

INF2100

Oppgaver 3.-15. oktober 2007

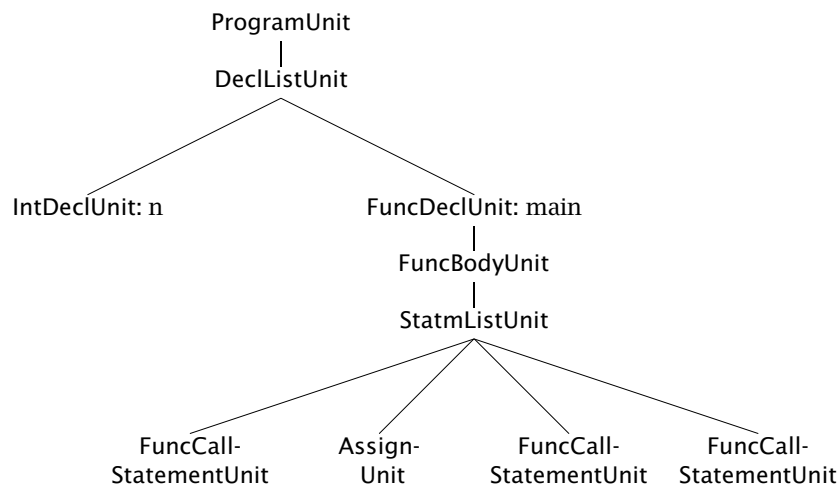
(Koden fra oppgave 2 og 3 finnes også i ~inf2100/oppgaver/uke-40/.)

Oppgave 1

På ark 13 og 14 (de to siste under «Kompilering») fra forelesningen 2. oktober står et kort RusC-program

```
int n;  
  
func main ()  
{  
    putchar('?'); n = getint()*2;  
    putint(n); putchar(10);  
}
```

og den øverste delen av syntakstreet til programmet; det ser slik ut:



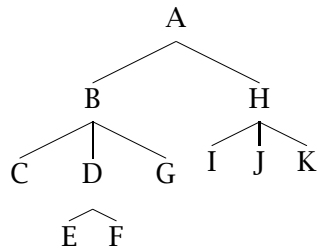
Tegn treet ferdig.

Hint Klassenavnene for alle metasymbolene står ikke nevnt, så du må selv finne på noen av dem.

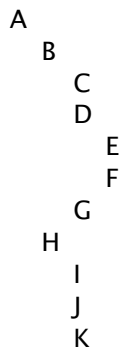
Oppgave 2

Her er et Java-program som bygger opp treet som er vist i figur 1 på neste side:

```
class Tree {  
    public static void main(String arg[]) {  
        Unit pa = new Unit("A"), pb = new Unit("B"),  
        pc = new Unit("C"), pd = new Unit("D"),  
        pe = new Unit("E"), pf = new Unit("F"),  
        pg = new Unit("G"), ph = new Unit("H"),  
        pi = new Unit("I"), pj = new Unit("J"),  
        pk = new Unit("K");
```



Figur 1: Et tre fra oppgave 2



Figur 2: Utskrift fra oppgave 2

```

    pa.first_child = pb;
    pb.first_child = pc; pb.next = ph;
    pc.next = pd;
    pd.first_child = pe; pd.next = pg;
    pe.next = pf;
    ph.first_child = pi;
    pi.next = pj;
    pj.next = pk;

    pa.print(0);
  }
}

class Unit {
  String name;
  Unit next = null, first_child = null;

  Unit(String n) {
    name = n;
  }

  void print (int level) {

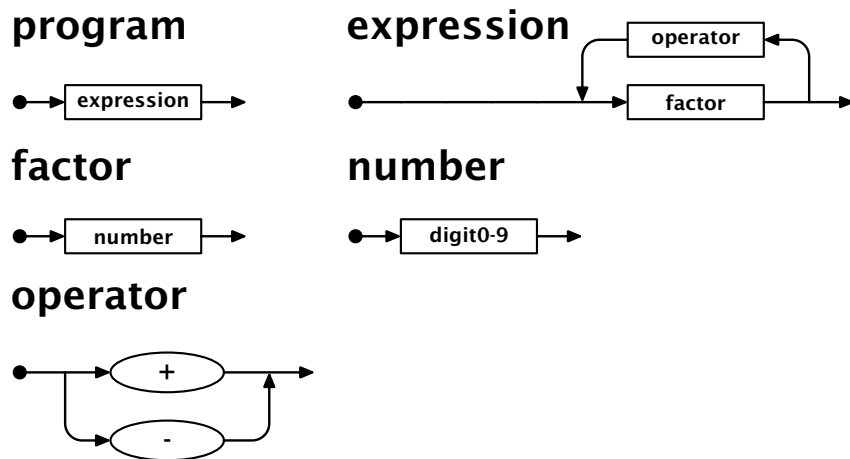
  }
}

```

Skriv metoden print slik at treet skrives ut som vist i figur 2. (Parameteren level angir hvor mye hver linje skal indenteres.)

Oppgave 3

Vi skal se på et enkelt språk E som består av meget enkle uttrykk med + og - på én-sifrete tall; grammatikken ser du i figur 3 på neste side. En implementasjon av



Figur 3: Grammatikken til språket E

dette språket ser slik ut:

```
import java.io.*;
```

```
class E {
    public static void main(String arg[]) {
        Scanner.init(); Scanner.readNext();

        Program p = Program.parse();
        if (Scanner.curToken != Token.eofToken)
            Error.error("Syntax error: Illegal "+Scanner.curToken);

        System.out.println("The value is "+p.eval());
    }
}
```

```
class Program {
    Expression e;

    public static Program parse() {
    }

    public int eval() {
        return e.eval();
    }
}
```

```
class Expression {
    ExprElem e;

    public static Expression parse() {
    }

    public int eval() {
        return e.eval();
    }
}
```

```
abstract class ExprElem {
    public abstract int eval();
}
```

```
class Factor extends ExprElem {
    Number n;
```

```

public static Factor parse() {
    Factor f = new Factor();
    f.n = Number.parse();
    return f;
}

public int eval() {
    return n.eval();
}
}

class Number {
    int val;

    public static Number parse() {
    }

    public int eval() {
        return val;
    }
}

class Operator extends ExprElem {
    Token op;
    ExprElem param1, param2;

    public static Operator parse() {
    }

    public int eval() {
        if (op == Token.plusToken)
            return param1.eval()+param2.eval();
        else
            return param1.eval()-param2.eval();
    }
}

enum Token { numberToken, plusToken, minusToken, eofToken, noToken }

class Scanner {
    public static Token curToken;
    public static int curNumber;

    private static LineNumberReader f;

    public static void init() {
        f = new LineNumberReader(new InputStreamReader(System.in));
    }

    public static void readNext() {
        curToken = Token.noToken;
        while (curToken == Token.noToken) {
            int c = '?';
            try {
                c = f.read(); // Read one character
            } catch (IOException e) {
                Error.error("Read error!");
            }
        }

        if (c < 0) {

```

```

        curToken = Token.eofToken;
    } else if (c == '+') {
        curToken = Token.plusToken;
    } else if (c == '-') {
        curToken = Token.minusToken;
    } else if (Character.isDigit(c)) {
        curToken = Token.numberToken; curNumber = c-'0';
    } else if (Character.isWhitespace(c)) {
        // Ignore space
    } else {
        Error.error("Illegal character: '"+(char)c+"'!");
    }
}
// For testing:
// System.out.println("Scanner: Read a "+curToken);
}
}

class Error {
    static void error(String message) {
        System.out.println("ERROR: "+message);
        System.exit(1);
    }
}
}

```

Fyll ut de tomme utgavene av parse.

Eksempel

Om vi sender inn programmet

1 + 2 - 1 + 9 + 2

skal svaret av evalueringen bli

The value is 13