

Listagem A

```
import java.util.ArrayList;
import java.util.List;

class Inteiro
{
    private int i;

    public Inteiro(int i){ this.i = i; }
    public void setVal(int i){ this.i = i; }
    public int getVal() { return i; }
    public String toString() { return "" + i; }
}

class Main
{
    public static void main(String[] args)
    {
        List <Inteiro> listaInteiros = new ArrayList<>();
        Inteiro i = new Inteiro(10);
        Inteiro j = new Inteiro(10);

        System.out.println(i == j);
        System.out.println(i.equals(j));

        listaInteiros.add(i);
        listaInteiros.add(j);

        j = i;
        j.setVal(2);
        System.out.println(i);
        System.out.println(j);
        System.out.println(listaInteiros.get(0));
        System.out.println(listaInteiros.get(1));
    }
}
```

Listagem B

```
1 interface IA { String getA(); }
2 interface IB extends IA { int getB(); }
3 interface IC { int getC(); }
4
5 abstract class D implements IB, IC {
6
7     private String a;
8     private int b, c;
9
10    public D(String a, int b, int c){
11        this.a = a;
12        this.b = b;
13        this.c = c;
14    }
15
16    public String getA(){ return a; }
17    public int getB(){ return b; }
18    public int getC(){ return c; }
19
20    public void setA(String a){ this.a = a; }
21    public void setB(int b) { this.b = b; }
22    public void setC(int c) { this.c = c; }
23
24    public boolean equals(Object o){
25        if(o == null || !(o instanceof D)){
26            return false;
27        }
28
29        return a.equalsIgnoreCase(o.getA());
30    }
31 }
32
33 class E extends D{
34     public E(String a, int b, int c){
35         setA(a); setB(b); setC(c);
36     }
37 }
38
39 class F extends E{
40
41     private double d;
42
43     public F(String a, int b, int c, double d){
44         setA(a); setB(b); setC(c);
45         this.d = d;
46     }
47
48     public double getD(){ return d; }
49
50     public boolean equals(Object o){
51         if(o == null || !(o instanceof F)){
52             return false;
53         }
54
55         return a.equalsIgnoreCase(o.getA()) && b == o.getB() && c == o.getC();
56     }
57 }
58
59
60 class Main
61 {
62     public static void main(String args[])
63     {
64         IB a = new E("abc", 1, 1);
65         D b = new F("abc", 1, 1, 2.2);
66         IA c = new E("abc", 3, 3);
```

```

67         IB d = new IB();
68         IC e = new F("ddd", 4, 4, 4.4);
69
70         IA r1[] = new D[5];
71         IA r2[] = new IB[5];
72         IB r3[] = new IA[5];
73         IC r4[] = new IC[5];
74
75         System.out.println(a);
76         System.out.println(b);
77         System.out.println(a.equals(b));
78         System.out.println(b.equals(a));
79         System.out.println(c.equals(b));
80         System.out.println(c.getA());
81         System.out.println(e.getA());
82     }
83 }

```

Listagem C

```

abstract class Passaro {
    public static String PASSARO = "Passaro", PERIQUITO = "Periquito", CANARIO = "Canario";

    public static Passaro criaPassaro(String tipoPassaro) {
        //Devolve um nova instância de Periquito ou Canario, ou null nos restantes casos
        /*..a..*/
    }
}

class Periquito extends Passaro { /*..b..*/ }

class Canario extends Passaro { /*..c..*/ }

class MainPassaro {

    public static void adicionaPassaro(Passaro p, Map<Passaro, Integer> gaiola) {
        if(gaiola == null || p == null) return;

        /*..d..*/
    }

    public static void main(String args[]) {
        Map</*..f..*/ > gaiola = new HashMap<>();
        Passaro p;

        for(int i=0; i<10; i++){
            adicionaPassaro(Passaro.criaPassaro(Passaro.PERIQUITO), gaiola);
            adicionaPassaro(Passaro.criaPassaro(Passaro.CANARIO), gaiola);
        }

        System.out.println(gaiola);
    }
}

```

Listagem D

```
1  class InteiroSomaRaizesQ {
2
3      static private double somaRaizesQuadradas = 0;
4      private double n;
5
6      public InteiroSomaRaizesQ (String n) throws Exception {
7          try{
8
9              try{
10                 this.n = Double.parseDouble(n);
11             }catch(NumberFormatException e){
12                 return;
13             }
14
15             if(this.n < 0){
16                 throw new Exception();
17             }
18
19             somaRaizesQuadradas += Math.sqrt(this.n);
20
21         }finally{
22             System.out.print("f1 ");
23         }
24     }
25
26     public static void main(String[] args) {
27
28         System.out.print("m1 ");
29
30         try{
31             new InteiroSomaRaizesQ("4.0");
32             new InteiroSomaRaizesQ("A.0");
33             new InteiroSomaRaizesQ("9.0");
34             new InteiroSomaRaizesQ("-4.0");
35             new InteiroSomaRaizesQ("9");
36         }catch(Exception e){
37             System.out.print("e1 ");
38             return;
39         }finally{
40             System.out.print("f2: " + somaRaizesQuadradas);
41         }
42     }
43 }
```

Listagem E

```
public class Semaforo {
    static final int TEMPO_VERDE = 4, TEMPO_VERMELHO = 4, TEMPO_AMARELO = 1;
    private int tempo;
    private String estado;
    private String estadoAoLigar;

    public Semaforo(){
        tempo = 0;
        estado = "desligado";
        estadoAoLigar = "vermelho";
    }
    public void passaTempo(){
        if (estado.equalsIgnoreCase("desligado")){
            return;
        }
        tempo--;
        if(tempo > 0){
            return;
        }
        if(estado.equalsIgnoreCase("vermelho")){
            estado = "verde";
            tempo = TEMPO_VERDE;
        }else if(estado.equalsIgnoreCase("verde")){
            estado = "amarelo";
            tempo = TEMPO_AMARELO;
        }else if(estado.equalsIgnoreCase("amarelo")){
            estado = "vermelho";
            tempo = TEMPO_VERMELHO;
        }
    }
    public void desliga( String ligarEm){
        if(estado.equalsIgnoreCase("desligado")){
            return;
        }
        estadoAoLigar = ligarEm;
        estado = "desligado";
        tempo = 0;
    }
    public void liga(){
        if(!estado.equalsIgnoreCase("desligado")){
            return;
        }
        estado = estadoAoLigar;
        if(estado.equalsIgnoreCase("vermelho")){
            tempo = TEMPO_VERMELHO;
        }else if(estado.equalsIgnoreCase("verde")){
            tempo = TEMPO_VERDE;
        }else if(estado.equalsIgnoreCase("amarelo")){
            tempo = TEMPO_AMARELO;
        }
    }
    public String toString(){
        return estado + "    " + tempo;
    }
}
```