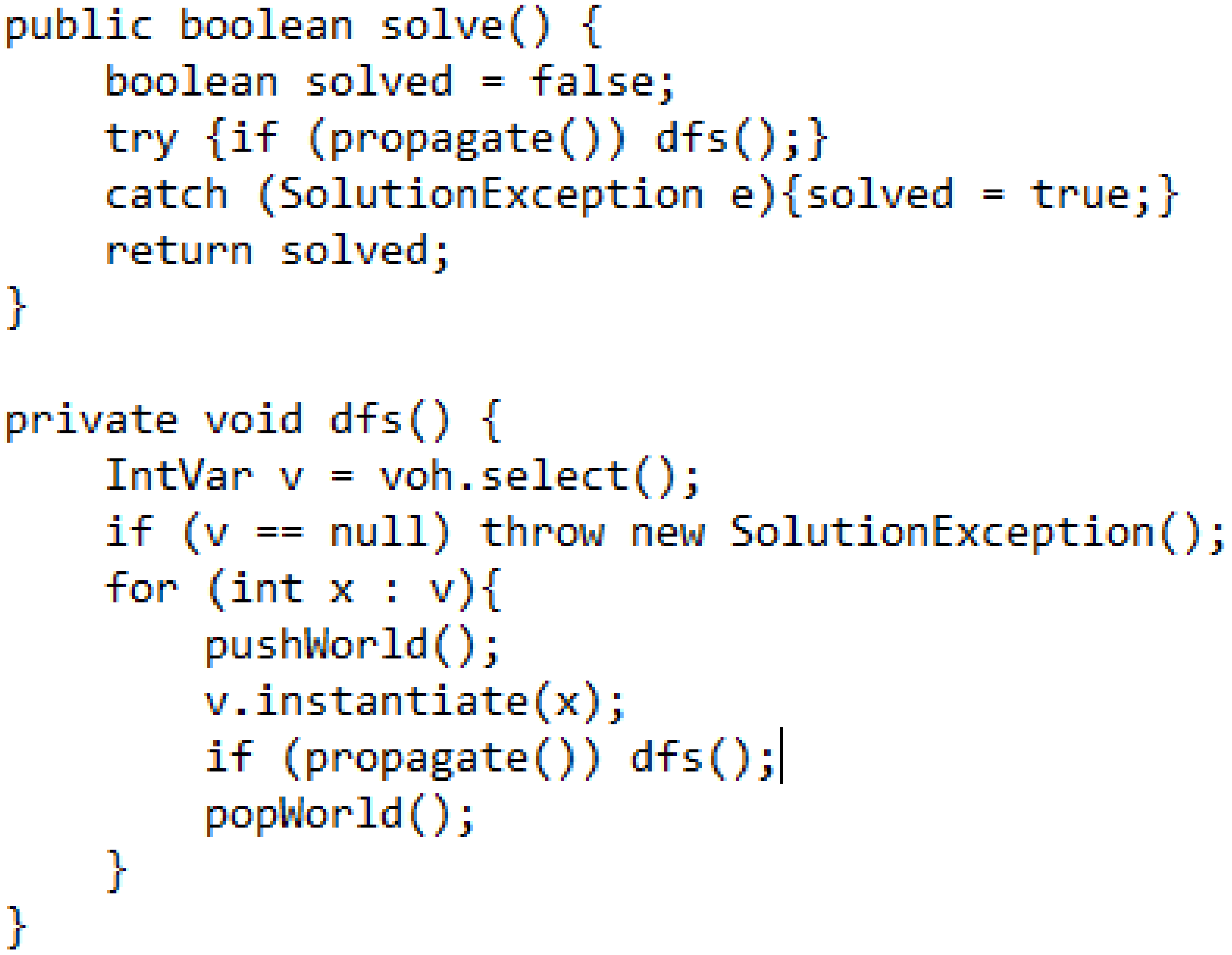
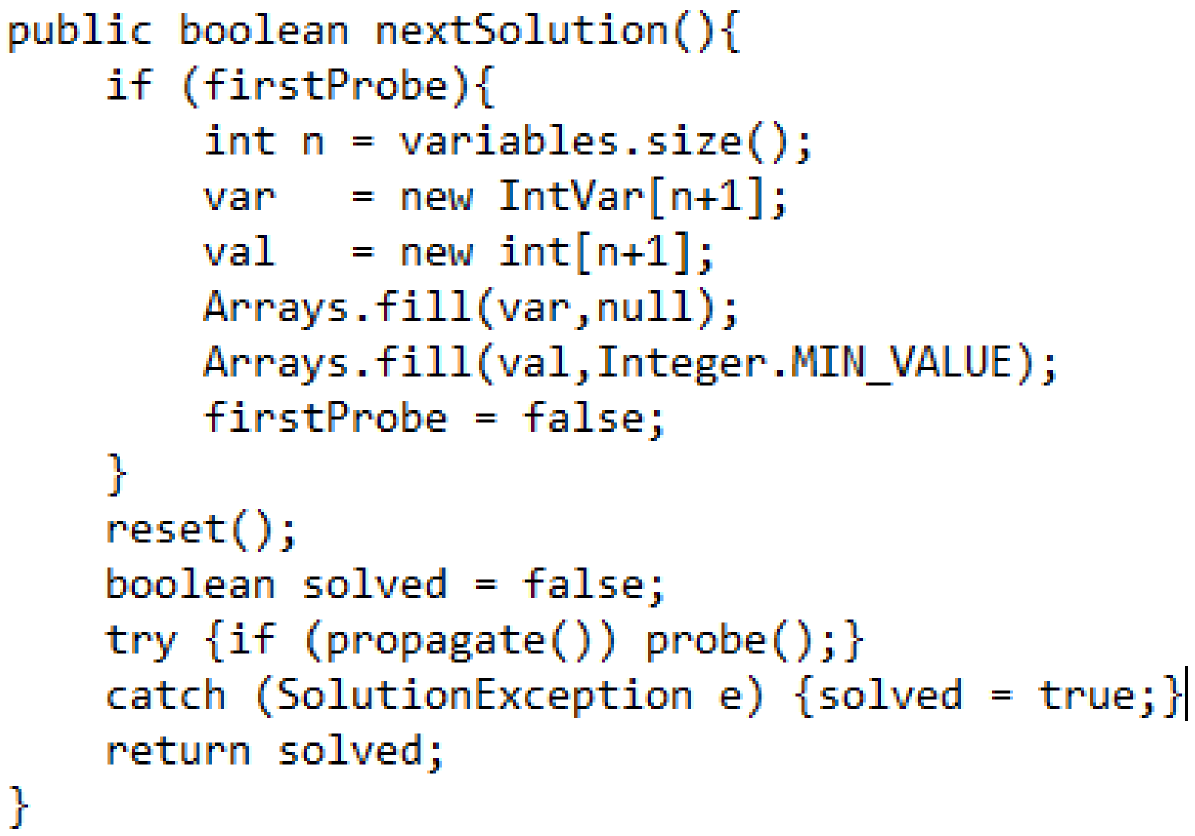
search

# Solve



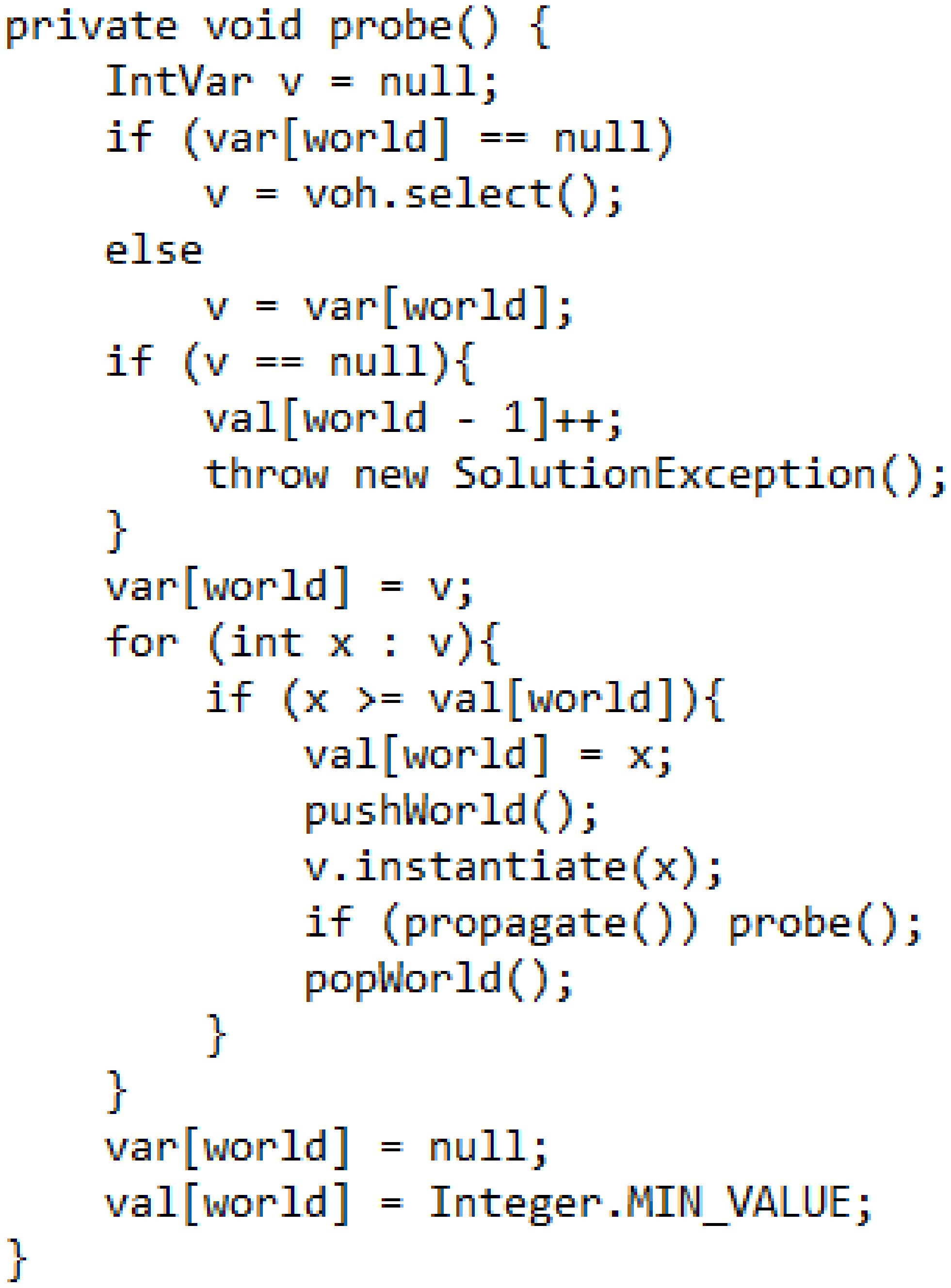
```
public boolean solve() {
    boolean solved = false;
    try {if (propagate()) dfs();}
    catch (SolutionException e){solved = true;}
    return solved;
}

private void dfs() {
    IntVar v = voh.select();
    if (v == null) throw new SolutionException();
    for (int x : v){
        pushWorld();
        v.instantiate(x);
        if (propagate()) dfs();
        popWorld();
    }
}
```

# nextSolution



```
public boolean nextSolution(){
    if (firstProbe){
        int n = variables.size();
        var   = new IntVar[n+1];
        val   = new int[n+1];
        Arrays.fill(var,null);
        Arrays.fill(val,Integer.MIN_VALUE);
        firstProbe = false;
    }
    reset();
    boolean solved = false;
    try {if (propagate()) probe();}
    catch (SolutionException e) {solved = true;}
    return solved;
}
```

nextSolution



```
private void probe() {
    IntVar v = null;
    if (var[world] == null)
        v = voh.select();
    else
        v = var[world];
    if (v == null){
        val[world - 1]++;
        throw new SolutionException();
    }
    var[world] = v;
    for (int x : v){
        if (x >= val[world]){
            val[world] = x;
            pushWorld();
            v.instantiate(x);
            if (propagate()) probe();
            popWorld();
        }
    }
    var[world] = null;
    val[world] = Integer.MIN_VALUE;
}
```