



Informatik für Schüler, Foliensatz 14 Wiederholung und Weiterführung 2D-Zeichenfeld

Prof. G. Kemnitz

Institut für Informatik, Technische Universität Clausthal
9. Juni 2010



Wiederholung

- Welchen Typ erhält die Variable »f« in der folgenden Anweisung?

```
f=open('test.txt', 'w')
```

- Was ist an der folgenden Anweisung falsch? Welchen Typ würde die Variable »s« erhalten, wenn die Anweisung ausführbar wäre?

```
s=f.read()
```

- Wie wird eine Datei zum Lesen geöffnet?
- Wie kann man eine Textdatei als eine Liste von Zeilen einlesen?
- Eine Zeichenkette in einer Variablen z unbekannter Länge soll auf das Vorhandensein des Zeichens 'a' getestet werden. Programmvorschlag? Definition als Funktion?



Sichtbarkeit

- Ein Programmobjekt (Datenobjekte oder Funktion) ist in einer Programmeinheit sichtbar, wenn der Name ohne Übersetzungsfehler genutzt werden kann.

Fact 1

Jedes genutzte Datenobjekt muss vor seiner Verwendung definiert sein.

- Variablen durch eine Wertezuweisung
- Funktionen werden mit dem Schlüsselwort »def«



Welche Objekte werden in dem folgenden Programmfragment unerlaubterweise vor ihrer Definition verwendet?

```
x=test()  
def test():  
    return 0  
print test()  
y=(x+1)/(2*y)
```

Experiment: Testen Sie, welchen Datentyp eine Funktion ohne Rückgabe-Anweisung hat?

```
def test():  
    pass  
type(test())
```



Lokale und globale Variablen

- Innerhalb einer Funktionsdefinition definierte Datenobjekte werden als lokale Variablen bezeichnet. Sie sind nur innerhalb der Funktionsdefinition sichtbar.
- Außerhalb einer Funktionsdefinition definierte Datenobjekte sind global. Später definierte globale Datenobjekte sind unsichtbar. Vor der Funktionsdefinition definierte globale Datenobjekte sind innerhalb der Funktionsdefinition nur lesbar.
- Namensgleiche lokale Datenobjekte verdecken die Sichtbarkeit globaler Datenobjekte.



Welche Variablen sind lokal, welche global und welche nicht definiert?

```
x=10
def test():
    s=''
    for idx in range(x):
        s += char(48+idx)
    x=x+1
    return s
#-----
r = test();
r += s
```




Aufgabe 14.2: Schachbrett

Das Programm soll auf dem Terminal ein Schachbrett ausgeben. Jedes der 8×8 Felder soll drei Zeichen hoch und fünf Zeichen breit sein. Die schwarzen Felder sind mit '*' und die weißen Felder mit '.' zu füllen.



Aufgabe 14.3: Brückenproblem mit Visualisierung

Das Brückenüberquerungsproblem aus Aufgabe 12.2 soll wie folgt modifiziert werden. Die Brücke soll statt 3 jetzt 6 Felder breit und statt zehn 20 Felder lang sein. Die Wahrscheinlichkeiten und der Rest sollen gleich bleiben. Ändern sie die Lösung entsprechend ab.

- Ändern sich die Wahrscheinlichkeit, dass der Proband ins Wasser fällt erheblich?

- Stellen Sie die Brücke graphisch als 2D-Feld aus zwei Ziffern großen Zahlen, die durch Leerzeichen getrennt sind, graphisch dar. Bevor der Proband die Brücke betritt, soll das Feld ausgedruckt wie folgt aussehen:

```
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

- Visualisieren sie den Weg des Probanden, indem Sie in jedem Feld, das er betritt, den Wert um Eins erhöhen.
- Lassen Sie den Probanden $100\times$ über die Brücke gehen und zählen Sie dabei wie oft die einzelnen Felder insgesamt betreten werden. Gibt es Felder, die der Proband bevorzugt betritt oder meidet?

