Aufgabenblatt Nr. 5 inf12 12.01.2022

Übungen zu GUI

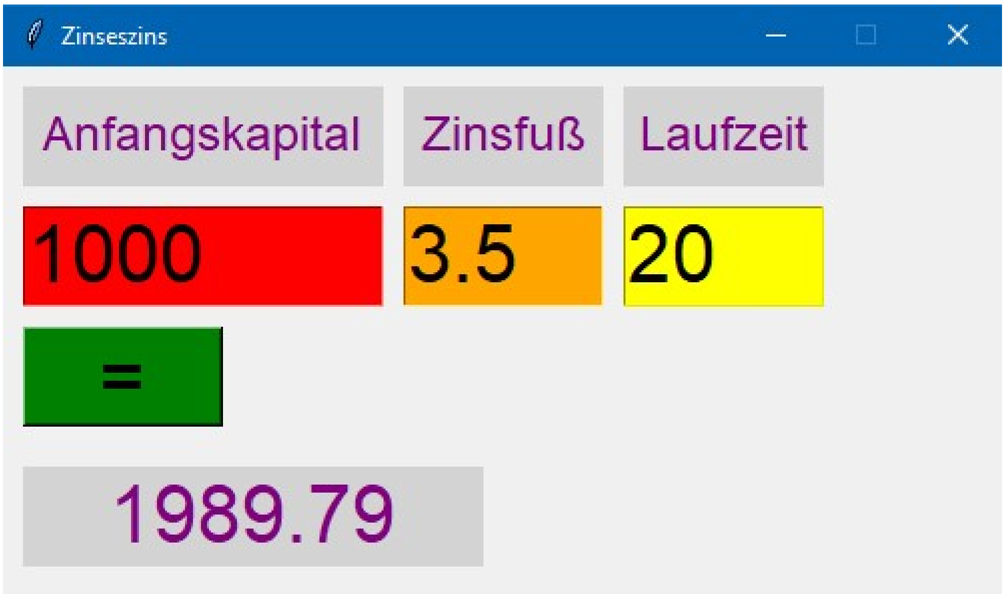
1. Der Algorithmus **ZINZESZINS** berechnet nach Eingabe des Anfangskapitals $\mathbf{K_0}$ (oder: des Preisindex K_0 zu Anfang), des Zinsfußes **p** (oder: der Inflationsrate) und der Laufzeit **n** (in Jahren) das Endkapital $\mathbf{K_n}$ (oder: den Preisindex) nach **n** Jahren gemäß folgender Formel:

$$K_n = K_0 \cdot (1 + p/100)^n$$

Erstelle einen in Python geschriebenen Quelltext mit grafischer Benutzeroberfläche (**g**raphical **u**ser **i**nterface, **GUI**), so daß die Eingabe der Daten und die Anzeige des Ergebnisses sich gestalten z. B. mit einer Benutzeroberfläche wie unten dargestellt.

Hinweise:

- man orientiere sich an dem Quelltext **Grundrechenarten_GUI.py**;
- die Variablen $\mathbf{K_n}$, $\mathbf{K_0}$ und **p** sind vom Typ **`float`**, **n** wahlweise **`int`** oder **`float`**;
- falls das Ergebnis eine Gleitkommazahl (also vom Typ **`float`**) ist, rundet die Python-Anweisung **`round(result,2)`** kaufmännisch auf zwei Nachkommastellen.



2. Der **Anhalteweg eines Fahrzeugs** ergibt sich als Summe von Reaktionsweg und Bremsweg:

Anhalteweg = Reaktionsweg + Bremsweg

Mit den Vereinbarungen

v_0 = Geschwindigkeit, aus der der Bremsvorgang eingeleitet wird (in m/s)

a = Bremsverzögerung (in m/s^2)

t_R = Reaktionszeit (in s)

gilt:

Reaktionsweg = $v_0 \cdot t_R$

Bremsweg = $\dfrac{v_0^2}{2 \cdot a}$

Schreibe ein Python-Programm mit grafischer Benutzeroberfläche GUI für den Algorithmus **ANHALTEWEG**, der nach Eingabe

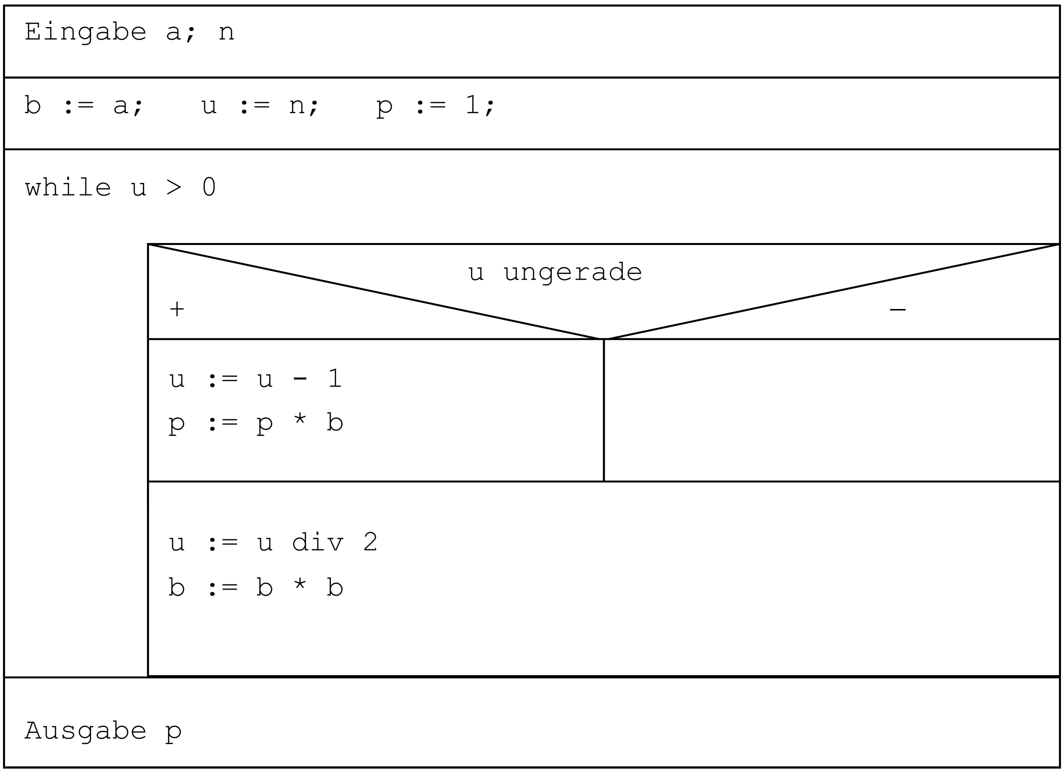
- der Geschwindigkeit v_0 (in km/h; beachte: 1 m/s = 3,6 km/h),
- der Reaktionszeit t_R (wähle t_R = 0,6 . . . . 1 s),
- des Straßenzustands

den Anhalteweg bei einer Gefahrenbremsung berechnet und ausgibt.
Der Straßenzustand bestimmt wesentlich die mögliche Bremsverzögerung a.

Annahmen:
trockene Fahrbahn: a = g = 9,81 $m/s^2 \approx 10\ m/s^2$
regennasse Fahrbahn: a = 0,7 g
schneebedeckte Fahrbahn: a = 0,2 g
vereiste Fahrbahn: a = 0,05 g

Korrektheit von Algorithmen

3. n sei eine natürliche Zahl, a eine von 0 verschiedene reelle Zahl.
   Gegeben ist folgender Algorithmus als Struktogramm:

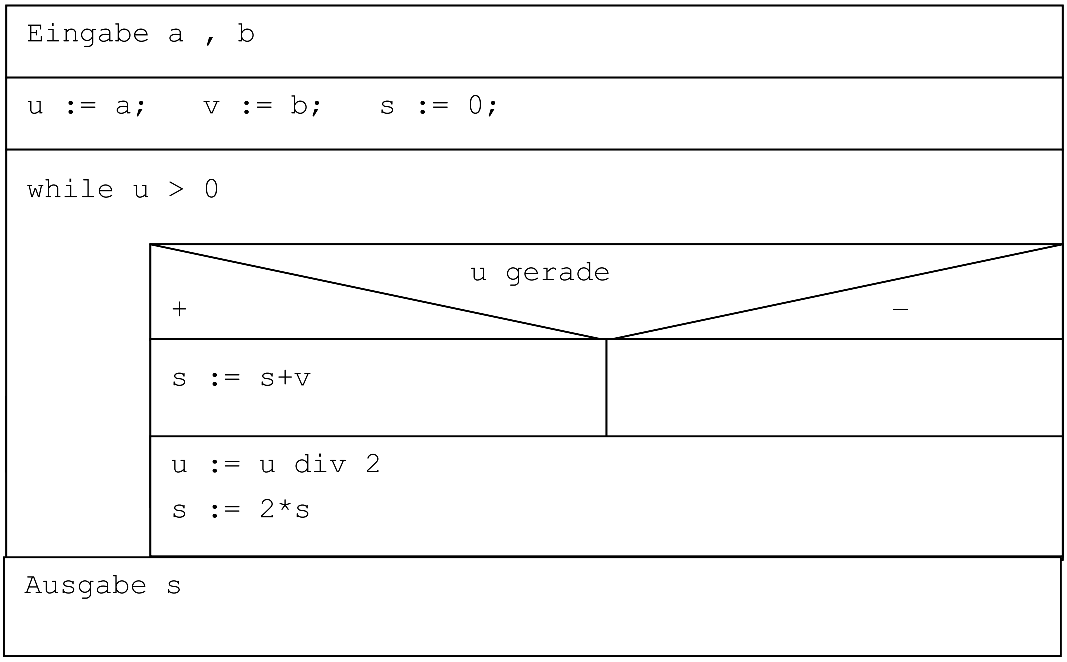


a) Codiere den Algorithmus in Python.
b) Teste das Programm; was bewirkt der Algorithmus vermutlich?

c) Führe für n = 7, n = 18, n = 52 jeweils einen Trace durch, um die Vermutung zu erhärten.

d) Formuliere eine Beziehung, die sich als Schleifeninvariante erweist, und bestätige unter Einbeziehung der Abbruchbedingung, daß der Algorithmus tatsächlich die Potenz $\mathbf{a^n}$ berechnet und ausgibt.

4. In einem Lehrbuch ist das Struktogramm des folgenden Algorithmus abgedruckt, von dem behauptet wird, daß er das Produkt der natürlichen Zahlen a und b berechne (dieses – im übrigen nicht schlechte – Buch gibt's tatsächlich!):



a) Verifizieren anhand eines Trace (oder indem man das Python-Programm schreibt und dieses testet), daß der Algorithmus das verlangte nicht leistet.

b) Korrigiere den Algorithmus, so daß er korrekt im Sinne der Spezifikation arbeitet; erstelle in geeigneter Weise Trace-Tabellen. Für Fortgeschrittene: beweise die Korrektheit vermöge vollständiger Induktion.