```java
/* Tema Esame: Schema - Esercizio 6 */

// ATTENZIONE: NON usare gli ENUMERATIVI con JML KeY perché
// KeY non riesce a dimostrare nulla se si utilizzano

public class Semaforo {

    // ColoreSemaforo è un array di 2 posizioni che contiene il colore
    // del semaforo relativo alla sua posizione (0=SEM1, 1=SEM2)
    /*@ spec_public @*/ int[] semaforo; //0=rosso, 1=verde, 2=giallo

    // INVARIANTI:
    // 1) mi assicuro che i semafori siano sempre 2
    //@ public invariant this.semaforo.length==2;
    // 2) mi assicuro che semaforo possa assumere solo i valori definiti nell'invariante ColoreSemaforo
    //@ public invariant (\forall int i; 0<=i && i<2; semaforo[i]==0 || semaforo[i]==1 || semaforo[i]==2);

    // POSTCONDIZIONI: mi assicuro che tutti i semafori all'inizio siano ROSSI
    //@ ensures ( this.semaforo[0]==0 && this.semaforo[1]==0);
    public Semaforo() {
        this.semaforo = new int[] {0, 0};

        // JML_1a: violo l'invariante 1)
        // this.semaforo = new int[] {0, 0, 0};

        // JML_1b: violo l'invariante 2)
        // this.semaforo = new int[] {25, 0};

        // JML_2: violo prima postcondizione
        // this.semaforo = new int[] {1, 0};

    }

    // PRECONDIZIONE: numSemaforo può essere solo 0 o 1
    //@ requires ( numSemaforo==0 || numSemaforo==1);

    // POSTCONDIZIONI:
    // 1)  Se sono entrambi rossi, numSemaforo è diventato verde
    //@ ensures ( ( \old(semaforo[0]==0) && \old(semaforo[1]==0) ) ==> semaforo[numSemaforo]==1 );

    // 2a) Se numSemaforo era verde, allora numSemaforo è diventato giallo
    //@ ensures ( \old(semaforo[numSemaforo]==1) ==> semaforo[numSemaforo]==2);

    // 2b)  Se numSemaforo era giallo, allora numSemaforo è diventato rosso
    //@ ensures ( \old(semaforo[numSemaforo]==2) ==> semaforo[numSemaforo]==0);

    // 3) Quando numSemaforo è rosso e l'altro semaforo non è rosso, se chiedo di cambiare il
    //    colore di numSemaforo allora il colore di numSemaforo NON cambia colore
    // NB: 1-numSemaforo = mi da il numero dell'altro semaforo
    //@ ensures ( ( \old(semaforo[numSemaforo]==0) && \old(semaforo[1-numSemaforo]!=0) )  ==> semaforo[numSemaforo]==0);
    public void cambiaColore(int numSemaforo) {
        // se sono entrambi rossi, metto a verde numSemaforo
        if(semaforo[0]==0 && semaforo[1]==0) {
```

```
54              semaforo[numSemaforo]=1;
55              // JML_4: violo postcondizione 1)
56              // semaforo[numSemaforo]=0;
57          }
58
59          // se chiedo di cambiare un semaforo che non è rosso, lo cambio
60          else if(semaforo[numSemaforo]!=0) {
61
62              // verde -> giallo
63              if(semaforo[numSemaforo]==1) {
64                  semaforo[numSemaforo]=2;
65                  // JML_5a: violo postcondizione 2a)
66                  // semaforo[numSemaforo]=0;
67
68              }
69
70              // giallo -> rosso
71              else if(semaforo[numSemaforo]==2) {
72                  semaforo[numSemaforo]=0;
73                  // JML_5b: violo postcondizione 2b)
74                  // semaforo[numSemaforo]=2;
75              }
76          }
77      }
78
79      public int[] getSemaforo() {
80          return semaforo;
81      }
82
83  //  public static void main(String[] args) {
84  //      Semaforo s = new Semaforo();
85
86          /*NOTA PER LE VIOLAZIONI DEI CONTRATTI*/
87          /*
88           * Utilizzo (1) "JML_numero" e (2) "JML_numero_check" per indicare
89           * con (1) qual è il contratto che viene violato con il codice scritto
90           * e con (2) qual è il codice nel main che mi permette di fare le
91           * chiamate che effettivamente violano quel contratto
92           *
93           * Per testare le violazioni bisogna prima commentare la parte di
94           * codice corretta e poi decommentare la parte di codice non corretta
95           * */
96
97          // JML_1a check:
98          // "20: JML invariant is false on leaving method Semaforo.Semaforo()"
99          // "20: //@ public invariant this.semaforo.length==2;"
100
101         // JML_1b check:
102         // "20: JML invariant is false on leaving method Semaforo.Semaforo()"
103         // "20: //@ public invariant (\forall int i; 0<=i && i<2; semaforo[i]==0 ||
    semaforo[i]==1 || semaforo[i]==2);"
104
105         // JML_2 check:
106         // "20: JML postcondition is false"
107         // "20: //@ ensures ( this.semaforo[0]==0 && this.semaforo[1]==0);"
108
109         // JML_3: violo range di valori di numSemaforo
```

```
110         // s.cambiaColore(3);
111         // JML_3 check:
112         // "110: JML precondition is false"
113         // "110: public void cambiaColore(int numSemaforo) {"
114
115         // JML_4: check
116         // s.cambiaColore(1);
117         // "51: JML postcondition is false"
118         // "51: //@ ensures ( ( \old(semaforo[0]==0) && \old(semaforo[1]==0) ) ==>
    semaforo[numSemaforo]==1 );"
119
120         // JML_5a: check
121         // s.cambiaColore(1);
122         // s.cambiaColore(1);
123         // "51: JML postcondition is false"
124         // "51: //@ ensures ( \old(semaforo[numSemaforo]==1) ==> semaforo[numSemaforo]==2);"
125
126         // JML_5b: check
127         // s.cambiaColore(1);
128         // s.cambiaColore(1);
129         // s.cambiaColore(1);
130         // "51: JML postcondition is false"
131         // "51: //@ ensures ( \old(semaforo[numSemaforo]==2) ==> semaforo[numSemaforo]==0);"
132
133         /* NOTA:
134             JML_6: la postcondizione 3) del metodo cambiaColore(..)
135             richiede di cambiare la logica del codice per essere violata
136             e quindi non provo a violarla (perché violerebbe anche le altre)
137         */
138
139         /*Debug code...*/
140 //      s.cambiaColore(1);
141 //      System.out.println("Colore sem2: "+s.getSemaforo()[1]);
142 //      s.cambiaColore(1);
143 //      System.out.println("Colore sem2: "+s.getSemaforo()[1]);
144 //      s.cambiaColore(1);
145 //      System.out.println("Colore sem2: "+s.getSemaforo()[1]);
146 //      s.cambiaColore(0);
147 //      System.out.println("Colore sem1: "+s.getSemaforo()[0]);
148 //      s.cambiaColore(0);
149 //      System.out.println("Colore sem1: "+s.getSemaforo()[0]);
150 //      s.cambiaColore(0);
151 //      System.out.println("Colore sem1: "+s.getSemaforo()[0]);
152
153 //  }
154 }
155
```