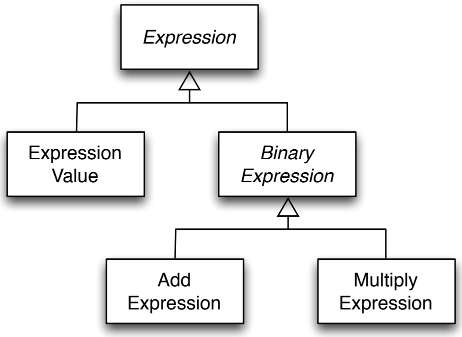
Consider the following Java class hierarchy for evaluating equations:



```
abstract public class Expression {
    abstract public Double evaluate();
}

public class ExpressionValue extends Expression {
    private double value;

    public ExpressionValue(double value) {
        this.value = value;
    }

    public void setValue(double value) {
        this.value = value;
    }

    @Override
    public Double evaluate() {
        return value;
    }
}

abstract public class BinaryExpression extends Expression {
    protected Expression left;
    protected Expression right;

    public BinaryExpression(Expression left, Expression right) {
        this.left = left;
        this.right = right;
    }
}

public class AddExpression extends BinaryExpression {
    public AddExpression(Expression left, Expression right) {
        super(left, right);
    }

    @Override
    public Double evaluate() {
        return left.evaluate() + right.evaluate();
    }
}
```

```
// which operations are not allowed?
ExpressionValue val1 = new ExpressionValue(4.2);
assertEquals(4.2, val1.evaluate(), .0001);
Expression val2 = new ExpressionValue(9.7);
assertEquals(9.7, val2.evaluate(), .0001);
BinaryExpression addition1 = new AddExpression(val1, val2);
assertEquals(4.2 + 9.7, addition1.evaluate(), .0001);
Expression addition2 = addition1;
val2.setValue(10.0);
AddExpression addition3 = new BinaryExpression(addition2, val2);
assertEquals(4.2 + 10.0 + 10.0, addition3.evaluate(), .0001);
```