



Die Klasse Stack in Java

Anwendung „UPN-Ausdrücke“



Beispiel
 $(4 + 7) * (9 - 3)$

47+ 93- *

„Bei der UPN werden erst die Operanden notiert und dann der zugehörige Operator!“

weitere Beispiele:

„normale“ Notation

UPN

$1 + 2$

1 2 +

$(3 + 24) * 4$

3 24 + 4 *

$3 + (24 * 4)$

3 24 4 * +

$(12 + 17) * (19 - 2)$

12 17 + 19 2 - *

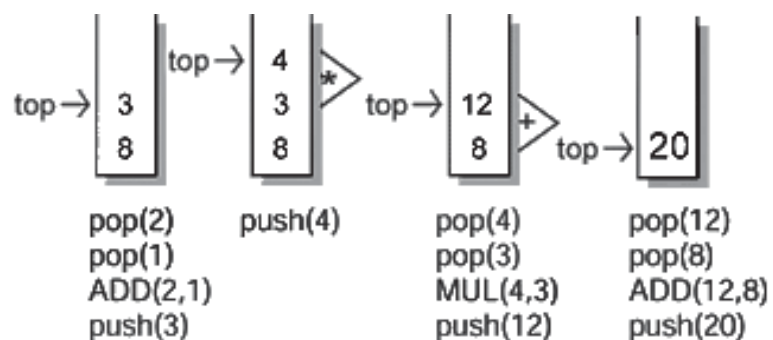
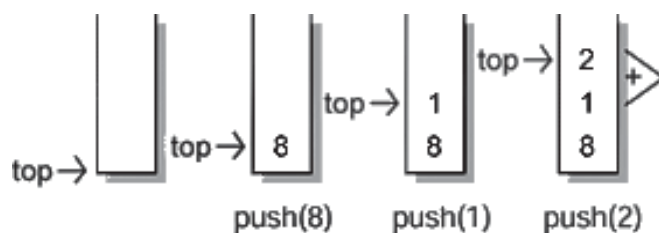
$12 / 4 * 3 - 5$

12 4 / 3 * 5 -

Algorithmus mit Stack (grob) - LIFO-Prinzip wird wieder deutlich!

„Zahlen kommen auf den Stack,
 Operatoren werden auf zwei letzte Zahlen angewendet“

Algorithmus am Beispiel



Quelltext mit Beispiel

```
public void btUPN_click() {
    //Berechnung eines (korrekt eingegebenen)
    //Integer-UPN-Terms
    //(mit Leerstellen, auch am Ende!) -
    //zum Schluss steht das Ergebnis auf dem Stack!
    String t = tfUPN.getText();
    //suche erste Leerstelle (Trenn-Markierung)
    int pos = t.indexOf(" ");
    //Auswertung aller Fälle -
    //bei Operator Rechnung, bei Zahl push
    switch (t.charAt(0)) {
        case '+', '-', '*', '/':
            int b = Integer.parseInt(s.top()); s.pop();
            int a = Integer.parseInt(s.top()); s.pop();
            switch (t.charAt(0)) {
                case '+': s.push(String.valueOf(a + b)); break;
                case '-': s.push(String.valueOf(a - b)); break;
                case '*': s.push(String.valueOf(a * b)); break;
                case '/': s.push(String.valueOf(a / b)); break;
            }
            break;
        default: s.push(t.substring(0, pos));
    }
    //Anpassen des Terms
    if (t.length() > 0) t = t.substring(pos+1);
    //Deaktivieren von Elementen
    //(wenn Term abgearbeitet)
    if (t.length() == 0) {
        btUPN.setDisable(true);
        tfUPN.setDisable(true);
        btKopie.setDisable(true);
    }
    gibAus();
    tfUPN.setText(t);
}
```

