

## Übungsblatt Nr. 2

### 1 Elementare Datentypen und Kontrollstrukturen

#### 1.1 Auswertungsreihenfolge

Was ist das Ergebnis und welchen Datentypen hat dieses? Begründen Sie dies, indem Sie die konkrete Abarbeitungsreihenfolge der Operatoren angeben.

**Hinweis:** Die Operatorreihenfolge findet sich auf dem Programmierhandzettel 1

**a)  $5 ** 2 + 12 * 2 > (3 - 2) * 10 \% 5$**

1.  $(.....) \rightarrow (3 - 2) \rightarrow 5 ** 2 + 12 * 2 > 1 * 10 \% 5$
2.  $x ** y \rightarrow 5 ** 2 \rightarrow 25 + 12 * 2 > 1 * 10 \% 5$
3.  $* \rightarrow 12 * 2 \rightarrow 25 + 24 > 1 * 10 \% 5$
4.  $* \rightarrow 1 * 10 \rightarrow 25 + 24 > 10 \% 5$
5.  $\% \rightarrow 10 \% 5 \rightarrow 25 + 24 > 0$
6.  $+ \rightarrow 25 + 24 \rightarrow 49 > 0$
7.  $> \rightarrow 49 > 0$
8. **Ergebnis:** True
9. **Datentypen:** Bool.

**b)  $3 * 4 // -19 // 3 / 4$**

1.  $-x \rightarrow 3 * 4 // -19 // 3 / 4$
2.  $* \rightarrow 3 * 4 // -19 // 3 / 4$
3.  $/ \rightarrow 3 * 4 // -19 // 3 / 4$
4.  $// \rightarrow 3 * 4 // -19 // 3 / 4$
5.  $// \rightarrow 3 * 4 // -19 // 3 / 4$
6. **Ergebnis:** -2.0
7. **Datentypen:** Float

#### 1.2 Eigenrecherche

Nehmen Sie an, Sie haben x und y in Python definiert. Erklären Sie den Unterschied zwischen „ $x == y$ “ und „ $x \text{ is } y$ “?

**Antwort:**

```
a = [1, 2, 3]
b = a
b is a
True
b == a
True
b = a[:]
b is a
False
b == a
True
```

Anhand des Beispiels kann man verstehen, dass „==“ eine Operation vergleicht, zwischen zwei Werte aber „is“ testet auf Identität.

## 1.3 Algorithmen

Beurteilen Sie, ob die folgenden Bearbeitungsvorschriften Algorithmen sind und argumentieren Sie:

a) Vorschrift a

1. Nehmen Sie eine beliebige Zahl x.
2. Zählen Sie die Sekunden der aktuellen Tageszeit seit 00:00:00 h hinzu.
3. Addieren Sie 7 hinzu und multiplizieren Sie mit 5.

**Antwort:**

- read(x)
- time\_now ← detetime.now.time()
- hours ← time\_now.getHours
- moments ← time\_now.getMoments
- seconds ← time\_now.getSeconds
- seconds ← seconds + moments \* 60 + hours \* 3600
- x ← x + seconds
- x ← x + 7
- x ← x \* 5
- print(x)

Es wird durch **Pseudocode** geschrieben.

b) Vorschrift b

1. Nehmen Sie eine Zahl x ungleich 0.
2. Entweder addieren Sie das Dreifache von x zu x und teilen das Ergebnis durch x.
3. Oder: Subtrahieren Sie 4 von x und subtrahieren das Ergebnis von x.

**Antwort:**

- Read(x)
- While(x=0)
  - Print(Gaben Sie eine Zahl ungleich 0)
  - Read(x)
- Y = 3 \* x
- Y = x + y
- Y = y / x
- Print(y)

Es wird durch **Pseudocode** geschrieben

## 1.4 Programmiersprachen

Worin besteht der wesentliche Unterschied zwischen eine compilierten und einer interpretierten Programmiersprache? Gibt es auch Mischformen? Recherchieren und beschreiben Sie in maximal jeweils 5 Sätzen die jeweiligen Vor- und Nachteile!

Antwort:

• **Der Unterschied:**

- Interpreter lest Zeile nach Zeile und er stoppt, wenn er einen Fehler findet.
- Compiler lest den ganzen Code und er zeigt alle Fehler.

- Java ist Mischformen
- **Vor- und Nachteile:**
  - **Interpreter:**
    - Er übersetzt eine Anweisung zu einer Zeit.
    - Das Übersetzen des Programms wird fortgesetzt, bis der erste Fehler auftritt. In diesem Fall wird es angehalten.
    - Das Debuggen ist einfach
    - Die Analyse dauert weniger Zeit, die Gesamtausführungszeit ist langsamer.
    - Es wird kein Zwischenobjektcode generiert, da sie speichereffizient sind.
  - **Compiler:**
    - Die Fehlermeldung wird erst nach dem Scannen des gesamten Programms generiert.
    - Die Analyse des Quellcodes erfordert viel Zeit, die Gesamtausführungszeit ist jedoch vergleichsweise kürzer.
    - Er scannt das gesamte Programm und übersetzt es als Ganzes in Maschinencode.
    - Erzeugt einen Zwischenobjektcode, der außerdem eine Verknüpfung erfordert und daher mehr Speicher benötigt.
    - Debugging ist vergleichsweise schwer.