

Praktikum zu  
**Einführung in die Informatik für  
LogWings und WiMas**  
Wintersemester 2014/15

**Übungsblatt 13**

Bearbeitungszeit:  
2.02.15 - 6.02.15

**Aufgabe 13.1 – Bäume**

**Bearbeitungszeit: 5 Minuten**

Wofür werden binäre Suchbäume verwendet?

---

Welche Operationen sind in ausbalancierten binären Suchbäumen schneller als in geordneten Listen (ohne Fußelement)?

---

Welche Eigenschaft hat ein In-Order-Durchlauf eines binären Suchbaumes?

---

Welches Problem kann auftreten, wenn die in einen binären Suchbaum eingefügten Elemente bereits sortiert waren?

---

**Aufgabe 13.2 – Syntaxdiagramme**

**Bearbeitungszeit: 15 Minuten**

Welche der folgenden arithmetischen Ausdrücke sind nach dem unten stehenden Syntaxdiagramm syntaktisch korrekt und welche sind falsch? Geben Sie zu jedem Ausdruck den entsprechenden Durchlauf im Diagramm an.

Beispiel:

$-5 * (3 ++7)$

$\alpha$  3 A 4  $\beta$  4  $\varphi$  13 6 7  $\alpha$  1 A 4  $\beta$  4  $\varphi$  10 5 2 A

---

korrekt



6.0E1

$\alpha$  1 A 4  $\beta$  1 2 4 5 8  $\varphi$   $\omega$

---



korrekt

a)  $-14 \% 2 * 3$



---

---

b)  $0xF5 - 0$



---

---

c)  $0b+5$



---

---

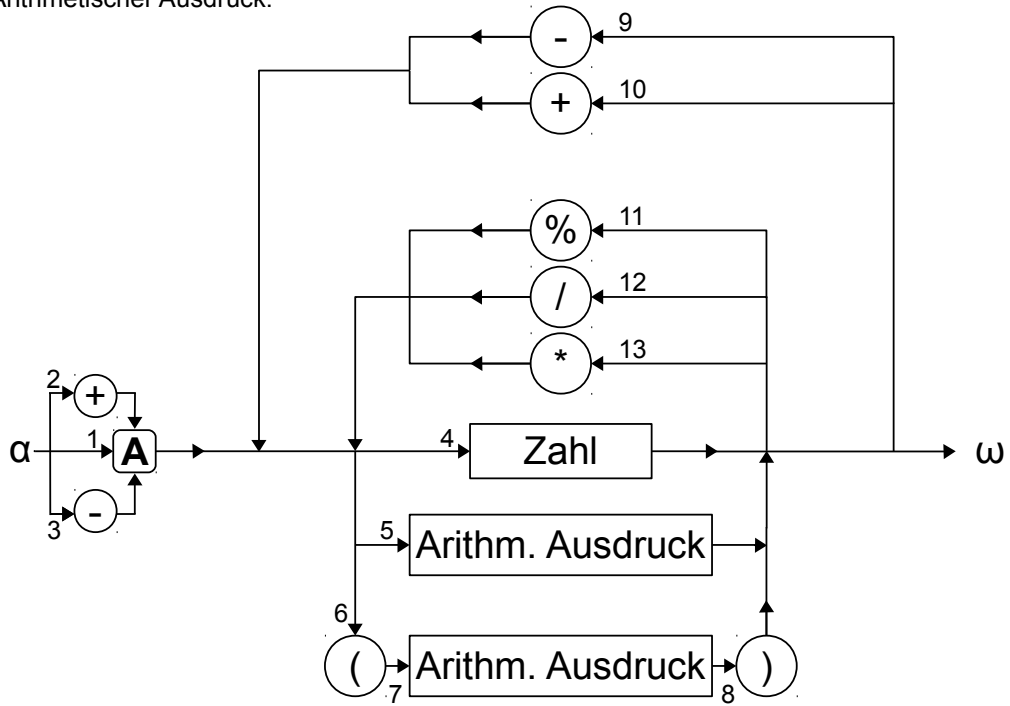
d)  $(.7 * 5 - 4)$



---

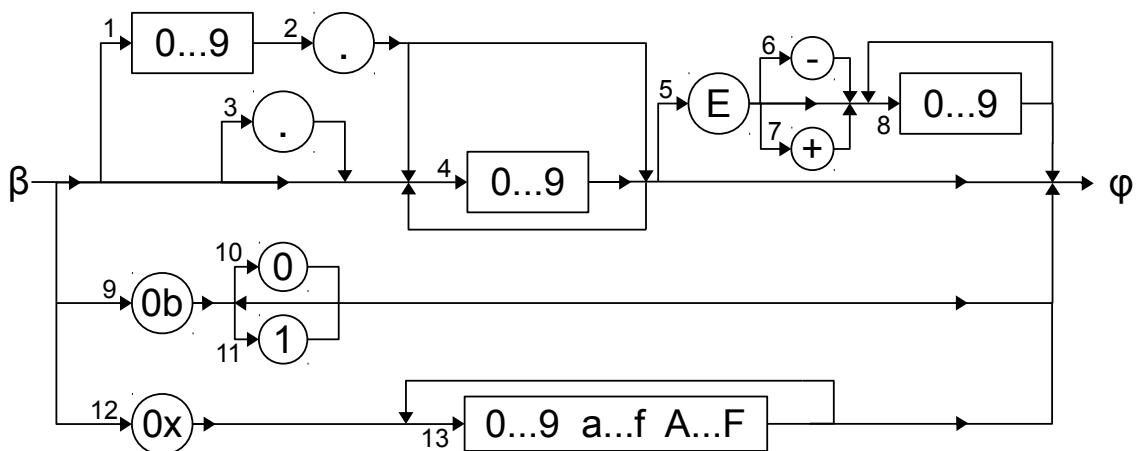
---

Arithmetischer Ausdruck:



A: Alle „-“ und „++“ müssen durch ein Leerzeichen getrennt werden.

Zahl:



### Aufgabe 13.3 – Baumdurchlauf

Bearbeitungszeit: 20 Minuten

Ein Knoten eines Binärbaums wird durch folgende Klasse repräsentiert:

```
class Node {
    int value;
    Node leftChild;
    Node rightChild;
}
```

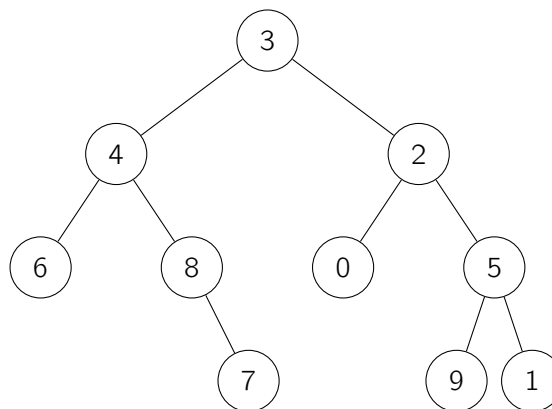
Wird die folgende Methode für die Wurzel eines solchen Baumes aufgerufen, so gibt sie die Werte (Markierungen) aller Knoten im Baum aus:

```
void print(Knoten k) {
    if (k == null) {
        return;
    }
    print(k.leftChild);
    print(k.rightChild);
    System.out.print(k.value + " ");
}
```

a) Welche Art von Baumdurchlauf realisiert die Methode **ausgeben**?

---

Betrachten Sie folgenden Binärbaum:



b) Geben Sie die Pre-Order-Reihenfolge für obigen Baum an:

---

c) Geben Sie die Post-Order-Reihenfolge für obigen Baum an:

---

d) Geben Sie die In-Order-Reihenfolge für obigen Baum an:

---

## Ergänzende Aufgaben

### Aufgabe 13.4 – Objektorientierung

**Bearbeitungszeit: 40 Minuten**

Betrachten Sie die folgende Klasse, die ein Produkt eines Supermarkts modelliert. Jedes Produkt hat einen Preis und eine Anzahl, wie oft es vorrätig ist.

```
class Product {  
    private double price;  
    private int stock;  
    public Product(double price) {...}  
    public double getPrice() {...}  
    public void buy(int amount) {...}  
    public boolean sell(int amount) {...}  
}
```

- a) Implementieren Sie den Konstruktor, der ein neues Produkt mit dem gegebenen Preis erzeugt. Das Produkt soll zunächst nicht vorrätig sein.
- b) Implementieren Sie die Methode **getPrice**, die den Preis des Produkts zurückgibt.
- c) Implementieren Sie die Methoden **buy** und **sell**, mit denen der Supermarkt das Produkt in der gegebenen Anzahl einkaufen oder verkaufen kann. Ist das Produkt beim Verkauf nicht in ausreichender Zahl verfügbar, soll nichts verkauft werden. Dies soll durch die Rückgabe des Wahrheitswerts **false** signalisiert werden.
- d) Schreiben Sie ein Programmfragment, das zwei Produkte — **flour** zu 1,99 EUR und **egg** zu 0,10 EUR — anlegt und dann 10 bzw. 800 Stück einkauft.
- e) Gehen Sie anschließend davon aus, dass ein Kunde für einen Kuchen eine Packung Mehl und acht Eier einkauft (die restlichen Zutaten für einen anständigen Kuchen sollen hier nicht beachtet werden). Berechnen Sie den Preis, den ein Produkt namens **cake** höchstens haben darf, damit es aus Sicht des Kunden finanziell attraktiver ist, dieses im Supermarkt zu kaufen. Weisen Sie dem Produkt dabei den höchst sinnvollen Preis zu.