PROPÄDEUTIKUM INFORMATIK SOSE 2018

Martin Mehlhose

INFORMATIONSVERANSTALTUNG

- Informationsveranstaltung zur Bewerbung und Studium an der TU Dresden
- **22. Mai 13:30 uhr** im Zimmer 27

LISTEN

LISTEN

- Listen dienem dem "effizientem" Speichern großer Datenmengen
- Listen werden durch den [] Operator markiert. Zum Beispiel [1,2,3,4,5,6,7]
- Die Daten müssen nicht alle den selben Typ haben

LISTEN

- Zugriff auf Listen Elemente über den Index. Achtung Listen sind 0 basiert!
- Zugriff erfolgt dann über eckige Klammern, z.B. list[0] für das erste Listenelement
- Die Elemente können über den Index sowohl ausgelesen als auch bearbeitet werden
- sollen alle Elemente einer Liste bearbeitet werden, dass am besten eine for-Schleife benutzen

LISTEN

- Wichtige Funktionen:

- len(list) : gibt sie Länge der Liste zurück
- mylist.pop(i) : entfernt das i'te Element aus der Liste
- mylist.append(i) : hängt i an das Ende der Liste an
- Zusammenfügen von zwei Listen über + Operator

LISTEN

- Achtung: Listen sind Referenzdatentypen
- -> wir können Listen nicht einfach kopieren
- -> in Funktionen geänderte Listen brauchen nicht über return zurückgegeben werden

- Vorsicht beim kopieren von Listen !

STRINGS

STRINGS

- String ist ein Datentyp der Zeichenketten speichern kann
- Einer der wichtigsten Datentypen
- Anlegen eines Strings durch ""

FORMATIERUNG

- Einfache Formatierungsaufgaben können durch Steuerzeichen erledigt werden
- Einrücken um einen Tabulator : **\t**
- Zeilenumbruch: **\n**
- Soll in einem String ein " ausgegeben werden, muss dies **\"** geschrieben werden
- Soll ein \ geschrieben werden, so muss/sollte dies \\ geschrieben werden

WICHTIGE FUNKTIONEN

- String build-in Funktionen werden immer über Punktnotation aufgerufen
- .count("xy") liefert die Anzahl der Vorkommen von "xy" im String
- .find("xy") liefert den ersten Index im String, an dem "xy" steht
- .split("xy") trennt den String an den Stellen xy und liefert eine Liste der einzelnen Komponenten
- .replace("alt", "neu") eretzt alle vorkommen von "alt" in "neu"

WICHTIGE FUNKTIONEN

- .lower() liefert den String in Kleinbuchstaben
- .upper() liefert den String in Großbuchstaben
- .isDigit() liefert True, wenn der String nur aus Zahlen besteht

MASCHINENZAHLEN

MASCHINENZAHLEN

- Wir rechnen hier mit 8 bit Zahlen
- Also Zahlen im Bereich von -128 bis + 127
- Umwandlung Dez in Bin durch iterarives teilen durch 2 und merken der Reste

MASCHINENZAHLEN

- Umrechnung von Bin nach Dec durch Multiplikation der letzten 7 bit mit ihrer jeweiligen Wertigkeit
- Beispiel: 0 0 0 0 1 0 1 0

MASCHINENZAHLEN

- Umrechnung von Bin nach Dec durch Multiplikation der letzten 7 bit mit ihrer jeweiligen Wertigkeit
- Beispiel: 0 0 0 0 1 0 1 0
- Lösung: $0*2^0 + 1*2^1 + 0*2^2 + 1*2^3 = 2 + 8 = 10$

MASCHINENZAHLEN

- Negative Zahlen
- Umrechnung des Betrags ins Binärsystem
- jedes Bit invertieren
- + 1 rechnen
- Beispiel:10 = 0 0 0 0 1 0 1 0
- Jedes Bit invertieren: 1 1 1 1 0 1 0 1
- 1 addieren: 1 1 1 1 0 1 1 0 = -10

KOMMAZAHLEN

- bei der Umrechnung Vor- und Nachkommateil getrennt betrachten
- Vorkommateil wird ganz normal umgerechnet
- Nachkommateil durch iteratives multiplizieren mit 2
- Beispiel:

KOMMAZAHLEN

- Mit dem Verfahren erhalten wir allerdings noch keine Gleitkommazahlen
- https://de.wikipedia.org/wiki/Gleitkommazahl
- https://www.itu.dk/~sestoft/bachelor/IEEE754_article.pdf

DATEIEN

DATEIEN

- Dateien werden mit der Funktion open("Dateiname", "Option") geöffnet
- Option kann "r" zum lesen oder "w" zum schreiben sein
- weitere Optionen verfügbar, aber hier erstmal uninterresant
- Die Datei kann dann zeilenweise über einer for-Schleife ausgelesen werden.

DATEIEN

- Schreiben in eine Datei mit der Funktion write()
- **Wichtig:** am Ende des Programms müssen die Filestreams mit der Funktion close() geschlossen werden
- Sonst droht Datenverlust

SUCHEN UND SORTIEREN

SUCHEN UND SORTIEREN

- Zum effenktivem Suchen ist eine Sortierung auf der Liste notwendig
- Auf unsortierten Listen bleibt nur die Möglichkeit jeden Schlüssel einzeln zu prüfen
- Wollen wir viel Suchen lohnt es sich die Liste vorher zu sortieren

INSERTION-SORT

- Sortieren der Liste L1:
- Lege eine leere Liste L2 an.
- Suche das kleinste Element in L1 und füge es in L2 ein
- Suche das zweit kleinste Element in L1 und füge es in L2 ein
- usw...

BUBBLE-SORT

- Durchlaufe Liste L und vertausche jeweils benachbarte Elemente, die nicht in Sortierordnung sind.
- Führe dies solange durch, bis kein Umtauschen meht nötig ist.

- Etwas effizienter, da in situ und die Vorsortierung teilweise genutzt werden kann.

SORTIERVERFAHREN

- Weitere wesentlich effizientere Sortierverfahren welche aber mit Recursion arbeiten
- z.B. Merge-Sort, Quick-Sort
- Trotzdem lohnen sich Sortierverfahren oft nur, wenn anschließend sehr viel auf der Liste gesucht wird.
- Alternative ist ein Binärer Suchbaum.