

```
1 package CapaDomini.PatroAdaptador;
2
3 public interface IZoombi {
4
5     public boolean moure(int posicioEnemic);
6
7     public int getDamage();
8
9     public int getPosicio();
10
11     public boolean isVisible();
12
13     public String toString();
14 }
15
```

```
1  package CapaDomini.PatroAdaptador;
2
3  import CapaDomini.ZoombiAmagat;
4
5
6  public class AdaptadorZoombiAmagat implements IZoombi {
7
8      private ZoombiAmagat zombi;
9
10     public AdaptadorZoombiAmagat(int[] args) {
11         if (args.length != 1)
12             throw new IllegalArgumentException("Paràmetre d'entrada no vàlid");
13
14         zombi = new ZoombiAmagat(args[0]);
15     }
16
17     @Override
18     public boolean moure(int posicioEnemic) {
19         return zombi.getPosicio() == posicioEnemic;
20     }
21
22     @Override
23     public int getDamage() {
24         return 0;
25     }
26
27     @Override
28     public int getPosicio(){
29         return zombi.getPosicio();
30     }
31
32     @Override
33     public boolean isVisible() {
34         return zombi.isVisible();
35     }
36
37     public String toString() {
38         return "Amagat a " + zombi.getPosicio();
39     }
40 }
41
```

```
1  package CapaDomini.PatroAdaptador;
2
3  import CapaDomini.ZoombiAtacant;
4
5
6  public class AdaptadorZoombiAtacant implements IZoombi {
7
8      private ZoombiAtacant zombi;
9
10     public AdaptadorZoombiAtacant(int[] args) {
11         if (args.length != 2)
12             throw new IllegalArgumentException("Paràmetre d'entrada no vàlid");
13
14         zombi = new ZoombiAtacant(args[0], args[1]);
15     }
16
17     @Override
18     public boolean moure(int posicioEnemic) {
19         return zombi.getPosicio() == posicioEnemic;
20     }
21
22     @Override
23     public int getDamage() {
24         return zombi.getDamage();
25     }
26
27     @Override
28     public int getPosicio() {
29         return zombi.getPosicio();
30     }
31
32     @Override
33     public boolean isVisible() {
34         return true;
35     }
36
37     public String toString() {
38         return "Atacant a " + zombi.getPosicio() + " fa dany " + zombi.getDamage();
39     }
40 }
41
```

```
1  package CapaDomini.PatroAdaptador;
2
3  import java.lang.reflect.Field;
4
5  import CapaDomini.ZoombiCaminant;
6
7  public class AdaptadorZoombiCaminant implements IZoombi {
8
9      private ZoombiCaminant zombi;
10
11      public AdaptadorZoombiCaminant(int[] args) {
12          if (args.length != 2)
13              throw new IllegalArgumentException("Paràmetre d'entrada no vàlid");
14
15          zombi = new ZoombiCaminant(args[0], args[1]);
16      }
17
18      @Override
19      public boolean moure(int posicioEnemic) {
20          return zombi.moure(posicioEnemic);
21      }
22
23      @Override
24      public int getDamage() {
25          return 0;
26      }
27
28      @Override
29      public int getPosicio() {
30          try {
31              Field privateField = ZoombiCaminant.class.getDeclaredField("posicio");
32              privateField.setAccessible(true);
33              return (int) privateField.get(zombi);
34          } catch (Exception e) {
35              e.printStackTrace();
36              throw new RuntimeException("Error posició del zombi:
37              AdaptadorZoombiCaminant");
38          }
39
40      @Override
41      public boolean isVisible() {
42          return true;
43      }
44
45      public String toString() {
46          return "Caminant a " + getPosicio();
47      }
48
49  }
50
```

```
16     public synchronized static FactoriaSingleton getInstance()  
17     {
```

```

1  package CapaDomini.PatroAdaptador;
2
3  import static org.junit.jupiter.api.Assertions.*;
4
5  import org.junit.jupiter.api.*;
6
7  import CapaAplicació.ETipusZoombi;
8  import CapaAplicació.FactoriaSingleton;
9  import CapaDomini.Personatge;
10
11  class TestPatroAdaptador {
12
13      private Personatge personatge;
14      private static FactoriaSingleton factoriaUnica;
15
16      @BeforeAll
17      static void setUpBeforeClass() throws Exception {
18          factoriaUnica = FactoriaSingleton.getInstance();
19      }
20
21      @BeforeEach
22      void setUp() throws Exception {
23          personatge = new Personatge( 5 );
24      }
25
26
27      @Test
28      void test_ZoombiAmagat() {
29          AdaptadorZoombiAmagat zombi = new AdaptadorZoombiAmagat( new int[] {
30              personatge.getPosicio() });
31          this.zoombiAmagat(zombi);
32      }
33
34      @Test
35      void test_AdaptadorZoombiAmagat() {
36          IZoombi zombi = factoriaUnica.getZombi(ETipusZoombi.ZoombiAmagat, new int[] {
37              personatge.getPosicio() });
38          assertEquals( AdaptadorZoombiAmagat.class , zombi.getClass());
39          this.zoombiAmagat((AdaptadorZoombiAmagat) zombi);
40      }
41
42      private void zoombiAmagat(AdaptadorZoombiAmagat zombi) {
43          assertFalse( zombi.moure( - personatge.getPosicio()));
44          assertTrue( zombi.moure(personatge.getPosicio()));
45          assertEquals(personatge.getPosicio(), zombi.getPosicio());
46          assertEquals(0, zombi.getDamage());
47          assertFalse( zombi.isVisible() );
48      }
49
50      @Test
51      void test_ZoombiCaminant(){
52          int posicio = 0;
53          int velocitat = 2;
54          AdaptadorZoombiCaminant zombi = new AdaptadorZoombiCaminant( new int[] {
55              posicio, velocitat });
56          this.zoombiCaminant(posicio, velocitat, zombi);
57      }
58
59      @Test
60      void test_AdaptadorZoombiCaminant() throws Exception {
61          int posicio = 0;
62          int velocitat = 2;
63          IZoombi zombi = factoriaUnica.getZombi(ETipusZoombi.ZoombiCaminant, new int[] {
64              posicio, velocitat });
65
66          assertEquals( AdaptadorZoombiCaminant.class , zombi.getClass());
67          this.zoombiCaminant(posicio, velocitat, (AdaptadorZoombiCaminant) zombi);
68      }
69
70      private void zoombiCaminant(int posicio, int velocitat, AdaptadorZoombiCaminant
71          zombi) {
72          assertEquals(posicio, zombi.getPosicio());

```

```

69         assertEquals(0, zombi.getDamage());
70         assertTrue( zombi.isVisible() );
71         // moure
72         posicio += velocitat;
73         assertFalse( zombi.moure(personatge.getPosicio()));
74         assertEquals(posicio, zombi.getPosicio());
75         // moure
76         posicio += velocitat;
77         assertFalse( zombi.moure(personatge.getPosicio()));
78         assertEquals(posicio, zombi.getPosicio());
79         // moure i atrapa
80         assertTrue( zombi.moure(personatge.getPosicio()));
81         assertEquals(personatge.getPosicio(), zombi.getPosicio());
82     }
83
84     @Test
85     void test_ZoombiAtacant(){
86         int posicio = personatge.getPosicio();
87         int damage = 2;
88         AdaptadorZoombiAtacant zombi = new AdaptadorZoombiAtacant( new int[] { posicio
89         , damage });
90         this.zoombiAtacant(posicio, damage, zombi);
91     }
92
93     @Test
94     void test_AdaptadorZoombiAtacant() {
95         int posicio = 5;
96         int damage = 2;
97         IZoombi zombi = factoriaUnica.getZoombi(ETipusZoombi.ZoombiAtacant, new int[]
98         { posicio, damage });
99
100         assertEquals( AdaptadorZoombiAtacant.class , zombi.getClass());
101         this.zoombiAtacant(posicio, damage, (AdaptadorZoombiAtacant) zombi);
102     }
103
104     private void zoombiAtacant( int posicio, int damage, AdaptadorZoombiAtacant zombi)
105     {
106         assertEquals(posicio, zombi.getPosicio());
107         assertEquals(damage, zombi.getDamage());
108         assertTrue( zombi.isVisible() );
109         assertFalse( zombi.moure( - personatge.getPosicio()));
110         assertTrue( zombi.moure(personatge.getPosicio()));
111     }

```

```

1  package CapaPresentacio;
2
3  import java.util.ArrayList;
4  import java.util.Scanner;
5
6  import CapaAplicació.ETipusZoombi;
7  import CapaAplicació.FactoriaSingleton;
8  import CapaDomini.Personatge;
9  import CapaDomini.PatroAdaptador.*;
10
11 public class JocZoombis {
12
13     private static Scanner myScanner = new Scanner(System.in);
14
15     public static void main(String[] args) {
16         System.out.println("Juguem!");
17
18         FactoriaSingleton factoriaUnica = FactoriaSingleton.getInstance();
19
20         Personatge personatge = new Personatge();
21
22         ArrayList<IZoombi> llistatZombis = new ArrayList<IZoombi>();
23         for (ETipusZoombi elTipus : ETipusZoombi.values()) {
24             llistatZombis.add(factoriaUnica.getZoombi(elTipus));
25         }
26
27         while (personatge.getVida() > 0) {
28             moure(personatge);
29
30             for (IZoombi zombi : llistatZombis) {
31
32                 if (personatge.getPosicio() == zombi.getPosicio()) {
33                     System.out.println("Has trobat un zoombi del tipus " + zombi);
34                     personatge.setVida(zombi.getDamage());
35                 } else if (zombi.moure(personatge.getPosicio())) {
36                     System.out.println("Un caminant s'ha mogut i t'ha atrapat");
37                     personatge.setVida(zombi.getDamage());
38                 }
39             }
40         }
41         System.out.println("Has mort");
42     }
43
44     private static void moure(Personatge personatge) {
45         System.out.println("Vols moure?\n 1.- moviment a l'esquerra \n 2.- moviment a\n la dreta \n 3.- quiet");
46         String input = myScanner.nextLine();
47         switch (input) {
48             case "3": break;
49             case "1":
50                 personatge.setPosicio(personatge.getPosicio() - 1);
51                 break;
52             case "2":
53                 personatge.setPosicio(personatge.getPosicio() + 1);
54                 break;
55             default:
56                 moure(personatge);
57         }
58     }
59
60 }
61
62 }
63

```