

9,5h

Katrin Szikora

**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



1. **Der Name der Variablen sollte descriptive sein.**

**Verbesserung:** Folgende Namen den Variablen zuweisen:

z = summe\_augenzahl

x = anzahl\_wurfel

y = augenzahl

Ein Bild, das Screenshot, Text, Multimedia-Software enthält.

Automatisch generierte Beschreibung

1. **Es sollen aussprechbare Namen für die Variablen verwendet werden.**

**Verbesserung:** anzahlSchleDurchL mit anzahl\_durchlaeufe ersetzen



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

**Verbesserung:** Der Funktion einen passenden Namen wie z.B. play\_dice\_game

Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

1. **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 + gewurfelt print(„Gesamtsumme:“, summeAugenzahl)





1. **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

1. **Benutzereingabe validieren**

Um sicherzugehen das der User eine gültige Eingabe macht.

**Verbesserung:** Einbau einer try-except Anweisung

**Implementierung**

Ein Bild, das Text, Screenshot enthält.

Automatisch generierte Beschreibung

**Testfälle**

Testfall 1

Input:

anzahl\_wurfel = 2

augenzahl = 6

weiter = nein

Ein Bild, das Text, Schrift, Screenshot enthält.

Automatisch generierte Beschreibung

Testfall 2

Input:

anzahl\_wurfel = 3

augenzahl = 7

weiter = ja

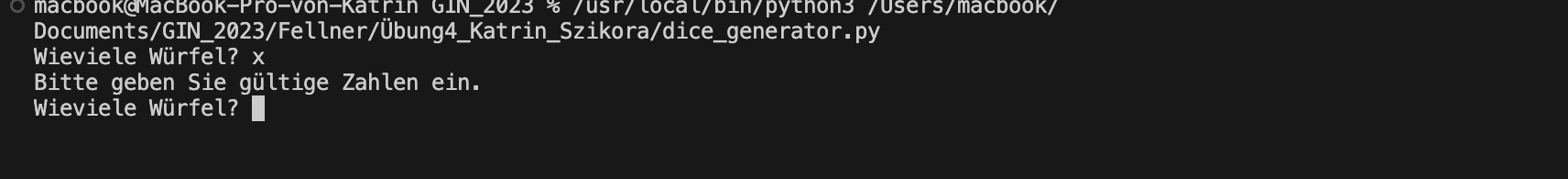
Ein Bild, das Text, Screenshot, Schrift, Schwarz enthält.

Automatisch generierte Beschreibung

Testfall 3

Input:

anzahl\_wurfel = x

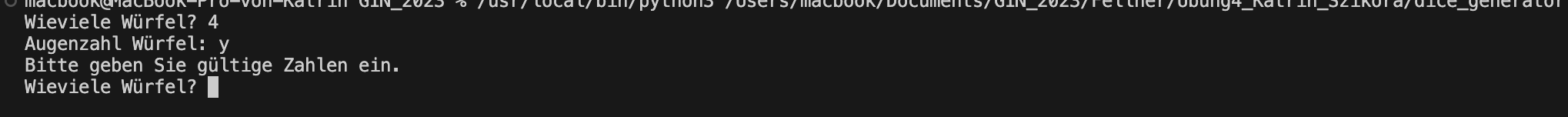


Testfall 4

Input:

anzahl\_wurfel = 4

augenzahl = y



**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**

Ein Bild, das Text, Screenshot, Software, Schrift enthält.

Automatisch generierte Beschreibung

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

**Fix:** input als int kennzeichnen

Ein Bild, das Text, Screenshot, Software, Multimedia-Software enthält.

Automatisch generierte Beschreibung

**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

Ein Bild, das Text, Software, Multimedia-Software, Grafiksoftware enthält.

Automatisch generierte Beschreibung

**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 Backslashs 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

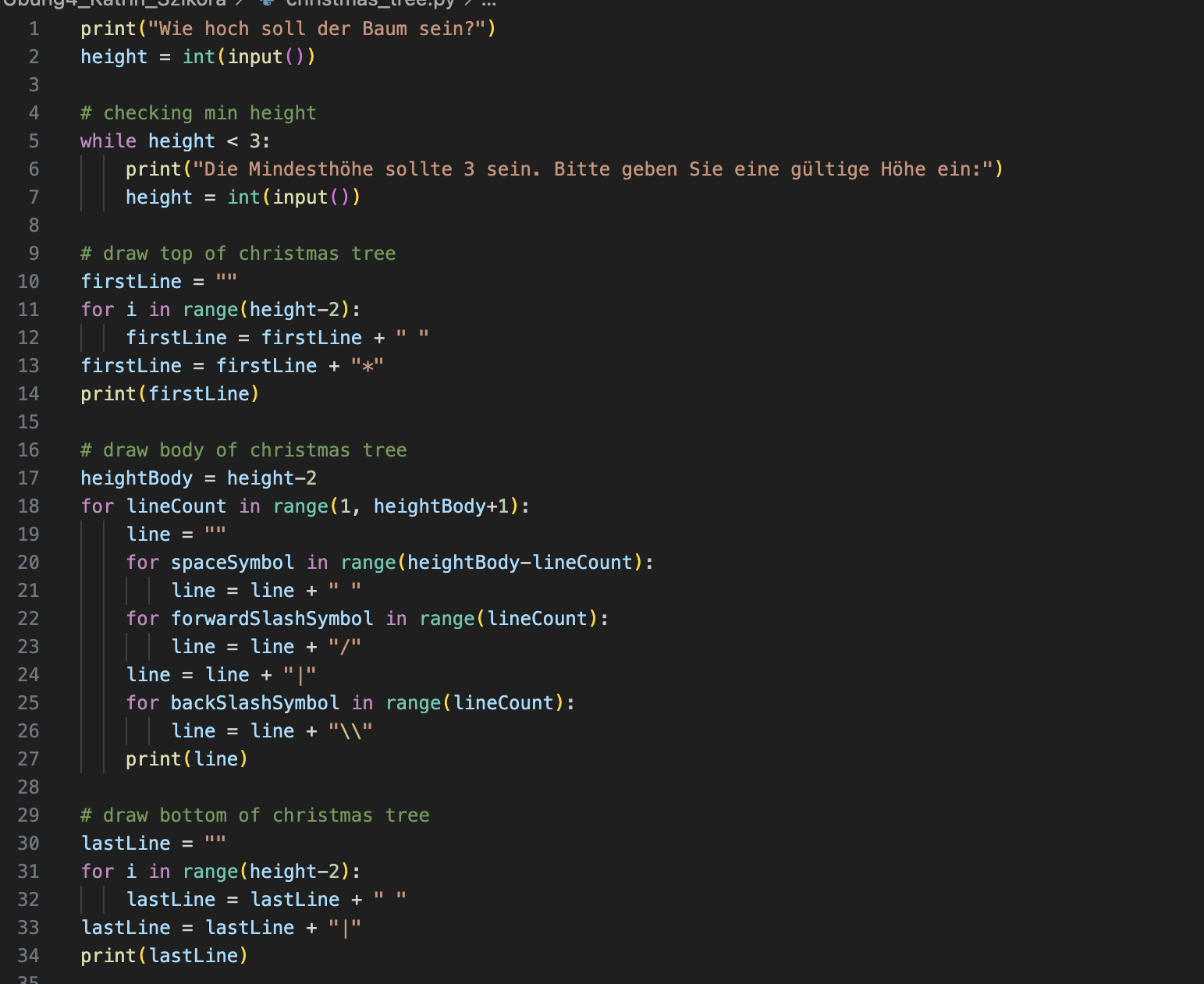
Ein Bild, das Text, Multimedia-Software, Software, Grafiksoftware enthält.

Automatisch generierte Beschreibung

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

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

**Implementierung**



**Testfälle**

Testfall 1

Input:

height = 2

Ein Bild, das Text, Screenshot, Schrift, Schwarz enthält.

Automatisch generierte Beschreibung

Testfall 2

Input:

height = 3

Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

Testfall 3

Input:

height = 9

Ein Bild, das Text, Schrift, Screenshot enthält.

Automatisch generierte Beschreibung

Testfall 4

Input:

height = 1

height = 0

height = 7

Ein Bild, das Text, Schrift, Screenshot enthält.

Automatisch generierte Beschreibung

Testfall 5

Input:

height = 14

Ein Bild, das Text, Screenshot enthält.

Automatisch generierte Beschreibung

Testfall 6

Input:

height = 23

Ein Bild, das Weihnachtsbaum, Screenshot enthält.

Automatisch generierte Beschreibung

**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**

Ein Bild, das Text, Screenshot, Karte Menü enthält.

Automatisch generierte Beschreibung

**Ein Bild, das Text, Screenshot, Software enthält.

Automatisch generierte Beschreibung**

**Testfälle**

Testfall 1

Input:

choice = 1

product = Mehl, Zucker, Apfelsaft

**Ein Bild, das Text, Screenshot, Karte Menü, Schrift enthält.

Automatisch generierte Beschreibung**

Testfall 2

Input:

choice = 2

Ein Bild, das Text, Screenshot, Schwarz, Schrift enthält.

Automatisch generierte Beschreibung

Testfall 3

Input:

choice = 3

decision\_delete = nein

Ein Bild, das Text, Screenshot, Schrift, Schwarz enthält.

Automatisch generierte Beschreibung

Testfall 3

Input:

choice = 3

decision\_delete = ja

Ein Bild, das Text, Screenshot, Schrift, Schwarz enthält.

Automatisch generierte Beschreibung

Testfall 4

Input:

choice = 2

Ein Bild, das Text, Screenshot, Schwarz, Schrift enthält.

Automatisch generierte Beschreibung

Testfall 5

Input:

choice = 4

product = Käse

Ein Bild, das Text, Screenshot, Schwarz, Schrift enthält.

Automatisch generierte Beschreibung

Testfall 6

Input:

choice = 2

Ein Bild, das Text, Screenshot, Schwarz, Schwarzweiß enthält.

Automatisch generierte Beschreibung

Testfall 7

Input:

choice = 5

Ein Bild, das Text, Screenshot, Schrift, Schwarz enthält.

Automatisch generierte Beschreibung

Testfall 8

Input:

choice = 2

Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

Testfall 9

Input:

choice = 4

product = Zucker

Ein Bild, das Text, Screenshot, Schrift, Schwarz enthält.

Automatisch generierte Beschreibung

Testfall 10

Input:

choice = 6

Ein Bild, das Text, Screenshot, Schrift, Schwarz enthält.

Automatisch generierte Beschreibung