IN2030

Løsningsforslag

Uke 38 2020

Oppgave 1

Uten testutskrifter — det fortsettes i ukeoppgavene neste uke. Merk at det her også antas at uttrykket er syntaktisk korrekt.

Expression

```
class Expression extends ESyntax {
    Term t;

    static Expression parse(Scanner s) {
        Expression e = new Expression();

        e.t = Term.parse(s);

        return e;
    }
}
```

Term

```
import java.util.ArrayList;

class Term extends ESyntax {
    ArrayList<Factor> operands = new ArrayList<>();
    ArrayList<Token> oprs = new ArrayList<>();

    static Term parse(Scanner s) {
        Term t = new Term();
        t.operands.add(Factor.parse(s));
        while (s.curToken().kind == TokenKind.plusToken ||
               s.curToken().kind == TokenKind.minusToken) {
            t.oprs.add(s.curToken());
            s.readNextToken();
            t.operands.add(Factor.parse(s));
        }
        return t;
    }
}
```

Factor

```
import java.util.ArrayList;

class Factor extends ESyntax {
    ArrayList<Atom> operands = new ArrayList<>();
    ArrayList<Token> oprs = new ArrayList<>();

    static Factor parse(Scanner s) {
        Factor f = new Factor();
        f.operands.add(Atom.parse(s));
        while (s.curToken().kind == TokenKind.astToken ||
               s.curToken().kind == TokenKind.slashToken) {
            f.oprs.add(s.curToken());
            s.readNextToken();
```

```
            f.operands.add(Atom.parse(s));
        }
        return f;
    }
}
```

Atom

```
class Atom extends ESyntax {
    Number num;
    InnerExpr expr;

    static Atom parse(Scanner s) {
        Atom a = new Atom();

        if (s.curToken().kind == TokenKind.leftParToken)
            a.expr = InnerExpr.parse(s);
        else
            a.num = Number.parse(s);

        return a;
    }
}
```

InnerExpr

```
class InnerExpr extends ESyntax {
    Expression expr;

    static InnerExpr parse(Scanner s) {
        InnerExpr ie = new InnerExpr();
        s.readNextToken();  // Skip past '('.
        ie.expr = Expression.parse(s);
        s.readNextToken();  // Skip past ')'.
        return ie;
    }
}
```

Number

```
class Number extends ESyntax {
    int val;

    static Number parse(Scanner s) {
        Number n = new Number();
        n.val = s.curToken().numVal;
        s.readNextToken();
        return n;
    }
}
```