

Abgabefrist: 24.11 23:55

Erreichte Punkte: _____

Name: Katrin SzikoraZeitaufwand in h: 9,5h*Beachten Sie die Abgakriterien! (siehe LVA Übersicht)*

1) Code Quality**5 Punkte**Prüfen Sie die Code Quality des Codes im File `dice_generator.py`.

- Dokumentieren Sie mind. 6 Fälle von schlechter Code Quality.
- Dokumentieren Sie, wie diese Fälle verbessert werden können.
- Überarbeiten Sie den vorhandenen Code mit den Verbesserungen.
- Erstellen Sie Testfälle für den überarbeiteten Code.

2) Debugging**8 Punkte**Der Code im File `christmas_tree.py` ist fehlerhaft.

- Debuggen Sie den Code mit einer der gelernten Variante (Schreibtischtest, Debugging mit `print()` oder Debugging über die Entwicklungsumgebung).
- Dokumentieren Sie Ihr Vorgehen beim Debuggen.
- Dokumentieren und fixen Sie die gefundenen Fehler.
- Debuggen und fixen Sie den gesamten Code.
- Erstellen Sie Testfälle für den funktionierenden Code.

Der Code soll folgendes Verhalten haben:

Es soll ein Weihnachtsbaum gezeichnet werden. Über die Konsole kann die Höhe eingegeben werden. Die Höhe gibt die Anzahl der Zeilen an, die der Baum hoch sein soll. Die niedrigste Höhe soll 3 sein.

Beispiele:

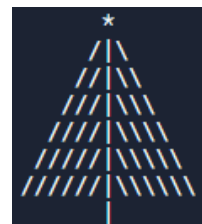
height = 3



height = 5



height = 8

**3) File Handling - Einkaufsliste****12 Punkte**

Erstellen Sie ein Programm, das als Einkaufsliste verwendet werden kann. Die Einkaufsliste soll die Produkte auch über einen Programm-Neustart hinweg speichern. Verwenden Sie dazu File Handling.

Folgende Features sollen implementiert werden:

- Ein Produkt auf die Einkaufsliste hinzufügen.
 - Den Inhalt der Einkaufsliste auf die Konsole ausgeben.
 - Die komplette Einkaufsliste löschen.
 - Ein bestimmtes Produkt von der Einkaufsliste löschen.
-

1. Code Quality

Lösungsidee

Zu Beginn habe ich mir den Code vom File dice_generator.py erst selbst genauer angesehen und dann zusammen mithilfe der Folien von der letzten LV die 6 Fälle von schlechter Code Quality ausfindig gemacht.

Lösung

1. Doppelte Verneinungen sollten vermieden werden.

Verbesserung: nichtweiter mit weiter ersetzen

```
4
5     nichtweiter = False
6     while(not nichtweiter):
7         z = 0
```

2. Der Name der Variablen sollte descriptive sein.

Verbesserung: Folgende Namen den Variablen zuweisen:

z = summe_augenzahl

x = anzahl_wurfel

y = augenzahl

```
6     while(not nichtweiter):
7         z = 0
8         print("Wieviele Würfel?")
9         x = int(input())
10        print("Augenzahl Würfel:")
11        y = int(input())
12
```

3. Es sollen aussprechbare Namen für die Variablen verwendet werden.

Verbesserung: anzahlSchleDurchL mit anzahl_durchlaufe ersetzen

```
15        for anzahlSchleDurchL in range2:
16            gewürfelt = random.randint(1, x)
```

4. Name der Funktion lässt nicht daraus schließen, was die Funktion macht.

Verbesserung: Der Funktion einen passenden Namen wie z.B. play_dice_game

```
3     def funktion1():
4
5         nichtweiter = False
6         while(not nichtweiter):
7             z = 0
8             print("Wieviele Würfel?")
9             x = int(input())
10            print("Augenzahl Würfel:")
11            y = int(input())
```

5. Man sollte sich nicht wiederholen.

Der Ausdruck `range2 = range(x)` ist überflüssig, die Schleife könnte direkt über `range(x)` gelaufen lassen werden.

```
print(„gewürfelt“ + str(gewürfelt))
```

Verbesserung: Schleife direkt über `range(x)` laufen lassen

Zweiten Ausdruck ersetzen mit `summeAugenzahl = summeAugenzahl + gewürfelt`
`print(„Gesamtsumme:“, summeAugenzahl)`

```
13     range2 = range(x)
14
15     for anzahlSchleDurchL in range2:
16         gewürfelt = random.randint(1, y)
```

```
5         for anzahlSchleDurchL in range2:
6             gewürfelt = random.randint(1, y)
7             print("gewürfelt " + str(gewürfelt))
8             z = z + gewürfelt
```

6. Der Code könnte in kleinere Funktionen unterteilt werden.

Eine Funktion sollte nur eine Sache machen. Bietet auch die Möglichkeit das Ergebnis über `return` auszugeben anstatt einer `print`-Funktion.

Verbesserung: Unterteilung in drei Funktionen eine für das Würfeln, die Eingabe des Users und die endgültige Ausgabe

7. Benutzereingabe validieren

Um sicherzugehen das der User eine gültige Eingabe macht.

Verbesserung: Einbau einer `try-except` Anweisung

Implementierung

```
1 import random
2
3 def wurfeln(anzahl_wurfel, augenzahl):
4     return sum(random.randint(1, augenzahl) for anzahl_durchlaufe in range(anzahl_wurfel))
5
6 def user_input():
7     try:
8         anzahl_wurfel = int(input("Wieviele Würfel? "))
9         augenzahl = int(input("Augenzahl Würfel: "))
10        return anzahl_wurfel, augenzahl
11    except ValueError:
12        print("Bitte geben Sie gültige Zahlen ein.")
13        return user_input()
14
15 def play_dice_game():
16     weiter = True
17     while weiter:
18         anzahl_wurfel, augenzahl = user_input()
19         summe_augenzahl = wurfeln(anzahl_wurfel, augenzahl)
20
21         print("Gesamtsumme:", summe_augenzahl)
22
23         print("Weiter? (ja/nein)")
24         weiter = input()
25         if(weiter == "ja"):
26             weiter = True
27         else:
28             weiter = False
29
30 play_dice_game()
31
32
```

Testfälle

Testfall 1

Input:

anzahl_wurfel = 2

augenzahl = 6

weiter = nein

```
macbook@MacBook-Pro-von-Katrin GIN_2023 % /usr/local/bin/python3 /Users/macbook/Documents/GIN_2023/Fell
Wieviele Würfel? 2
Augenzahl Würfel: 6
Gesamtsumme: 9
Weiter? (ja/nein)
nein
macbook@MacBook-Pro-von-Katrin GIN_2023 %
```

Testfall 2

Input:

anzahl_wurfel = 3

augenzahl = 7

weiter = ja

```
macbook@MacBook-Pro-von-Katrin GIN_2023 % /usr/local/bin/python3 /Users/macbook/Documents/GIN_2023/Fellner/Ubun
Wieviele Würfel? 3
Augenzahl Würfel: 7
Gesamtsumme: 14
Weiter? (ja/nein)
ja
Wieviele Würfel? 
```

Testfall 3

Input:

anzahl_wurfel = x

```
macbook@macbook-Pro-von-Katrin: GIN_2023 % /usr/local/bin/python3 /Users/macbook/Documents/GIN_2023/Fellner/Übung4_Katrin_Szikora/dice_generator.py
Wieviele Würfel? x
Bitte geben Sie gültige Zahlen ein.
Wieviele Würfel? █
```

Testfall 4

Input:

anzahl_wurfel = 4

augenzahl = y

```
macbook@macbook-Pro-von-Katrin: GIN_2023 % /usr/local/bin/python3 /Users/macbook/Documents/GIN_2023/Fellner/Übung4_Katrin_Szikora/dice_generator.py
Wieviele Würfel? 4
Augenzahl Würfel: y
Bitte geben Sie gültige Zahlen ein.
Wieviele Würfel? █
```

2) Debugging (Dauer ca. 3,5h)

Lösungsidee/Vorgehen beim Debuggen

Ich habe mich entschieden den Code in der Entwicklungsumgebung zu Debuggen. Zuerst habe ich den Code mit verschiedenen Eingaben laufen lassen, um zu sehen wo Probleme auftreten. Danach habe ich Breakpoints gesetzt an den Stellen bis zu denen ich den Code überprüfen will. Bei diesem Beispiel habe ich die Breakpoints so gesetzt damit vor Beginn der Schleife gestoppt wird. Daraufhin habe ich das Debug-Programm gestartet. Sobald mir ein Bug aufgefallen ist, habe ich diesen gefixt und bin dann weiter im Debug-Programm vorgegangen, bis der letzte behoben war und der Code fehlerfrei lief.

Dokumentation und Fixen Fehler

```
1 print("Wie hoch soll der Baum sein?")
2 height = input()
3
4 #draw top of christmas tree
5 firstLine = ""
6 for i in range((height)+5):
```

Exception has occurred: TypeError X
can only concatenate str (not "int") to str
File "/Users/macbook/Documents/GIN_2023/Fellner/falsechristmas_tree.py", line 6, in <module>
for i in range(height+5):
~~~~~  
TypeError: can only concatenate str (not "int") to str

**Bug:** Der Input für die Variable height wurde nicht als int gekennzeichnet und verhindert somit das Rechnen damit.

**Fix:** input als int kennzeichnen

```
1 print("Wie hoch soll der Baum sein?")
2 height = int(input())
3
4 #draw top of christmas tree
5 firstLine = ""
6 for i in range(height+5):
7     firstLine = firstLine + " "
8     firstLine = firstLine + "*"
9 print(firstLine)
```

**Bug:** Mit dem Ausdruck `height+5` für die `range` steht das Sternchen nicht an der Spitze des Baumes.

**Fix:** Änderung auf `height-2` (Höhe des Körpers des Baumes), um so den richtigen Abstand zur Mitte des Baumes zu erzeugen

```
11 #draw body of christmas tree
12 heightBody = height-2
13 for lineCount in range(1, heightBody+1):
14     line = ""
15     for spaceSymbol in range(heightBody-lineCount):
16         line = line + " "
17     for forwardSlashSymbol in range(lineCount):
18         line = line + "/"
19     line = line + "?"
20     for forwardSlashSymbol in range(heightBody-lineCount):
21         line = line + "\"\"
22     print(line)
23
24 #draw bottom of christmas tree
25 lastLine = ""
```

**Bug:** Für den Stamm ist mit „?“ das falsche Zeichen hinterlegt. Auf der rechten Seite des Baumes werden die die Backslash Symbole nicht symmetrisch zur linken Seite dargestellt, Grund dafür ist der falsche Ausdruck in der `range` (`heightBody-lineCount`). Die Backslashes nehmen mit zunehmender Zeile ab anstatt zu, also genau umgekehrt als eigentlich gedacht. Erkennt man gut bei Eingabe einer größeren Höhe.

**Fix:** „?“ mit „|“ ersetzen, um den Stamm korrekt dazustellen

Die zweite Variable `forwardSlashSymbol` mit `backSlashSymbol` austauschen  
`Range(heightBody-lineCount)` bei `backSlashSymbol` ändern auf `lineCount`, um dasselbe Ergebnis wie rechts zu erhalten

```
21 line = line + "\"\"
22 print(line)
23
24 #draw bottom of christmas tree
25 lastLine = ""
26 for i in range(height-2):
27     lastLine = lastLine + " "
28     lastLine = lastLine + "|"
29     lastLine = lastLine + "\"\"
30     lastLine = lastLine + "\"\"
31 print(lastLine)
```

**Bug:** Für den unteren Teil des Stammes werden drei | anstatt einem ausgegeben.

**Fix:** Die zwei überflüssigen Zeilen Code löschen

## Implementierung

```
Übung4_Katrin_Szikora > - christmas_tree.py > ...
1 print("Wie hoch soll der Baum sein?")
2 height = int(input())
3
4 # checking min height
5 while height < 3:
6     print("Die Mindesthöhe sollte 3 sein. Bitte geben Sie eine gültige Höhe ein:")
7     height = int(input())
8
9 # draw top of christmas tree
10 firstLine = ""
11 for i in range(height-2):
12     firstLine = firstLine + " "
13 firstLine = firstLine + "*"
14 print(firstLine)
15
16 # draw body of christmas tree
17 heightBody = height-2
18 for lineCount in range(1, heightBody+1):
19     line = ""
20     for spaceSymbol in range(heightBody-lineCount):
21         line = line + " "
22     for forwardSlashSymbol in range(lineCount):
23         line = line + "/"
24     line = line + "|"
25     for backSlashSymbol in range(lineCount):
26         line = line + "\"\"
27     print(line)
28
29 # draw bottom of christmas tree
30 lastLine = ""
31 for i in range(height-2):
32     lastLine = lastLine + " "
33 lastLine = lastLine + "|"
34 print(lastLine)
35
```

## Testfälle

### Testfall 1

Input:

height = 2

```
Documents/GIN_2023/Fellner/Übung4_Katrin_Szikora/christmas_tree.py
Wie hoch soll der Baum sein?
2
Die Mindesthöhe sollte 3 sein. Bitte geben Sie eine gültige Höhe ein:
3
*
/|\
|
o macbook@MacBook-Pro-von-Katrin Fellner %
```

## Testfall 2

Input:

height = 3

```
macbook@MacBook-Pro-von-Katrin-Fellner % ./usr/local/bin/python3 /Users/macbook/Documents/Git_2023/fellner/0001
Wie hoch soll der Baum sein?
3
  *
 /|\
 |
macbook@MacBook-Pro-von-Katrin-Fellner %
```

## Testfall 3

Input:

height = 9

[illegible]

## Testfall 4

Input:

height = 1

height = 0

height = 7

[illegible]



## Testfall 5

Input:

height = 14


[illegible]

## Testfall 6

Input:

height = 23

```
Wie hoch soll der Baum sein?  
23
```



### 3. File Handling – Einkaufsliste

#### Lösungsidee

Das Programm für die Einkaufsliste beginnt mit einer Funktion für ein Menü, mit welchem man die verschiedenen Features des Programms anhand der Nummer auswählen kann. Für jedes Feature habe ich eine eigene Funktion erstellt. Die Items, die der User der Einkaufsliste hinzufügt, werden im Array „shopping\_list“ gespeichert. Um die Benutzereingabe an bestimmten Stellen zu validieren habe ich eine else-Funktion eingefügt. Damit der Inhalt im Array in einem Textfile gespeichert wird und die Einkaufsliste auch über den Neustart hinweg zu speichern sind die Funktionen save\_shopping\_list (Schreibmodus) und load\_shopping\_list (Lesemodus) zuständig.

#### Implementierung

```
1  shopping_list = []
2
3  def menu():
4      print("Meine Einkaufsliste")
5      print("1. Produkt zur Einkaufsliste hinzufügen")
6      print("2. Inhalt Einkaufsliste ausgeben")
7      print("3. Komplette Einkaufsliste löschen")
8      print("4. Produkt von Einkaufsliste löschen")
9      print("5. Programm beenden")
10
11
12  def add_products(product, shopping_list):
13      shopping_list.append(product)
14      save_shopping_list(shopping_list)
15      print(f"{product} wurde erfolgreich hinzugefügt.")
16
17
18  def show_shopping_list(shopping_list):
19      print("Meine Einkaufsliste")
20      if len(shopping_list) == 0:
21          print("Deine Einkaufsliste ist leer. Füge zuerst etwas hinzu.")
22          return
23      else:
24          for index, product in enumerate(shopping_list,1):
25              print(f"{index}. {product}")
26
27  def delete_shopping_list(shopping_list):
28      print("Bist du sicher, dass du deine komplette Einkaufsliste löschen willst?")
29      decision_delete = input()
30      if decision_delete == "ja":
31          shopping_list.clear()
32          save_shopping_list(shopping_list)
33          print("Die Einkaufsliste wurde gelöscht.")
34      else:
35          print("Löschvorgang abgebrochen.")
36
37  def remove_product(delete_product, shopping_list):
38      if delete_product in shopping_list:
39          shopping_list.remove(delete_product)
40          save_shopping_list(shopping_list)
41          print(f"{delete_product} wurde erfolgreich entfernt.")
42      else:
43          print(f"{delete_product} ist nicht auf deiner Einkaufsliste")
```

```

45
46 def save_shopping_list(shopping_list):          # Inhalt des Arrays shopping_list wird im File shopping_list.txt gespeichert
47     file = open("shopping_list.txt", "w")
48     for item in shopping_list:
49         file.write(item + "\n")
50     file.close()
51
52 def load_shopping_list():          # Einkaufsliste über Neustart hinweg speichern
53     shopping_list = []
54     file = open("shopping_list.txt", "r")
55     shopping_list = file.readlines()
56     file.close()
57     return shopping_list
58
59 def main():
60     shopping_list = load_shopping_list()
61
62     while True:
63         menu()
64         choice = input("Gib deine Auswahl ein (1-5). ")
65
66         if choice == "1":
67             product = input("Was möchtest du deiner Einkaufsliste hinzufügen? ")
68             add_products(product, shopping_list)
69         elif choice == "2":
70             show_shopping_list(shopping_list)
71         elif choice == "3":
72             delete_shopping_list(shopping_list)
73         elif choice == "4":
74             product = input("Welches Produkt möchtest du von deiner Einkaufsliste entfernen? ")
75             remove_product(product, shopping_list)
76         elif choice == "5":
77             print("Programm beendet")
78             save_shopping_list(shopping_list)
79             break
80         else:
81             print("Ungültige Eingabe. Bitte wähle eine Option von 1 bis 5.")
82
83     main()

```

## Testfälle

### Testfall 1

Input:

choice = 1

product = Mehl, Zucker, Apfelsaft

```

2. Inhalt Einkaufsliste ausgeben
3. Komplette Einkaufsliste löschen
4. Produkt von Einkaufsliste löschen
5. Programm beenden
Gib deine Auswahl ein (1-5). 1
Was möchtest du deiner Einkaufsliste hinzufügen? Mehl
Mehl wurde erfolgreich hinzugefügt.
Meine Einkaufsliste
1. Produkt zur Einkaufsliste hinzufügen
2. Inhalt Einkaufsliste ausgeben
3. Komplette Einkaufsliste löschen
4. Produkt von Einkaufsliste löschen
5. Programm beenden
Gib deine Auswahl ein (1-5). 1
Was möchtest du deiner Einkaufsliste hinzufügen? Zucker
Zucker wurde erfolgreich hinzugefügt.
Meine Einkaufsliste
1. Produkt zur Einkaufsliste hinzufügen
2. Inhalt Einkaufsliste ausgeben
3. Komplette Einkaufsliste löschen
4. Produkt von Einkaufsliste löschen
5. Programm beenden
Gib deine Auswahl ein (1-5). 1
Was möchtest du deiner Einkaufsliste hinzufügen? Apfelsaft
Apfelsaft wurde erfolgreich hinzugefügt.

```

## Testfall 2

Input:

choice = 2

```
Meine Einkaufsliste
1. Produkt zur Einkaufsliste hinzufügen
2. Inhalt Einkaufsliste ausgeben
3. Komplette Einkaufsliste löschen
4. Produkt von Einkaufsliste löschen
5. Programm beenden
Gib deine Auswahl ein (1-5). 2
Meine Einkaufsliste
1. Mehl
2. Zucker
3. Apfelsaft
Meine Einkaufsliste
1. Produkt zur Einkaufsliste hinzufügen
```

## Testfall 3

Input:

choice = 3

decision\_delete = nein

```
Meine Einkaufsliste
1. Produkt zur Einkaufsliste hinzufügen
2. Inhalt Einkaufsliste ausgeben
3. Komplette Einkaufsliste löschen
4. Produkt von Einkaufsliste löschen
5. Programm beenden
Gib deine Auswahl ein (1-5). 3
Bist du sicher, dass du deine komplette Einkaufsliste löschen willst?
nein
Löschvorgang abgebrochen.
Meine Einkaufsliste
1. Produkt zur Einkaufsliste hinzufügen
```

## Testfall 3

Input:

choice = 3

decision\_delete = ja

```
Meine Einkaufsliste
1. Produkt zur Einkaufsliste hinzufügen
2. Inhalt Einkaufsliste ausgeben
3. Komplette Einkaufsliste löschen
4. Produkt von Einkaufsliste löschen
5. Programm beenden
Gib deine Auswahl ein (1-5). 3
Bist du sicher, dass du deine komplette Einkaufsliste löschen willst?
ja
Die Einkaufsliste wurde gelöscht.
Meine Einkaufsliste
1. Produkt zur Einkaufsliste hinzufügen
```

## Testfall 4

Input:

choice = 2

```
Die Einkaufsliste wurde gelöscht.
Meine Einkaufsliste
1. Produkt zur Einkaufsliste hinzufügen
2. Inhalt Einkaufsliste ausgeben
3. Komplette Einkaufsliste löschen
4. Produkt von Einkaufsliste löschen
5. Programm beenden
Gib deine Auswahl ein (1-5). 2
Meine Einkaufsliste
Deine Einkaufsliste ist leer. Füge zuerst etwas hinzu.
Meine Einkaufsliste
```

## Testfall 5

Input:

choice = 4

product = Käse

```
Meine Einkaufsliste
1. Wasser
2. Brot
3. Käse
4. Salat
Meine Einkaufsliste
1. Produkt zur Einkaufsliste hinzufügen
2. Inhalt Einkaufsliste ausgeben
3. Komplette Einkaufsliste löschen
4. Produkt von Einkaufsliste löschen
5. Programm beenden
Gib deine Auswahl ein (1-5). 4
Welches Produkt möchtest du von deiner Einkaufsliste entfernen? Käse
Käse wurde erfolgreich entfernt.
Meine Einkaufsliste
1. Produkt zur Einkaufsliste hinzufügen
```

## Testfall 6

Input:

choice = 2

```
Käse wurde erfolgreich entfernt.
Meine Einkaufsliste
1. Produkt zur Einkaufsliste hinzufügen
2. Inhalt Einkaufsliste ausgeben
3. Komplette Einkaufsliste löschen
4. Produkt von Einkaufsliste löschen
5. Programm beenden
Gib deine Auswahl ein (1-5). 2
Meine Einkaufsliste
1. Wasser
2. Brot
3. Salat
Meine Einkaufsliste
1. Produkt zur Einkaufsliste hinzufügen
```

## Testfall 7

Input:

choice = 5

```
3. Salat
Meine Einkaufsliste
1. Produkt zur Einkaufsliste hinzufügen
2. Inhalt Einkaufsliste ausgeben
3. Komplette Einkaufsliste löschen
4. Produkt von Einkaufsliste löschen
5. Programm beenden
Gib deine Auswahl ein (1-5). 5
Programm beendet
macbook@MacBook-Pro-von-Katrin Übung4 Katrin Szikora %
```

## Testfall 8

Input:

choice = 2

```
Gib deine Auswahl ein (1-5). 5
Programm beendet
macbook@MacBook-Pro-von-Katrin Übung4_Katrin_Szikora % /usr/local/bin/python3 /Users/macbook/Documents/Uni/GIN
iste_final.py
Meine Einkaufsliste
1. Produkt zur Einkaufsliste hinzufügen
2. Inhalt Einkaufsliste ausgeben
3. Komplette Einkaufsliste löschen
4. Produkt von Einkaufsliste löschen
5. Programm beenden
Gib deine Auswahl ein (1-5). 2
Meine Einkaufsliste
1. Wasser

2. Brot

3. Salat

Meine Einkaufsliste
```

## Testfall 9

Input:

choice = 4

product = Zucker

```
Meine Einkaufsliste
1. Produkt zur Einkaufsliste hinzufügen
2. Inhalt Einkaufsliste ausgeben
3. Komplette Einkaufsliste löschen
4. Produkt von Einkaufsliste löschen
5. Programm beenden
Gib deine Auswahl ein (1-5). 4
Welches Produkt möchtest du von deiner Einkaufsliste entfernen? Zucker
Zucker ist nicht auf deiner Einkaufsliste
Meine Einkaufsliste
1. Produkt zur Einkaufsliste hinzufügen
```

## Testfall 10

Input:

choice = 6

```
Gib deine Auswahl ein (1-5). 6
Ungültige Eingabe. Bitte wähle eine Option von 1 bis 5.
Meine Einkaufsliste
1. Produkt zur Einkaufsliste hinzufügen
2. Inhalt Einkaufsliste ausgeben
```