

Cassaforte.java

```
1/* TemaEsame_Schema_Inventato3: Esercizio 6 */
2
3public class Cassaforte {
4    /*@ spec_public @*/ private boolean sensori[];
5
6    //INVARIANTI
7    //@ public invariant sensori.length == 4;
8    //@ public invariant sensori != null;
9
10   //POSTCONDIZIONI
11   // 1) Tutti i sensori all'inizio sono accesi
12   //@ ensures (\forall int i; 0<=i && i<sensori.length; sensori[i] == true);
13   public Cassaforte() {
14       sensori = new boolean[] {true, true, true, true};
15   }
16
17   //PRECONDIZIONI
18   // 1) il valore del sensore deve essere compreso tra 0 e 3
19   //@ requires 0<=i && i<=3;
20   //POSTCONDIZIONI
21   // 1) se il metodo restituisce true i sensori adiacenti sono accesi
22   //@ ensures ((\result == true && i == 0) ==> (sensori[3] && sensori[1]));
23   //@ ensures ((\result == true && i == 3) ==> (sensori[0] && sensori[2]));
24   //@ ensures ((\result == true && !(i == 0 || i == 3)) ==> (sensori[i-1] && sensori[i+1]));
25   // 2) se il metodo restituisce false uno dei due sensori adiacenti è spento
26   //@ ensures ((\result == false && i == 0) ==> !(sensori[3] && sensori[1]));
27   //@ ensures ((\result == false && i == 3) ==> !(sensori[0] && sensori[2]));
28   //@ ensures ((\result == false && !(i == 0 || i == 3)) ==> !(sensori[i-1] &&
29   sensori[i+1]));
30   /*@ pure @*/
31   public boolean adiacenti(int i) {
32       if(i == 0)
33           return (sensori[3] && sensori[1]);
34       if(i == 3)
35           return (sensori[0] && sensori[2]);
36       else
37           return (sensori[(i-1)] && sensori[(i+1)]);
38   }
39
40   //PRECONDIZIONI
41   // 1) il valore del sensore deve essere compreso tra 0 e 3
42   //@ requires 0<=i && i<=3;
43   //POSTCONDIZIONI
44   // 1) se il sensore i è spento allora lo riaccendo
45   //@ ensures ((\old(sensori[i]) == false) ==> (sensori[i] == true));
46   // 2) se i sensori adiacenti ad i sono accesi allora spengo i
47   //@ ensures ((adiacenti(i) && \old(sensori[i])) ==> (sensori[i] == false));
48   public void switchSensore(int i) {
49       if(!sensori[i])
50           sensori[i] = true;
51       else {
52           if(adiacenti(i))
53               sensori[i] = false;
54       }
55   }
56   // public static void main(String[] args) {
```

Cassaforte.java

```

57 //      Cassaforte c = new Cassaforte();
58 //      c.switchSensore(0);
59 //      System.out.println(c.sensori[0]);
60 //      c.switchSensore(1);
61 //      System.out.println(c.sensori[1]);
62 //      c.switchSensore(2);
63 //      System.out.println(c.sensori[2]);
64 //      c.switchSensore(0);
65 //      System.out.println(c.sensori[0]);
66 //      c.switchSensore(2);
67 //      System.out.println(c.sensori[2]);
68 //
69 //  }
70 //
71
72 }
73

```

Proofs

Max. Rule Application

10000

Method Treatment

☐ Contract ☒ Expar

Dependency Contract

☒ On ☐ Off

Query Treatment

☒ On ☐ Off

Arithmetic Treatment

☐ Base ☒ DefOps

Stop at

☐ Default ☒ Unclos

Type	Target	Contract	Proof Reuse	Proof Result	N...	Br...	Ti...	G	G
Cassaforte	Cassaforte()	JML operation contract 0	New Proof	Closed	453	13	18...		
Cassaforte	switchSensore(...	JML operation contract 0	New Proof	Closed	17...	80	27...		
Cassaforte	adiacenti(int)	JML operation contract 0	New Proof	Closed	582	25	21...		