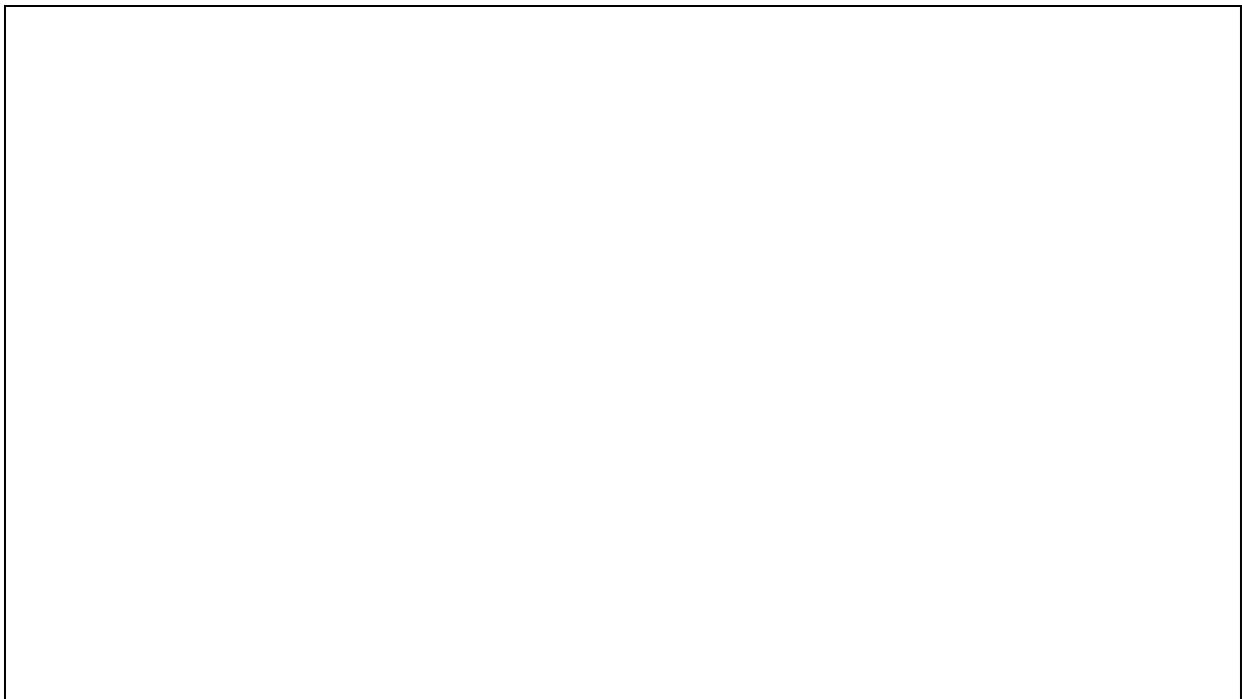
Computation: Reguläre Ausdrücke in endliche Automaten überführen (5 Punkte)

Erstellen Sie einen Automaten für das Programm Exorciser, der dem folgenden regulären Ausdruck entspricht:

$a(baa^* | a(aa)^*)b$

Hinweis: Betrachten Sie zunächst $abaa^*b$, dann $aa(aa)^*b$. Sie können die beiden Teillösungen gut zusammenfügen.

- Bezeichnen Sie die Zustände gut leserlich
- Markieren Sie den Startzustand des Automaten.
- Markieren Sie deutlich, welche Zustände akzeptieren sind und welche nicht.
- Zeichnen Sie die Zustandsübergänge gut erkenntlich ein: Start- und Endzustand des Übergangs sowie Eingabe, für die dieser Übergang gewählt wird.



Computation: Reguläre Ausdrücke (5 Punkte)

Zur Erinnerung die Bausteine von regulären Ausdrücken, die wir kennen gelernt haben:

a	Das Zeichen a
.	Ein beliebiges Zeichen
[abc]	Ein beliebiges Zeichen aus der Menge {a, b, c}
[a-f]	Ein beliebiges Zeichen aus der Menge {a, b, c, d, e, f}
\d	eine Ziffer [0-9]
X Y	X oder Y
X*	Eine beliebige Wiederholung von X
X+	Mindestens einmal X
X?	Höchstens einmal X
{n}	der vorangehende Ausdruck muss exakt n mal vorkommen
{m,n}	der vorangehende Ausdruck muss mindestens m mal und darf höchstens n mal vorkommen
(xyz)	Gruppierung: xyz müssen miteinander vorkommen
[^xyz]	Nicht x, nicht y, nicht z
\X	X ein Sonderzeichen \()[\]*+?{} ^\$-

Aufgabe 1: Regulären Ausdruck analysieren

Sie finden in einem Programm den folgenden regulären Ausdruck:

((25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.)\{3\}(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)

Der fett markierte Teil vorne und hinten ist identisch: Da brauchen Sie keine Unterschiede zu suchen.

Machen Sie drei möglichst unterschiedliche Beispiele von Zeichenketten, die von diesem Ausdruck akzeptiert werden:

Machen Sie drei möglichst unterschiedliche Beispiele von Zeichenketten, die von diesem Ausdruck *nicht* akzeptiert werden:

Beschreiben Sie präzise in Worten, welche Eingaben der Ausdruck akzeptiert:

--

Aufgabe 2: Regulären Ausdruck erstellen

Ein Noteneingabe-Programm braucht einen regulären Ausdruck, um zu überprüfen, ob die Eingabe eine gültige Schulnote ist.

Geben Sie einen regulären Ausdruck an, der sowohl ganze Noten als auch halbe Noten (zum Beispiel 5.5) und Viertelnoten (zum Beispiel 5.75, 5.25) akzeptiert:

--