

Klassen



```
class Op {
    int val;
    Op left, right;

    Op (int num){
        val = num; left = null; right = null;
    }
    Op (Op l, Op r){
        left = l; right = r; val = 0;
    }
    int evaluate(){
        int res = val;
        if (left != null) res += left.Evaluate();
        if (right != null) res += right.Evaluate();
        return res;
    }
}
```

Was gibt folgender Code aus?

```
Op l = new Op(9);
Op r = new Op(12);
Op op = new Op(l,r);
op = new Op(op,op);
Out.println(op.evaluate());
```

(1) 9

(2) 12

(3) 21

(4) 30

(5) 33

(6) 42

Klassen

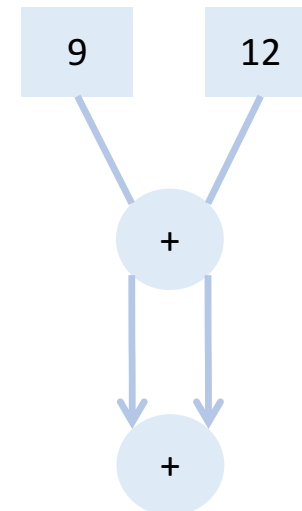


```
class Op {
    int val;
    Op left, right;

    Op (int num){
        val = num; left = null; right = null;
    }
    Op (Op l, Op r){
        left = l; right = r; val = 0;
    }
    int evaluate(){
        int res = val;
        if (left != null) res += left.Evaluate();
        if (right != null) res += right.Evaluate();
        return res;
    }
}
```

Was gibt folgender Code aus?

```
Op l = new Op(9);
Op r = new Op(12);
Op op = new Op(l,r);
op = new Op(op,op);
Out.println(op.evaluate());
```



Klassen



```
class Op {
    int val;
    Op left, right;

    Op (int num){
        val = num; left = null; right = null;
    }
    Op (Op l, Op r){
        left = l; right = r; val = 0;
    }
    int evaluate(){
        int res = val;
        if (left != null) res += left.Evaluate();
        if (right != null) res += right.Evaluate();
        return res;
    }
}
```

Was gibt folgender Code aus?

```
Op l = new Op(9);
Op r = new Op(12);
Op op = new Op(l,r);
op = new Op(op,op);
Out.println(op.evaluate());
```

(1) 9

(2) 12

(3) 21

(4) 30

(5) 33

(6) 42 ←