

## Anwendung der Java-Klasse BinaryTree

Die beiden folgenden Methoden erstellen jeweils einen Baum mit Hilfe der Klasse BinaryTree:

```
public BinaryTree<String> getBaum1() {
    BinaryTree<String> baum = new BinaryTree<String>(new String("*"));
    baum.getLeftTree().setContent(new String("7"));
    baum.getRightTree().setContent(new String("+"));
    baum.getRightTree().getLeftTree().setContent(new String("8"));
    baum.getRightTree().getRightTree().setContent(new String("-"));
    baum.getRightTree().getRightTree().getLeftTree().setContent(new String
        ("7"));
    baum.getRightTree().getRightTree().getRightTree().setContent(new
        String("/"));
    baum.getRightTree().getRightTree().getRightTree().getLeftTree().
        setContent(new String("6"));
    baum.getRightTree().getRightTree().getRightTree().getRightTree().
        setContent(new String("3"));
    return baum;
}
```

```
public BinaryTree<String> getBaum2() {
    BinaryTree<String> baum = new BinaryTree<String>(new String("+"));
    baum.getLeftTree().setContent(new String("*"));
    baum.getRightTree().setContent(new String("-"));
    baum.getLeftTree().getLeftTree().setContent(new String("-"));
    baum.getLeftTree().getRightTree().setContent(new String("2"));
    baum.getRightTree().getLeftTree().setContent(new String("+"));
    baum.getRightTree().getRightTree().setContent(new String("7"));
    baum.getLeftTree().getLeftTree().getLeftTree().setContent(new String("
        3"));
    baum.getLeftTree().getLeftTree().getRightTree().setContent(new String("
        5"));
    baum.getRightTree().getLeftTree().getLeftTree().setContent(new String("
        3"));
    baum.getRightTree().getLeftTree().getRightTree().setContent(new String
        ("2"));
    return baum;
}
```

## Aufgaben

1. Zeichnen Sie die beiden Bäume auf, die durch die Methoden erstellt wurden.
2. Erstellen Sie eine Methode, die einen solchen Baum In-Order traversiert.
3. Erweitern Sie eine Kopie dieser Methode so, dass jede Rechnung eines Teilbaumes in Klammern steht.



## Lösungen:

Aufgaben:

1.

```
2. public String inOrderTraversierung(BinaryTree<String> baum) {
    String ausgabe = new String();
    String content = baum.getContent();
    if (!baum.getLeftTree().isEmpty()) {
        ausgabe = ausgabe + this.inOrderTraversierung(baum.getLeftTree()
        );
    }
    ausgabe = ausgabe + content;
    if (!baum.getRightTree().isEmpty()) {
        ausgabe = ausgabe + this.inOrderTraversierung(baum.getRightTree
        ());
    }
    return ausgabe;
}
```

```
public String inOrderTraversierungKlammern(BinaryTree<String> baum)
{
    String ausgabe = new String();
    String content = baum.getContent();
    if (content.contentEquals("*") || content.contentEquals("+") ||
        content.contentEquals("-") || content.contentEquals("/")) {
        ausgabe = ausgabe + "(";
    }
    if (!baum.getLeftTree().isEmpty()) {
        ausgabe = ausgabe + this.inOrderTraversierungKlammern(baum.
        getLeftTree());
    }
    ausgabe = ausgabe + content;
    if (!baum.getRightTree().isEmpty()) {
        ausgabe = ausgabe + this.inOrderTraversierungKlammern(baum.
        getRightTree());
    }
    if (content.contentEquals("*") || content.contentEquals("+") ||
        content.contentEquals("-") || content.contentEquals("/")) {
        ausgabe = ausgabe + ")";
    }
    return ausgabe;
}
```

