

Ugeseddel 7

30. oktober - 6. november

- Læs kapitel 12 og 13

- Løs følgende opgaver

Opgave 23: 10.2 (1 point)

Opgave 24: 11.2 (1 point)

Opgave 25: 11.3 (2 point)

Opgave 26: Se de næste sider (3 point, ikke-obligatorisk)

Afleveringsfrist: **tirsdag den 13. november**

På de følgende sider er angivet en Java-klasse til udregning af aritmetiske udtryk. Klassen er tilgængelig fra kursets hjemmeside via henvisningen "Kode til opgaver".

Tilføj muligheden for at udregne aritmetiske udtryk, som indeholder potensopløftning. Som symbol for potensopløftning anvendes \wedge . Bemærk at \wedge er højre-associativ. Således udregnes 2^{2^3} som $2^{(2^3)}$.

```
import java.util.*;

public class Calculator {
    static final int PLUS = 0, MINUS = 1, MULT = 2,
                   DIV = 3, LPAR = 4, RPAR = 5,
                   CONST = 6, EOS = 7;

    int token;
    double value;
    StringTokenizer str;

    double valueOf(String s) {
        str = new StringTokenizer(s, "+-*/()= ", true);
        getToken();
        return expression();
    }

    double expression() {
        double v = term();
        while (token == PLUS || token == MINUS)
            if (token == PLUS)
                { getToken(); v += term(); }
            else
                { getToken(); v -= term(); }
        return v;
    }

    double term() {
        double v = factor();
        while (token == MULT || token == DIV)
            if (token == MULT)
                { getToken(); v *= factor(); }
            else
                { getToken(); v /= factor(); }
        return v;
    }

    double factor() {
        double v = 0;
        if (token == CONST)
            v = value;
        else if (token == LPAR) {
            getToken();
            value = expression();
            if (token != RPAR)
                error("missing right paranthesis");
        } else
            error("illegal factor");
        getToken();
        return v;
    }
}
```

```

void error(String msg) {
    throw new RuntimeException(msg);
}

void getToken() {
    String s;
    try {
        s = str.nextToken();
    } catch(NoSuchElementException e) {
        token = EOS;
        return;
    }
    if (s.equals(" ")) getToken();
    else if (s.equals("+")) token = PLUS;
    else if (s.equals("-")) token = MINUS;
    else if (s.equals("*")) token = MULT;
    else if (s.equals("/")) token = DIV;
    else if (s.equals("(")) token = LPAR;
    else if (s.equals(")")) token = RPAR;
    else {
        try {
            value = Double.valueOf(s).doubleValue();
            token = CONST;
        } catch(NumberFormatException e) {
            error("constant expected");
        }
    }
}

public static void main(String arg[]) {
    Calculator calc = new Calculator();
    System.out.println(calc.valueOf("3*2+4*5"));
}
}

```