```
class Expr
    begin
        instance
            int kind, priority;
            □char operator;
            Expr left, right;
            Expr init(
                int kind, priority;
                □char operator;
                Expr left, right;
                )
                begin
                    this.kind = kind;
                    this.priority = priority;
                    this.operator = operator;
                    this.left = left;
                    this.right = right;
                    return this;
                end
    end

int IDENT = 1, BIN = 2;

void parenth(
    int level, priority; Expr expr; var □char result; )
    begin
        □char result1;
        toString( level + 1, expr, result1 );
        if expr.priority < priority then
            result = "(" + result1 + ")";
        else
            result = result1;
        end
    end

void toString( int level; Expr expr; var □char result; )
    begin
        □char left, right;
        if expr.kind == IDENT then
            result = expr.operator;
        elif expr.kind == BIN then
            parenth( level + 1, expr.priority, expr.left, left );
            parenth( level + 1, expr.priority+1, expr.right, right );
            result = left + expr.operator + right;
        end
    end

Expr expr1 = new Expr.init( IDENT, 3, "a", null, null );
Expr expr2 = new Expr.init( IDENT, 3, "b", null, null );
Expr expr3 = new Expr.init( IDENT, 3, "c", null, null );
Expr expr4 = new Expr.init( BIN, 1, "+", expr1, expr2 );
Expr expr = new Expr.init( BIN, 2, "*", expr4, expr3 );
□char result = null;
toString( 1, expr, result );
println( result );
```