Erinnerung Struktogramm

Struktogramme dienen der übersichtlichen Darstellung von Problemen, Algorithmen oder Funktionen. Dazu werden sie in ihre elementaren Grundstrukturen zerlegt (Anweisungen, Kontrollstrukturen). Folgende Richtlinien gelten bei der Erstellung:

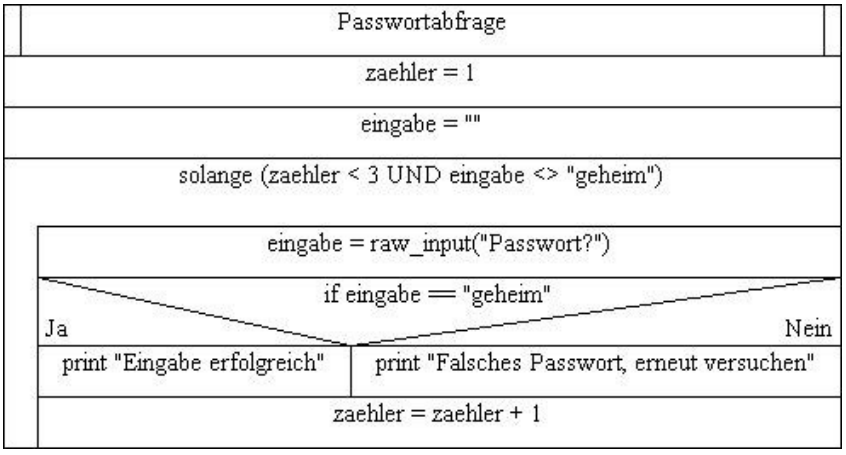
- für jede Anweisung ein Rechteck
- Schleifen bestehen aus Kopf- und Anweisungsteil
  - Kopf bei Zählschleifen: Laufvariable, Anfangswert , Solange-Bedingung, Schrittweite
  - Kopf bei bedingten Schleifen: solange-Bedingung
- bedingte Anweisungen bestehen aus Bedingungs- und Anweisungsteil für Erfüllung, sowie Nichterfüllung

Beispiel

Quellcode

```
def passwortabfrage()
  zaehler = 1
  eingabe = „“
  while ((zaehler < 3) and (eingabe <> „sicher“)):
    eingabe = raw_input(„Passwort?“)
    if (eingabe == „geheim“):
      print(„Eingabe erfolgreich“)
    else
      print(„Falsches Passwort, erneut versuchen“)
    zaehler = zaehler +1
```

Struktogramm



Aufgabe 1

Erstelle ein Struktogramm für die Funktion `caesar(klartext, schluessel)` vom letzten AB . Zeige deine Lösung am Lehrerpult vor bevor Du weiterarbeitest.

Wie sicher ist das Caesar-Verfahren gegen Eves Angriffe?

Aufgabe 2

Die Datei `03_krypto.py` enthält die Funktionen `caesar` und `decaesar` und eine globale Variable `geheimeBotschaft`. Eve möchte den Inhalt der Botschaft ermitteln, und nicht nur das, sie möchte eine python-Funktion haben, die ihr das Entziffern aller geheimen Botschaften erleichtert.

Die Brute-Force-Herangehensweise: Der zu entziffernde Text wird mit Hilfe der schon fertigen `decaesar`-Funktion nacheinander mit jedem Schlüssel, beginnend bei 1, entschlüsselt.

a. Führe das Verfahren mit Hilfe der `decaesar`-Funktion aus und notiere die ersten zwei und den letzten Funktionsaufruf.

Für clevere Faule: Formuliere die Aufrufe in einer Zählschleife. Notiere diese Anweisung.

b. Definiere eine Funktion `caesarBruteForce`, welche die Schleife ausführt. Du hast dafür mehrere Möglichkeiten:

- Verwende die `print`-Funktion, um die Ergebnisse von `decaesar` auszugeben.
- Eleganter ist die Sammlung der einzelnen Entschlüsselungsversuche in einer Variable `gesamttext`, die dann über die `return`-Anweisung zurückgegeben wird. Informiere dich dazu im Quelltext der `caesar`-Funktion über die Initialisierung, Zuweisung in der Schleife und Rückgabe der Variable `geheimtext`.

c. Schätze die Sicherheit des Caesar-Verfahrens ein:

Alice und Bob wollen während der Klassenarbeit ihre Ergebnisse zu einer Aufgabe vergleichen. Sie wollen Eve daran aber nicht teilhaben lassen. Eve ist bereit, die Nachricht weiterzugeben, da sie ihr Smartphone mit der Brute-Force-App dabei hat.

Unterscheide bei der Einschätzung der Sicherheit des Caesar-Verfahrens, ob Eve es schafft, ihr Smartphone zu verwenden.

Aufgabe 3 (nach erfolgreichem Finden der eleganten Lösung in 1b.)
Teil der Häufigkeitsanalyse: einzelne Zeichen zählen

Entwickle und implementiere einen Algorithmus für das Zählen eines bestimmten Zeichens in einem Text und notiere den Quelltext der fertigen Funktion:

```
def ermittleHaeufigkeit(text, zeichen):
    ...
    return haeufigkeit
```