

Restgeld

Dickbauer Y., Moser P., Perner M.

PS Computergestützte Modellierung, WS 2016/17

January 6, 2017

Outline

- 1 Aufgabenstellung
- 2 Flow Chart
 - Verwendete Funktionen

Aufgabenstellung

30 Besucher wollen zu einer Veranstaltung. Der Eintritt zur Veranstaltung beträgt 5 Euro. Die Besucher der Veranstaltung können entweder mit 5-Euro oder 10-Euro-Scheinen bezahlen. In der Kasse liegt zu Beginn eine konfigurierbare Anzahl an 5-Euro-Scheinen.

Grundproblem

Die Wahrscheinlichkeit, dass ein Besucher mit einem 10-Euro-Schein bezahlen möchte, beträgt 60 Prozent, dass ein Besucher mit einem 5-Euro-Schein bezahlen möchte, beträgt 40 Prozent.

Alternative

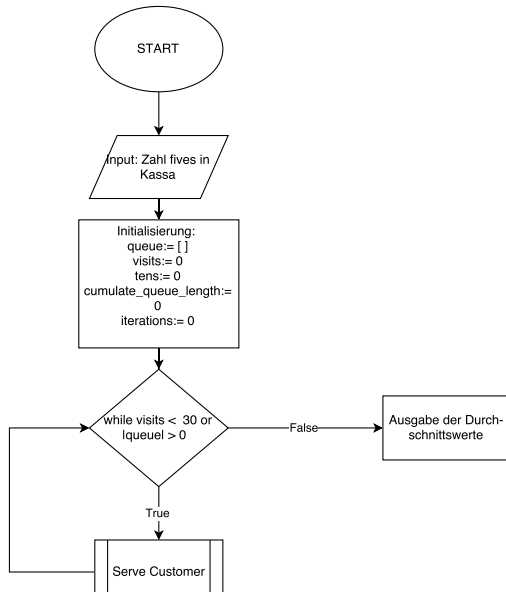
Es gibt 15 Personen mit einem 10 Euro-Schein und 15 Personen mit einem 5 Euro-Schein, die in zufälliger Reihenfolge zur Kasse kommen.

Aufgabenstellung

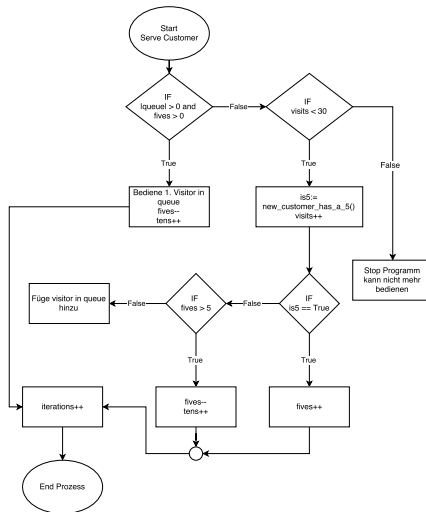
Stellen Sie bei den einzelnen Alternativen mittels Simulation fest, ob jeweils genug 5- Euro-Scheine in der Kasse sind, damit die Besucher mit 10-Euro-Schein das Restgeld bekommen. Stellen Sie fest, ob bzw. wie viele Besucher warten müssen und wie lange sie warten müssen bzw. ob Besucher am Ende sogar ohne Restgeld übrig bleiben. Stellen Sie im Rahmen der Präsentation den Ablauf des Programmes anhand von selbstgewählten Zufallszahlen vor.

- Eingabe: Anzahl an 5-Euro-Scheinen zu Beginn
- Output: Je Runde Geldschein, mit dem der Kunde bezahlen möchte, und Anzahl an 5- und 10-Euro-Scheinen in der Kasse.

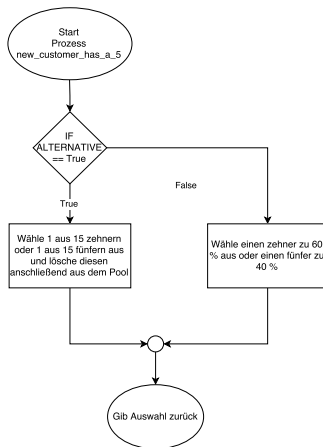
Flow Chart Hauptprogramm Grundproblem



Flow Chart Unterprozess Serve Customer



Flow Chart Unterprozess new_customer_has_a_5() inklusive Alternative



Funktion `user_input(input_vars, [use_defaults])`

- Diese Funktion verlangt vom User die geforderten Eingabeparameter und gibt diese als von der Programmiererin gewünschten Datentyp wieder zurück
- Funktion verlangt als ersten Eingabeparameter die Liste *input_vars*
- Falls *use_defaults == True* wird der User nicht nach Eingabe gefragt (Dient zum Testen)
- Diese Liste besteht wiederum aus Listen mit je Länge = 3:
 - 0: Text, welcher dem User ausgegeben wird
 - 1: Datentyp (int/float/str)
 - 2: Default value: Dieser Wert wird zurueckgegeben, falls *use_defaults == True*

```
1 x, y = user_input((  
2     ('Geben_Sie_einen_X_Wert_ein', int, 10),  
3     ('Geben_Sie_einen_Y_Wert_ein', int, 5), False):
```



Funktion `loaded_random_choice(..)`

- Diese Funktion verlangt eine WSKL Liste als Eingabeparameter
- Gibt einen Index zurück, welcher 0 bis $|probability_list| - 1$ sein kann.
- Diese Indizes haben eine gewichtete WSKL, welche jeweils an der Position in der Eingabeliste steht
- Beispiel `probability_list := [0.9, 0.1]` \Rightarrow mit $p=90\%$ wird 0 zurückgegeben, $p=10\%$ für 1

```
1 def loaded_random_choice(probability_list):
2     n = len(probability_list)
3     random_number = random.random()
4     cum_p = 0
5     for i in range(n):
6         cum_p += probability_list[i]
7         if cum_p > random_number:
8             return i
9     return None
```

