

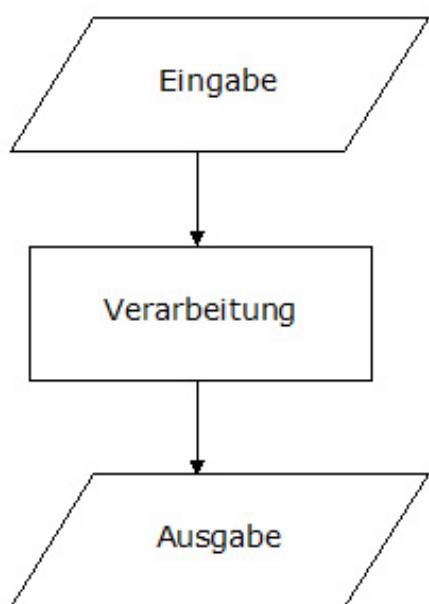
Informatik

inf11 08.10.2020

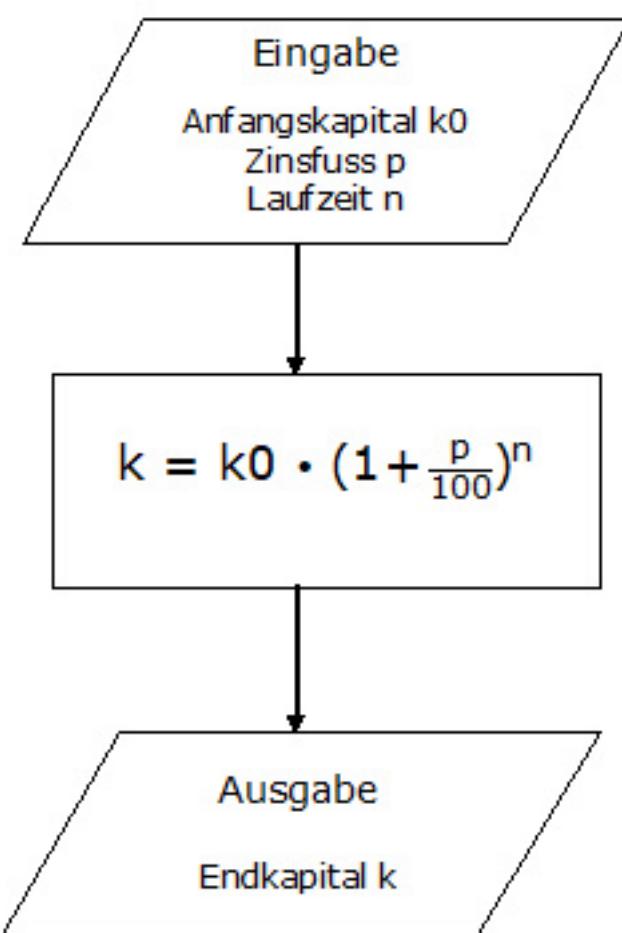
Definition:

Unter einem **Algorithmus** verstehen wir ein aus endlich vielen Anweisungen bestehendes allgemeines Verfahren, welches eine Klasse von Problemen in endlich vielen Schritten löst.

Allgemeines Flußdiagramm
eines Algorithmus:



Flußdiagramm des Algorithmus
„Zinseszins“:



1. Lineare Algorithmen

Wenn bei der Abarbeitung eines Algorithmus die einzelnen Anweisungen sich längs eines einzigen Pfades aneinanderreihen, sprechen wir von einem linearen Algorithmus; insbesondere gibt es hier keine Verzweigungen. Beispiele: Quaderberechnung, Zinseszinsberechnung.

Aufgabe 1

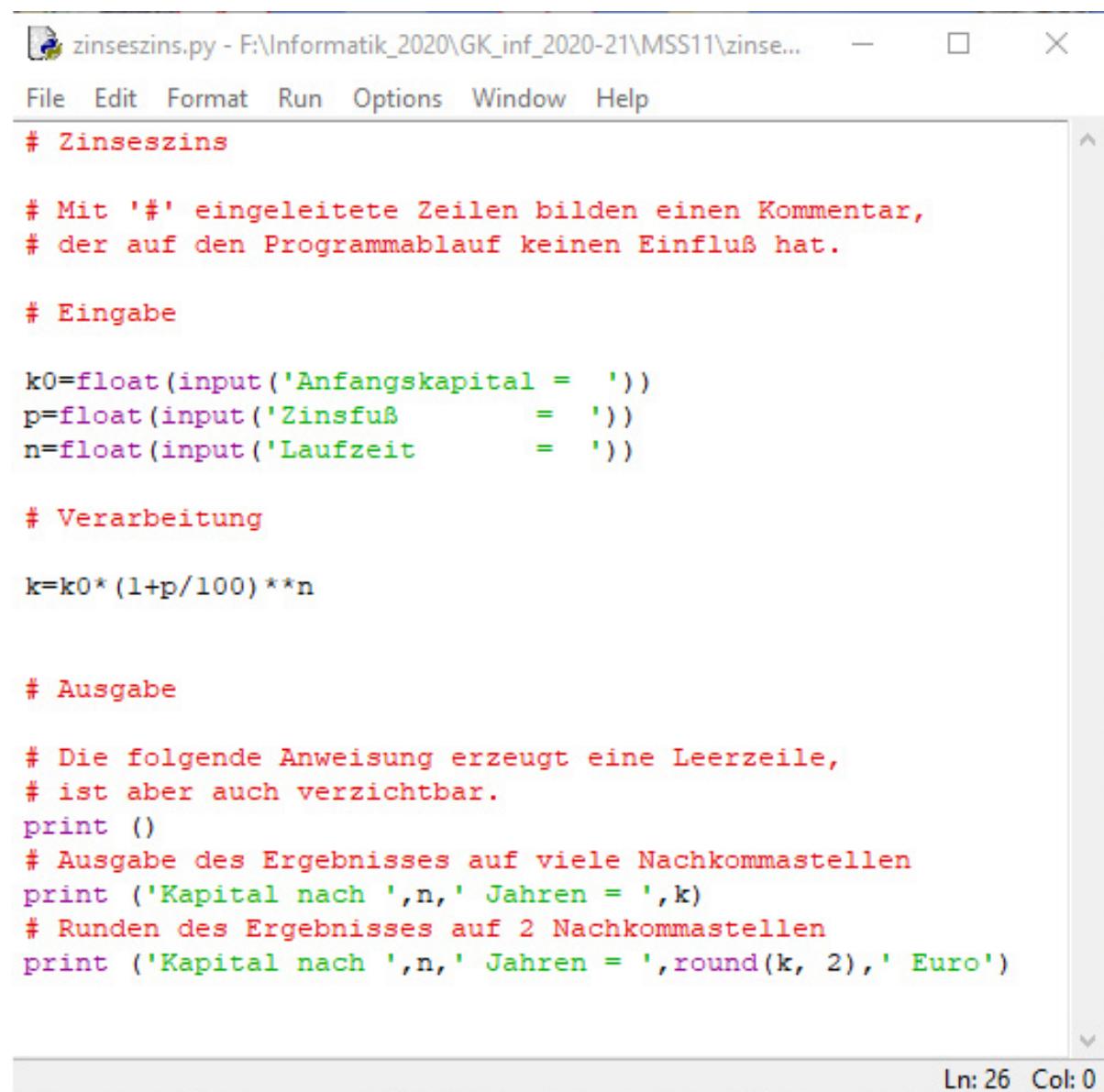
Der Algorithmus „Quaderberechnung“ ermittelt Volumen V und Oberfläche O eines Quaders, nachdem man die Länge a , Breite b und Höhe c eingegeben hat.

Aufgabe 2

Wird ein Anfangskapital k_0 zu einem Zinssatz von p % über eine Laufzeit von n Jahren mit Zinseszins angelegt, verfügt man am Ende der Laufzeit über das Kapital k . Der Algorithmus „Zinseszins“ ermittelt das Endkapital k nach Eingabe des Anfangskapitals k_0 , des Zinsfußes p und der Laufzeit n . (Bemerkung: In entsprechender Weise lässt sich die Entwicklung des Preisindex nach n Jahren bei einer Inflationsrate von p % ermitteln.)

Lösung:

Den Ablauf entnehmen wir obenstehendem Flussdiagramm, welches von der gewählten Programmiersprache unabhängig ist. Wir codieren den Algorithmus in der Programmiersprache „Python“.



```

zinseszins.py - F:\Informatik_2020\GK_inf_2020-21\MSS11\zinse...
File Edit Format Run Options Window Help
# Zinseszins

# Mit '#' eingeleitete Zeilen bilden einen Kommentar,
# der auf den Programmablauf keinen Einfluß hat.

# Eingabe

k0=float(input('Anfangskapital = '))
p=float(input('Zinsfuß      = '))
n=float(input('Laufzeit      = '))

# Verarbeitung

k=k0*(1+p/100)**n

# Ausgabe

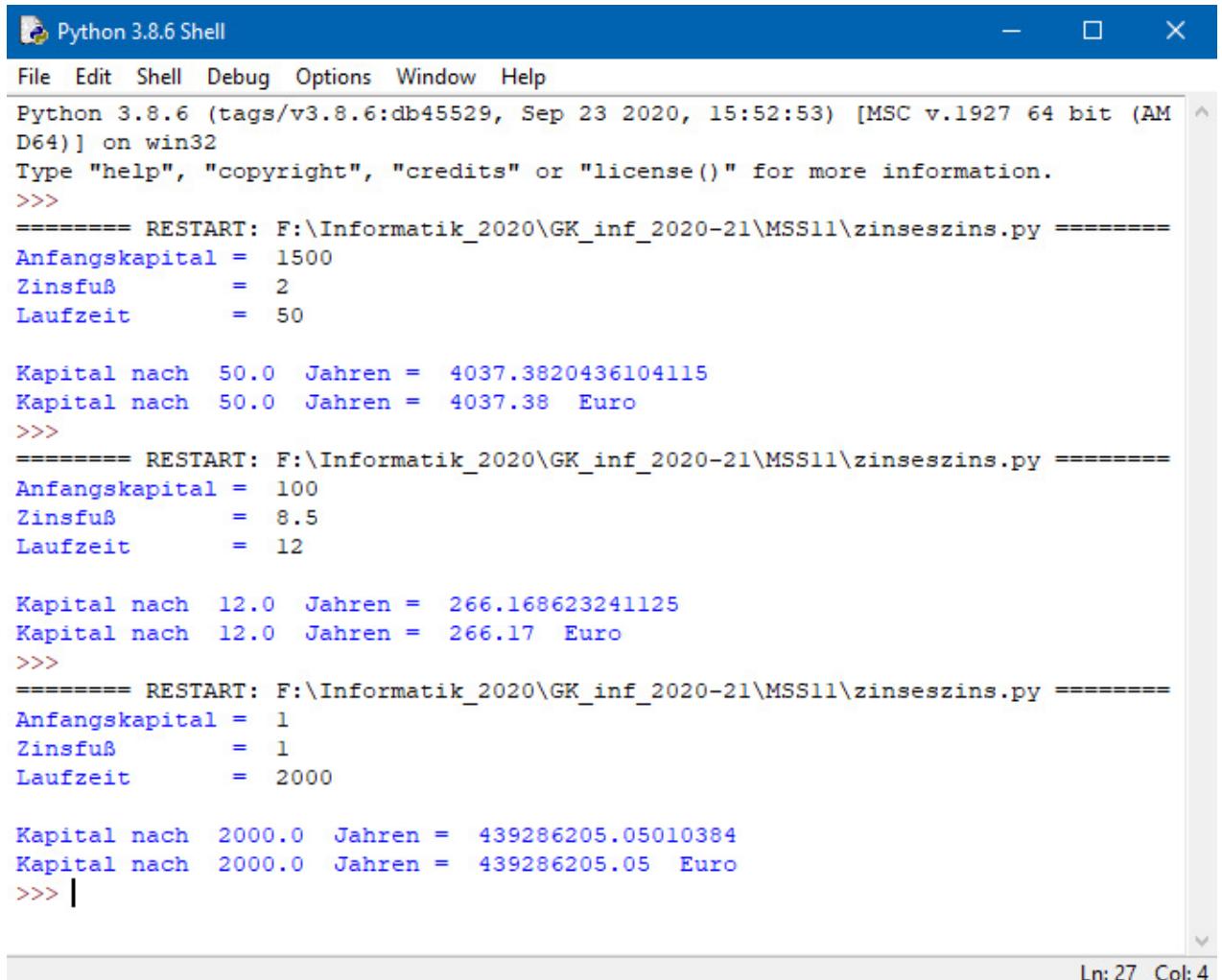
# Die folgende Anweisung erzeugt eine Leerzeile,
# ist aber auch verzichtbar.
print()
# Ausgabe des Ergebnisses auf viele Nachkommastellen
print ('Kapital nach ',n,' Jahren = ',k)
# Runden des Ergebnisses auf 2 Nachkommastellen
print ('Kapital nach ',n,' Jahren = ',round(k, 2), ' Euro')

```

Ln: 26 Col: 0

Wenn wir nach Eingabe des Programmtextes im „IDLE“-Editor den Button „Run“ anklicken, öffnet sich ein Kontextmenue, und wir starten das Programm durch Klick auf „Run Module“.

Nachdem man bestätigt hat, den eventuell geänderten Programmtext zu speichern, öffnet sich die „Python Shell“, in der man die Eingaben macht und in der dann die Ausgabe des Ergebnisses oder der Ergebnisse erfolgt, falls man den Run-Befehl wiederholt erteilt.



```

Python 3.8.6 Shell
File Edit Shell Debug Options Window Help
Python 3.8.6 (tags/v3.8.6:db45529, Sep 23 2020, 15:52:53) [MSC v.1927 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: F:\Informatik_2020\GK_inf_2020-21\MSS11\zinseszins.py ======
Anfangskapital = 1500
Zinsfuß       = 2
Laufzeit       = 50

Kapital nach 50.0 Jahren = 4037.3820436104115
Kapital nach 50.0 Jahren = 4037.38 Euro
>>>
===== RESTART: F:\Informatik_2020\GK_inf_2020-21\MSS11\zinseszins.py ======
Anfangskapital = 100
Zinsfuß       = 8.5
Laufzeit       = 12

Kapital nach 12.0 Jahren = 266.168623241125
Kapital nach 12.0 Jahren = 266.17 Euro
>>>
===== RESTART: F:\Informatik_2020\GK_inf_2020-21\MSS11\zinseszins.py ======
Anfangskapital = 1
Zinsfuß       = 1
Laufzeit       = 2000

Kapital nach 2000.0 Jahren = 439286205.05010384
Kapital nach 2000.0 Jahren = 439286205.05 Euro
>>> |
```

Ln: 27 Col: 4