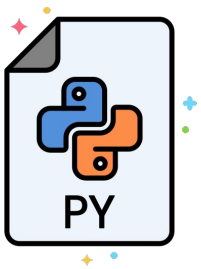
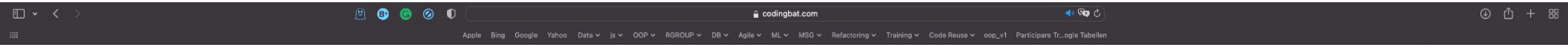
3. Funktionen in Python





CodingBat code practice

id/email
password
log in
forgot password | create account

Welcome to Codingbat. See help for the latest.

Java | Python

Warmup-1 ☆☆☆☆
Simple warmup problems to get started, no loops (solutions available)

Warmup-2 ☆☆☆
Medium warmup string/list problems with loops (solutions available)

String-1 ☆☆☆☆
Basic python string problems -- no loops

List-1 ☆☆☆☆
Basic python list problems -- no loops.

Logic-1 ☆☆☆
Basic boolean logic puzzles -- if else and or not

Logic-2 ☆☆☆
Medium boolean logic puzzles -- if else and or not

String-2 ☆☆
Medium python string problems -- 1 loop.

List-2 ☆☆
Medium python list problems -- 1 loop.

Python Help

- Python Example Code
- Python Strings
- Python Lists
- Python If Boolean
- Code Badges

JavaScript

Very Easy

Tags (optional)

HIDE COMPLETE

Level 0

Watch a quick demo on how Edabit works. WATCH DEMO

How Edabit Works
This is an introduction to how challenges on Edabit work. In the Code tab above you'll see a starter function that looks like this: function hello() { } All you have to do is type return "hello edabit.com" between the curly braces { } and then click the Check button. If you did this correctly, the button will turn red and ...
language_fundamentals strings Very Easy

Return the Sum of Two Numbers
Create a function that takes two numbers as arguments and returns their sum. Examples addition(3, 2) → 5 addition(-3, -6) → -9 addition(7, 3) → 10 Notes Don't forget to return the result. If you get stuck on a challenge, find help in the Resources tab. If you're really stuck, unlock solutions in the Solutions tab.
algebra language_fundamentals math numbers Very Easy

Convert Minutes into Seconds
Write a function that takes an integer minutes and converts it to seconds. Examples convert(5) → 300 convert(3) → 180 convert(2) → 120 Notes Don't forget to return the result. If you get stuck on a challenge, find help in the Resources tab. If you're really stuck, unlock solutions in the Solutions tab.
language_fundamentals math numbers Very Easy

Return the Next Number from the Integer Passed
Create a function that takes a number as an argument, increments the number by +1 and returns the result. Examples addition(0) → 1 addition(9) → 10 addition(-3) → -2 Notes Don't forget to return the result. If you get stuck on a challenge, find help in the Resources tab. If you're really stuck, unlock solutions in the So ...
algebra math numbers Very Easy

Area of a Triangle
Write a function that takes the base and height of a triangle and return its area. Examples triArea(3, 2) → 3 triArea(7, 4) → 14 triArea(10, 10) → 50 Notes The area of a triangle is: (base * height) / 2 Don't forget to return the result. If you get stuck on a challenge, find help in the Resources tab. If you're really st ...
geometry math numbers Very Easy

Convert Age to Days
Create a function that takes the age in years and returns the age in days. Examples calcAge(65) → 23725 calcAge(0) → 0 calcAge(20) → 7300 Notes Use 365 days as the length of a year for this challenge. Ignore leap years and days between last birthday and now. Expect only positive integer inputs.
algebra algorithms math Very Easy

Buggy Code (Part 1)
Fix the code in the code tab to pass this challenge (only syntax errors). Look at the examples below to get an idea of what the function should do. Examples cubes(3) → 27 cubes(5) → 125 cubes(10) → 1000 Notes READ EVERY WORD CAREFULLY, CHARACTER BY CHARACTER! Don't overthink this challenge; it's not supposed to be hard.
bugs language_fundamentals Very Easy

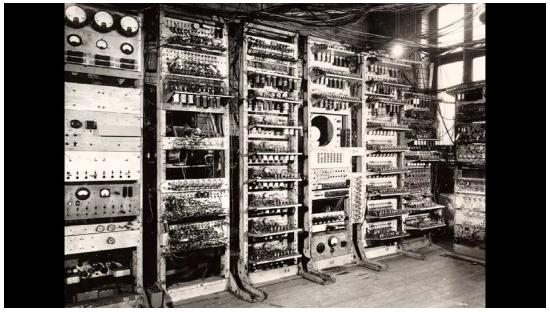
Return the First Element in an Array
Create a function that takes an array containing only numbers and return the first element. Examples getFirstValue([1, 2, 3]) → 1 getFirstValue([80, 5, 100]) → 80 getFirstValue([-500, 0, 50]) → -500 Notes The first element in an array always has an index of 0.
arrays language_fundamentals Very Easy

Power Calculator
Create a function that takes voltage and current and returns the calculated power. Examples circuitPower(230, 10) → 2300 circuitPower(110, 3) → 330 circuitPower(480, 20) → 9600 Notes Requires basic calculation of electrical circuits (see Resources for info).
math numbers Very Easy

Convert Hours into Seconds
Write a function that converts hours into seconds. Examples howManySeconds(2) → 7200 howManySeconds(10) → 36000 howManySeconds(24) → 86400 Notes 60 seconds in a minute, 60 minutes in an hour Don't forget to return your answer.
language_fundamentals math numbers Very Easy

Inhalt

- Strings
- Dateien (ab Lab 4)
  - Ein- und Ausgabe
- Struktur eines Programms



Strings – einige Methoden

`s.strip([z]), s.rstrip([z]), s.lstrip([z])`

- liefert eine Kopie von s, Leerzeichen am Rand, bzw. nur am rechten oder nur am linken Rand entfernt

```
s=" example  "
```

`s.split([z])`

- liefert eine Liste mit allen Elementen aus dem String
  - mit einem Leerzeichen trennt
- Trennzeichen kann mit z spezifiziert werden

```
s = "abc"
s.split() → ["a","b","c"]

s = "a%b%c"
s.split("%") → ["a","b","c"]
```

Strings – einige Methoden

`s.isdigit()`
- sind alle Zeichen in s Ziffern?

`s.isalpha()`
- sind alle Zeichen in s Buchstaben?

`s.isalnum()`
- sind alle Zeichen in s Ziffern oder Buchstaben oder '_'?

`s.isspace()`
- sind alle Zeichen in s Leerzeichen?

Strings – einige Methoden

- Liefert einen String mit allen Elementen
- mit einem spezifizierten Zeichen getrennt

```
words = ["Ana", "are", "mere"]

result = ",".join(words)

print(result)

#Ana,are,mere
```

Dateien in Python

- Bisher haben wir Eingaben stets über die Konsole eingelesen
- ist natürlich nur mit kleinen Eingaben sinnvoll möglich
- Programme können ihre Eingabe auch aus Dateien lesen
  - die Ergebnisse in Dateien abspeichern.

Dateien in Python

- Inhalt: eine Folge von Zeichen (Buchstaben, Ziffern, etc.)
- Textdateien enthalten keine Informationen über die Darstellung der Zeichen bzw. Formatierung
- Der Text kann auf Zeilen aufgeteilt sein: ‘\n’

Dateien in Python

```
f = open(filename, mode)
```

- öffnet die Datei filename zum Lesen (mode 'r') oder Schreiben (mode 'w') und liefert ein „Dateiobjekt“
- mit diesem Objekt kann man die Datei bearbeiten
- normalerweise werde Dateien im „Textmodus“ geöffnet
  - Daten haben eine lesbare Darstellung

```
f.close()
```

- schließt die Datei wieder

```
f = open("words.txt", "w")
f.write("Ana are mere")
f.close()
```

Dateien in Python

Methoden für die Ein- und Ausgabe

- `f.write(something)`
  - schreibt den String something in f
- `f.read()`
  - liest die gesamte Datei
- `f.readline()`
  - liest eine einzelne Zeile
- `f.readlines()`
  - liest alle Zeilen

Dateien in Python

Datei (data.txt) mit drei Zeilen:
Ana
are
mere

Ergebnis: ["Ana", "are", "mere"]

```
f = open("data.txt", "r")
l = []
s = f.readline()
while s != "":
# Dateiende erreicht?
   l.append(s)
   s = f.readline()

f.close()
print (l)
```

```
f = open("Data.txt", "r")
l = []

for s in f:
   l.append(s)

f.close()
print (l)
```

Dateien in Python

- Dateien sollten immer geschlossen werden, sobald man nicht mehr auf sie zugreifen möchte
- Mit der with Anweisung können Dateien automatisch geschlossen werden, sobald der with-block verlassen wird.

```
with open(filename) as f:
   for line in f:
      print(line)

#funktioniert wie
f = open(filename, "r")
for line in f:
   print(line)
f.close()
```

Exkurs: Dateien in Python

Gegeben sei die folgende

```
Datei (Student-ID, Name, Note)
101 Bob 10
104 Lob 5
103 Dob 9
109 Dobby 4
```

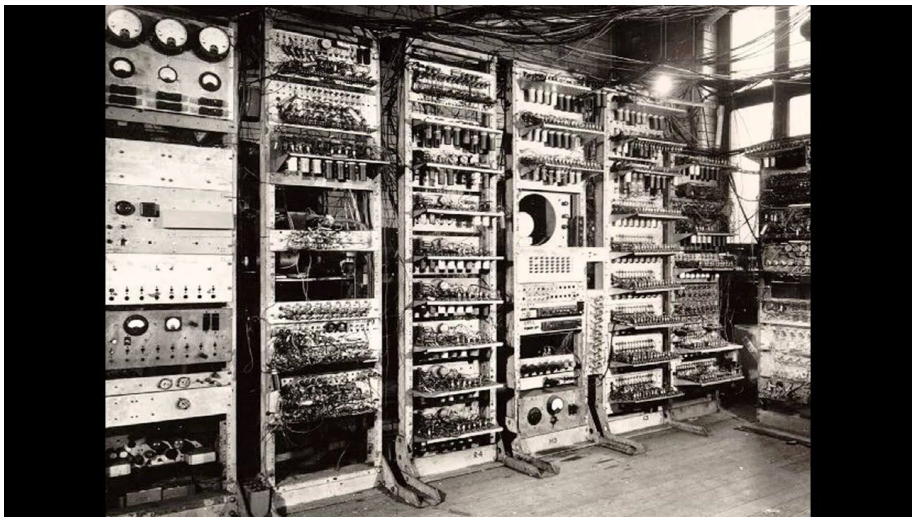
1. Schreibe eine Funktion, die ein Student in der Datei abspeichert.
2. Schreibe eine Funktion, die den “besten” Student liefert.
3. Schreibe eine Funktion, die den Namen eines Studenten ändert.

...es **kann** nützlich sein

Labor 4+

a long long time ago….

- einmal war es einfach
- die Welt war irgendwie größer
- 60s - 70s → Computer waren groß, langsam und Anwendungsfälle waren äußerst begrenzt



- Nicht nur Hardware ist schneller geworden, sondern wir leben in einer vernetzten Welt
- Heute läuft Software auf Handys, Computern, Smartwatches und Haushaltsgeräten
- beeinflusst alle Aspekte unseres Lebens
- und jeder kann Code schreiben

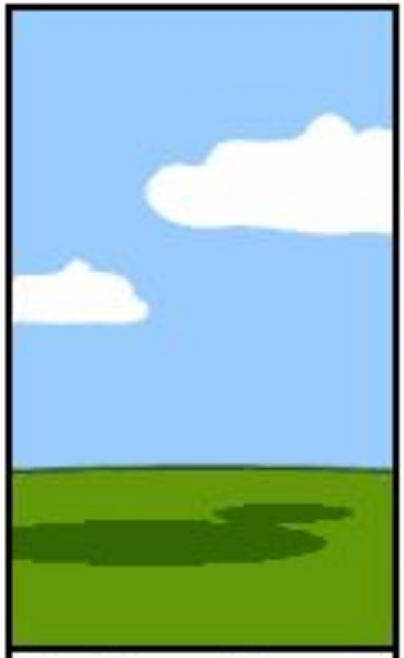
How the customer explained it

How the Project Leader understood it

How the Analyst designed it

How the Programmer wrote it

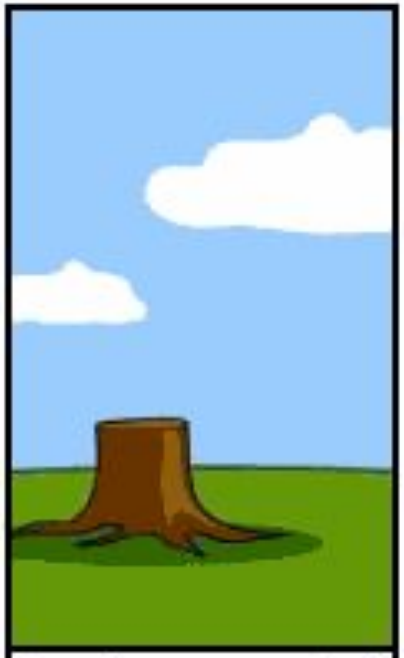
How the Business Consultant described it



How the project was documented

What operations installed

How the customer was billed



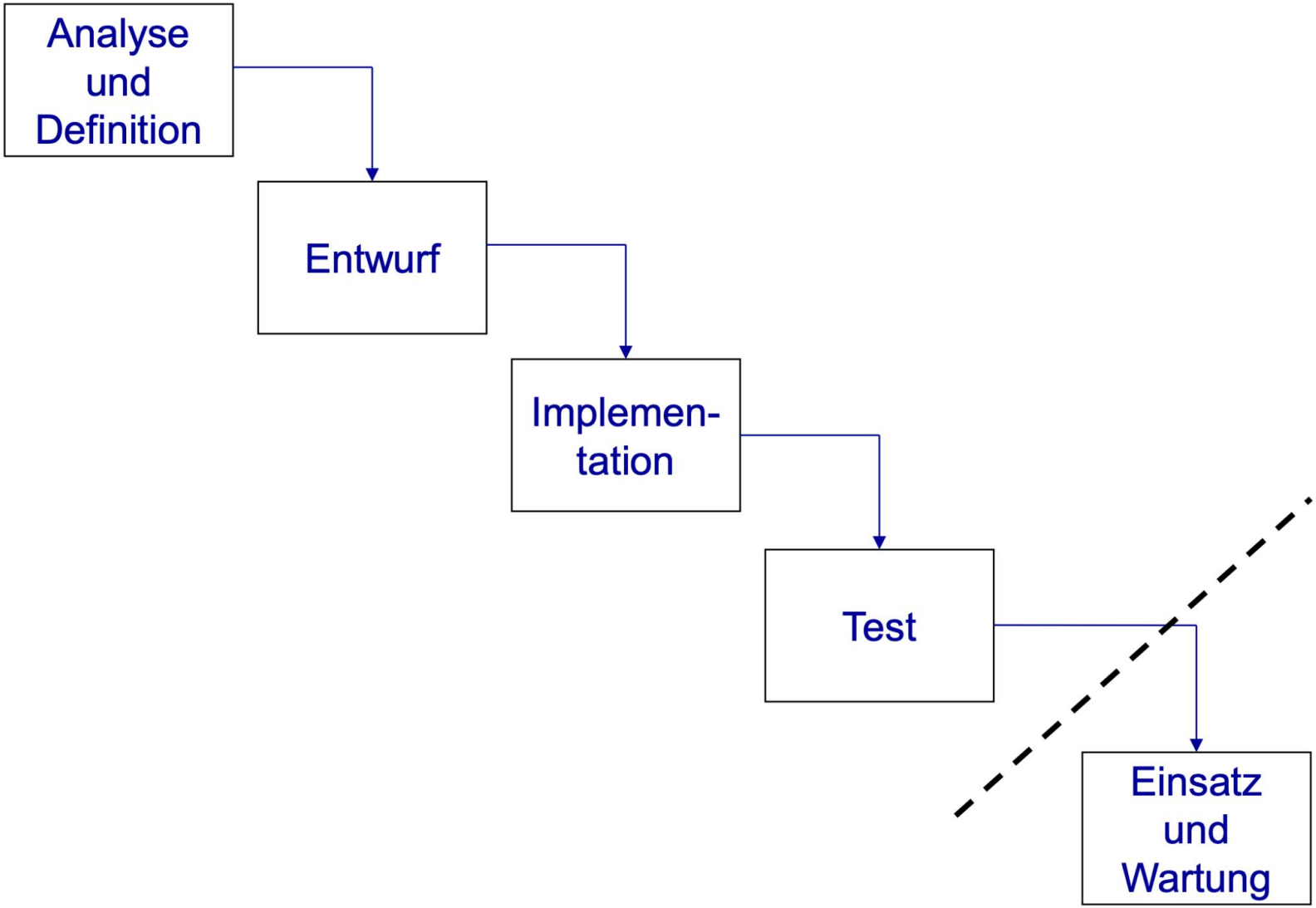
How it was supported

What the customer really needed

Woran liegt es?

- nicht vollständige Anforderungen
- inkompetente Mitarbeiter
- fehlende Unterstützung durch das Management
- zu große Erwartungen
- falsche Schätzung der Zeit/Kosten
- Managementfehler

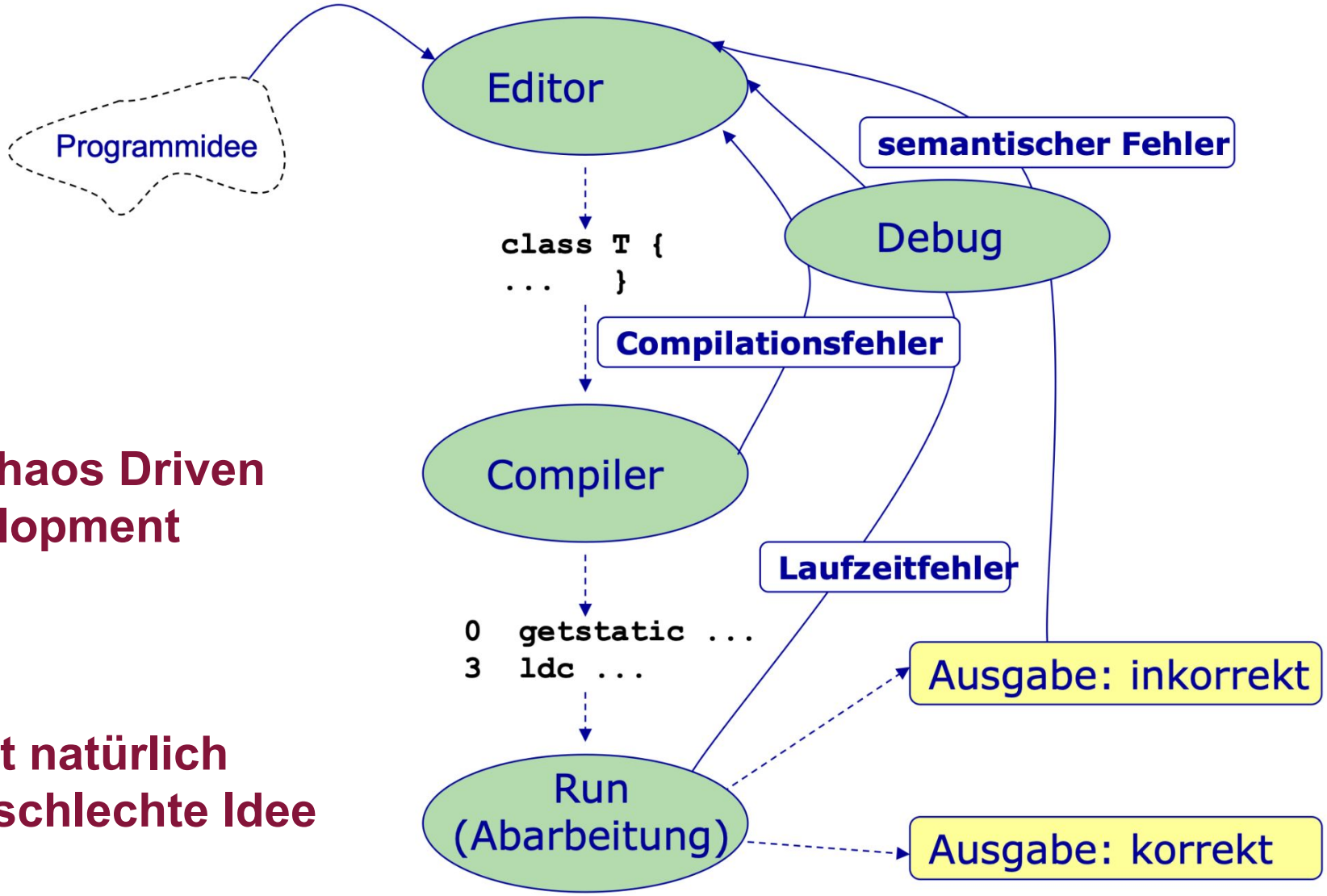
Klassisches Modell für Softwareentwicklung



Phasen der Softwareentwicklung

- Analyse und Definition
  a. Analyse des Problems + Definition der Anforderungen
  b. Kooperation: Auftraggeber ↔ Auftragnehmer
     Produktdefinition
  c. Output: Systemspezifikation, Anforderungsdefinition, Pflichtenheft
- Entwurf (Design)
  a. Struktur, Aufbau, Komponenten der SW
  b. und ihre Beziehungen festlegen
  c. Output: Software-Architektur
- Implementation
  a. Software-Architektur "ausgefüllt"
  b. Programmierung der Komponenten
  c. Output:Programm
- Test
  a. Test der Komponenten, Test ihrer Integration, Systemtest
  b. Output: Testfälle, Ergebnisse des Testlaufs

Softwareentwicklung: erster Versuch



Chaos Driven Development

ist natürlich eine schlechte Idee

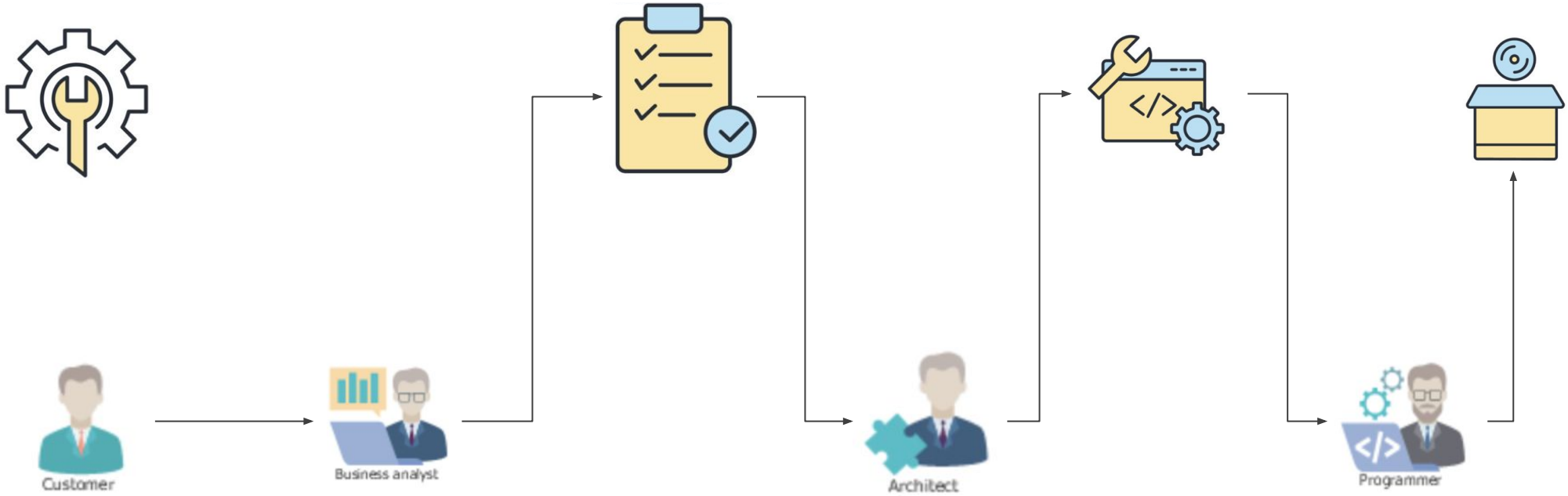
Dokumenten der Softwareentwicklung

Kundenwünsche

Spezifikation

Entwurf

Code



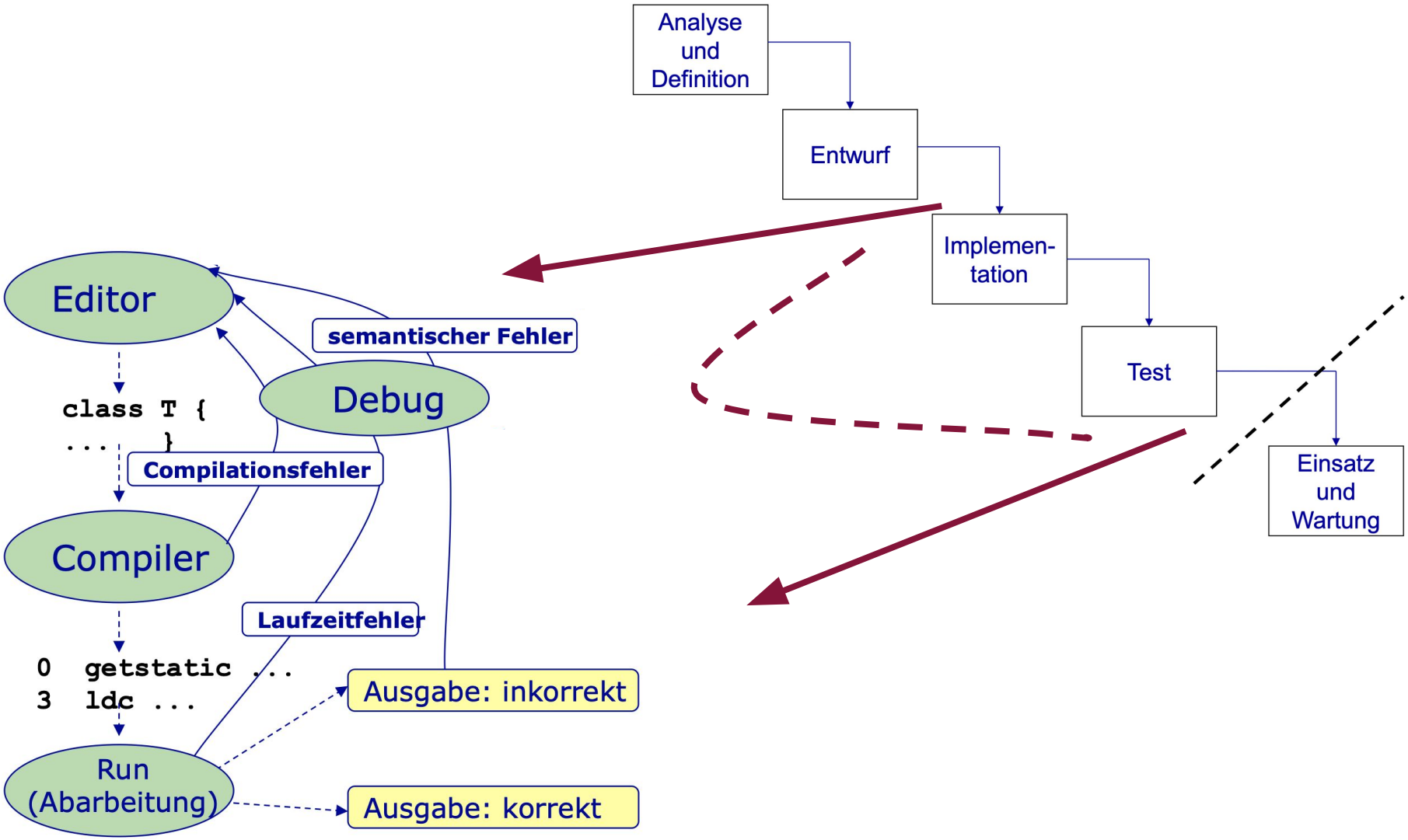
Stakeholder

Berater

Architekt

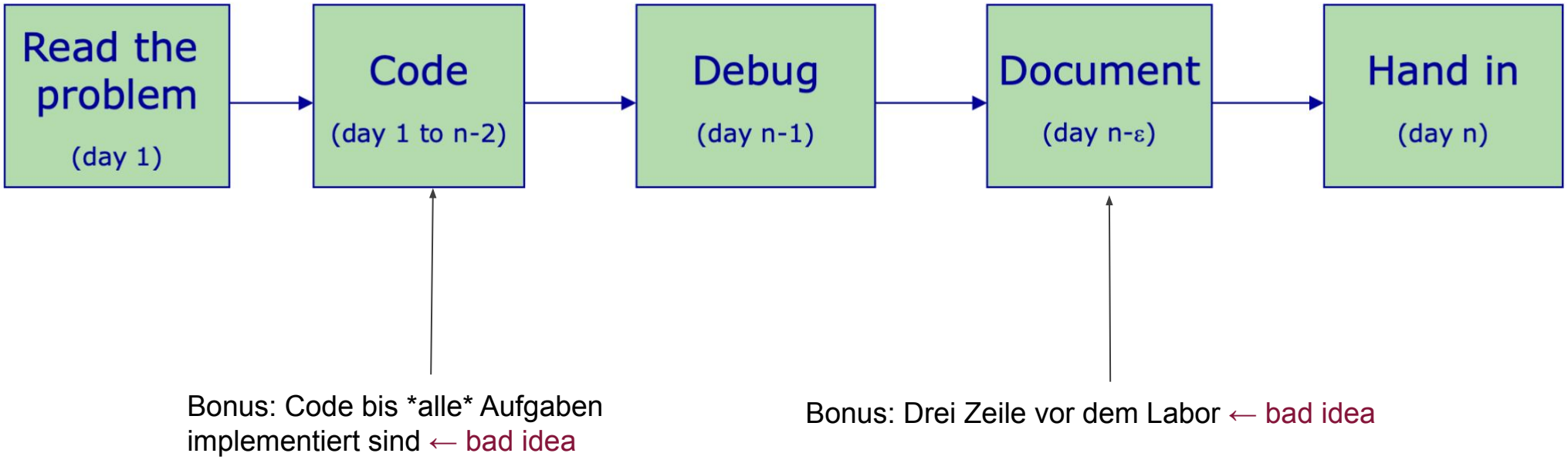
Dev

Softwareentwicklung: zweiter Versuch



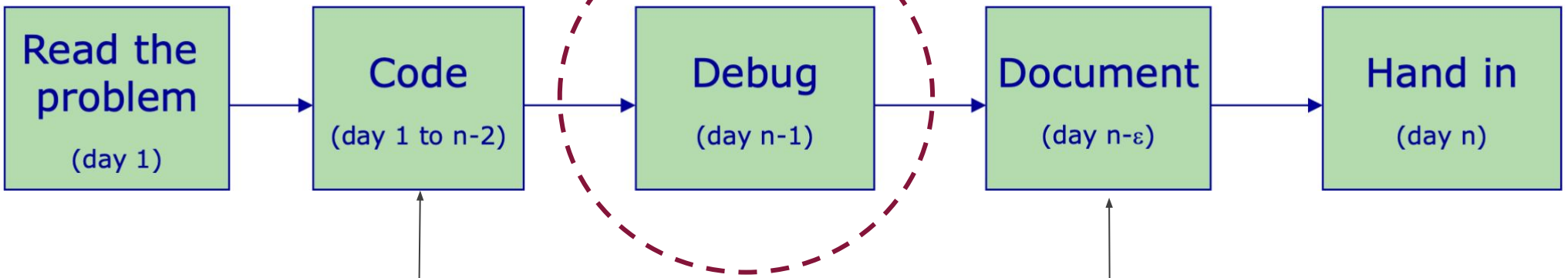
Softwareentwicklung: was wird der Student machen

…in der Praxis



Softwareentwicklung: was wird der Student machen

90% der Zeit wird hier verbracht



Bonus: Code bis *alle* Aufgaben implementiert sind

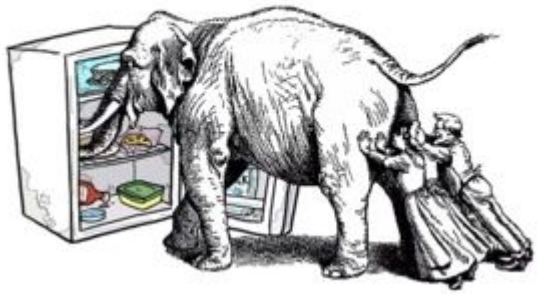
Bonus: Drei Zeile vor dem Labor

Softwareentwicklung

das bedeutet: man braucht einen ‘Plan’ und sollte nicht direkt springen und Code schreiben…

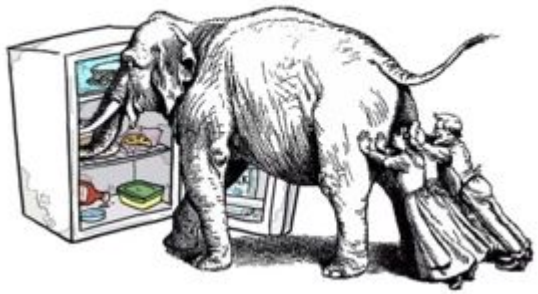

:wq_
$ killall vim -9_

Wie stellt man in einen Kühlschrank einen Elefant?



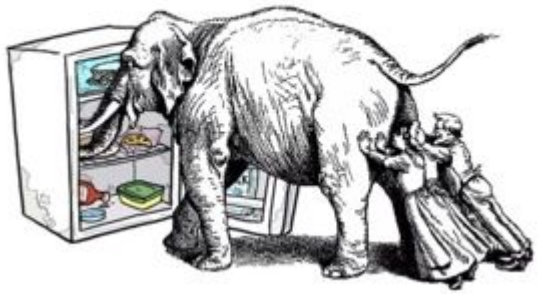
Wie stellt man in einen Kühlschrank einen Elefant?

1. Man öffnet den Kühlschrank



Wie stellt man in einen Kühlschrank einen Elefant?

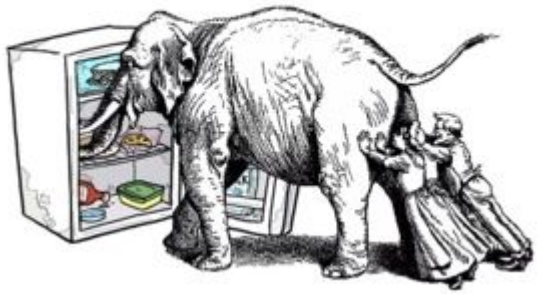
1. Man öffnet den Kühlschrank
2. stellt den Elefant hinein



Wie stellt man in einen Kühlschrank einen Elefant?

1. Man öffnet den Kühlschrank
2. stellt den Elefant hinein
3. schließt die Tür

...aber



wie denkt ein Entwickler?

wie denkt ein Entwickler?

1. teile das Problem in **mehrere Probleme** auf

wie denkt ein Entwickler?

1. teile das Problem in **mehrere Probleme** auf
2. finde **Lösungen** für die kleine Probleme

wie denkt ein Entwickler?

1. teile das Problem in **mehrere Probleme** auf
2. finde **Lösungen** für die kleine Probleme
3. stelle die Lösungen zusammen

wie denkt ein Entwickler?

1. teile das Problem in **mehrere Probleme** auf
2. finde **Lösungen** für die kleine Probleme
3. stelle die Lösungen zusammen
4. Aufräumen (**refactor**)

teile das Problem auf

1. Was für einen Kühlschrank habe ich?
2. Aber der Elefant?
3. Wo findet man Elefanten?
4. Transport
5. Was sollte man machen, falls der Elefant zu groß ist?

finde Lösungen

finde Lösungen

1. Was für einen Kühlschrank nutze ich?

finde Lösungen

1. Was für einen Kühlschrank nutze ich? - Mein

finde Lösungen

1. Was für einen Kühlschrank nutze ich? - Mein
2. Aber der Elefant?

finde Lösungen

1. Was für einen Kühlschrank nutze ich? - Mein
2. Aber der Elefant? - Afrikanischer

finde Lösungen

1. Was für einen Kühlschrank nutze ich? - Mein
2. Aber der Elefant? - Afrikanischer
3. Wo findet man Elefanten?

finde Lösungen

1. Was für einen Kühlschrank nutze ich? - Mein
2. Aber der Elefant? - Afrikanischer
3. Wo findet man Elefanten? - Afrika

finde Lösungen

1. Was für einen Kühlschrank nutze ich? - Mein
2. Aber der Elefant? - Afrikanischer
3. Wo findet man Elefanten? - Afrika
4. Transport

finde Lösungen

1. Was für einen Kühlschrank nutze ich? - Mein
2. Aber der Elefant? - Afrikanischer
3. Wo findet man Elefanten? - Afrika
4. Transport - mit dem Flugzeug, im Gepäck

finde Lösungen

1. Was für einen Kühlschrank nutze ich? - Mein
2. Aber der Elefant? - Afrikanischer
3. Wo findet man Elefanten? - Afrika
4. Transport - mit dem Flugzeug, im Gepäck
5. Was sollte man machen, falls der Elefant zu groß ist?

stelle die Lösungen zusammen

1. ich leihe ein shrinkgun von Gru
2. fliege nach Südafrika
3. besichtige einen Elephant-Park
4. finde einen Elefant im Park
5. schieße mit dem shrinkgun den Elefant
6. lege den Elefant ins Gepäck
7. fahre zum Flughafen
8. fliege zurück
9. fahre nach Hause
10. stecke den Elefant in den Kühlschrank

und für Programmierung….

Wir brauchen eine Funktion, das die Anzahl von Erscheinungen aller Elemente in einer Liste bestimmt.
Beispiel:

- input: `l = [1, 2, 6, 5, 3, 4, 2, 4, 1]`
- output: `1 - 2, 2 - 2, 3 - 1, 4 - 2, 5 - 1`

Fragen:

- wie kann man das output repräsentieren?

Schritte:

1. man muss das Output initialisieren
2. man muss alle Elemente der Liste durchgehen
3. für ein Element bestimmt man die Anzahl von Erscheinungen
4. man fügt die anzahl in Output
5. man gibt das Output zurück

und für Programmierung….

Fragen:

- wie kann man das output repräsentieren?
  - Dictionary

Schritte:

- man muss das Output initialisieren
  - `d = {}`
- man muss alle Elemente der Liste durchgehen
  - `for elem in l:`
- für ein Element man bestimmt die Anzahl von Erscheinungen
  - **neue Funktion: `anzahl()`**
- man fügt die anzahl in Output
  - `d["elem"] = a`
- man gibt das Output zurück
  - `return d`

und für Programmierung….

```
      l = [1,2,6,5,3,4,2,4,1]

      def my_funk(l):
S1.         d = {}

S2.         for elem in l:
S3.            a = anzahl(elem,l):
S4.            d["elem"] = a

S5.         return d
```

und für Programmierung….

jetzt muss man das Gleiche für die Anzahl Funktion machen

```
def anzahl(el, l):
    a = 0

    for elem in l:
        if el == elem:
            a += 1

    return a
```

Labor 3+

- Man muss Konsoleanwendungen implementieren
- Code muss in Funktionen unterteilt werden
- Jede Funktion muss nur genau eine Sache tun
- Funktionen führen entweder Input/Output Operationen oder Berechnungen durch, aber nicht beides!
- Non-UI-Funktionen müssen spezifiziert werden
- Man muss alle Non-UI-Funktionen testen

Jede Funktion muss nur genau eine Sache tun

```
def magik_funktion():
    a = int(input("a="))
    b = int(input("b="))

    sum = a + b

    print("Sum of a and b is", sum)
```

Jede Funktion muss nur genau eine Sache tun

```
def magik_funktion():
    a = int(input("a="))
    b = int(input("b="))

    sum = a + b

    print("Sum of a and b is", sum)
```

Jede Funktion muss nur genau eine Sache tun

```
def add(a,b):
    sum = a + b
    return sum

def print_message(msg, val):
    print (msg, val)

a = int(input("a="))
b = int(input("b="))

sum = add(a,b)
print_message("Sum of a and b is", sum)
```

Non-UI-Funktionen müssen spezifiziert werden

```
def absolute_value(num):
    """Diese Funktion gibt den absoluten Wert
         einer eingegebenen Zahl zurück
    """

    if num >= 0:
        return num
    else:
        return -num

def main():
   print(absolute_value(2))
   print(absolute_value(-4))

main()
```

assert

```
assert <condition>
```

evaluiert <condition> und
→ führt zum Fehler, wenn <condition> False
→ geht weiter, wenn <condition> True

```
assert 1 == 1
```
→ OK

```
s = "abc"
assert len(s) == 3
```
→ OK

```
assert len(s) == 2
```

Man muss alle Non-UI-Funktionen testen

- testen = man muss sicherstellen, dass die Funktion richtig funktioniert
- testet die funktion mit unterschiedlichen Anwendungsfälle

```
def add(a,b):
    return a + b

def test_add_1():
    assert add(1,2) == 3

def test_add_2():
    assert add(2,-2) == 0
```

```
def run_test():
    test_add_1()
    test_add_2()

run_tests()
```

Exkurs: Testing und Dokumentation

Schreibe Funktionen für die folgenden Anforderungen:

1. für zwei Vektors (als Listen gespeichert) das kartesische Produkt berechnet
2. den Mittelwert eines Vektors ermittelt