

INF2100

Oppgaver 19.-24. september 2007

1 Språket

Denne uken skal vi implementere et meget enkelt språk. Syntaksen er vist i figur 1 og betydning er meget enkel:

- Interpreten vår skal lese heltall fra tastaturet (dvs *standard inn*); etter hvert heltall skal det stå et semikolon.
- Tallene skal skrives ut på skjermen (dvs *standard ut*), ett på hver linje.

(Dette språket skal utvides neste uke.)

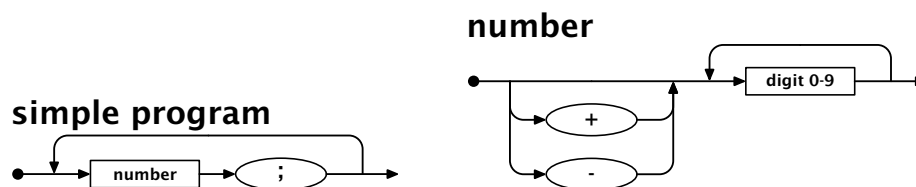
1.1 Et eksempel

Anta at vi har en «kildefil» med navn `demo.simple` og som ser slik ut:

```
314;  
6  
;  
1; -345; +003;  
-;
```

Her er et eksempel på en kjøring med denne filen:

```
>java SimpleNumExpr <demo.simple  
314  
6  
1  
-345  
3  
Error in line 5: Illegal number: '-'
```



Figur 1: Programmeringsspråkets syntaks er definert av disse to jernbandediagrammene.

Og her er en testkjøring av skanneren:

```
>java SimpleNumExpr -testscanner <demo.simple
Scanner: numberToken 314
Scanner: semicolonToken
Scanner: numberToken 6
Scanner: semicolonToken
Scanner: numberToken 1
Scanner: semicolonToken
Scanner: numberToken -345
Scanner: semicolonToken
Scanner: numberToken 3
Scanner: semicolonToken
Error in line 5: Illegal number: '-'
```

2 Koden

Her er en implementasjon av en interpret for språket; koden ligger også på filen `~inf2100/kode/numbers/SimpleNumExpr.java` og er tilgjengelig på nettet som <http://www.ifi.uio.no/~inf2100/kode/numbers/SimpleNumExpr.java>.

```
import java.io.*;

class SimpleNumExpr {
    public static void main(String arg[]) {
        Scanner.readNext();
        if (arg.length>0 && arg[0].equals("-testscanner")) {
            while (true) {
                System.out.print("Scanner: "+Scanner.curToken);
                if (Scanner.curToken == Token.numberToken)
                    System.out.print(" "+Scanner.curNumber);
                System.out.println();

                if (Scanner.curToken == Token.eofToken)
                    break;
                Scanner.readNext();
            }
        } else {
            while (true) {
                if (Scanner.curToken == Token.eofToken)
                    break;

                Scanner.check(Token.numberToken);
                System.out.println(Scanner.curNumber);
                Scanner.readNext();
                Scanner.check(Token.semicolonToken);
                Scanner.readNext();
            }
        }
    }
}
```

```

    }
}

enum Token {
    eofToken, noToken, numberToken, semicolonToken;
}

class Scanner {
    public static Token curToken = Token.noToken;
    public static long curNumber = 0;
    public static int curLineNum = 0;

    private static LineNumberReader stdIn =
        new LineNumberReader(new InputStreamReader(System.in));
    private static String curLine = "";
    private static int curPos = 0;

    public static void readNext() {
        curToken = Token.noToken;
        while (curToken == Token.noToken) {
            // ** Her mangler kode! **
        }
    }

    private static long readNumber() {
        String n = ""+curLine.charAt(curPos-1);

        while (Character.isDigit(curLine.charAt(curPos)))
            n += curLine.charAt(curPos++);

        try {
            return Long.parseLong(n.startsWith("+") ? n.substring(1) : n);
        } catch (NumberFormatException e) {
            Error.error("line "+Scanner.curLineNum,
                "Illegal number: '"+n+"'!");
        }
        return 0; // Will never be run!
    }

    static void check(Token t1) {
        if (curToken != t1)
            Error.error("line "+curLineNum,
                "Expected a "+t1+" but found a "+curToken);
    }
}

class Error {
    static public void error(String where, String message) {
        System.out.println("Error" +

```

```
                (where.equals("") ? "" : " in "+where) +  
                ": " + message);  
        System.exit(1);  
    }  
  
    static public void error(String message) {  
        error("", message);  
    }  
}
```

3 Oppgaven

Oppgaven er å skrive koden som mangler i Scanner-klassen slik at den oppfører seg som vist i eksemplene.