

POO - testul 1

1. (3p = 10x0.3p) Completăți în grila de mai jos, în ordine, cele 10 valori afișate pe rânduri separate în urma execuției următorului program JAVA:

```

public class A {
    private static String s = "POO";
    private int i;
    private String t;

    static {
        s = "TOAMNA";
    }

    public A(String y) {
        i--;
        s += y;
    }

    public A() {
        this("3");
        i = 55;
        t = new String("TESTUL1");
    }

    public void setI(int i) {
        i = 44;
    }

    public static double method(A a) {
        int v = 15;
        a.i = v;
        a = new A("5");
        a.i = 5*v;
        return a.i/2;
    }

    public static int count() {
        int x=0;
        et: for(int i=0; i<10; i++) {
            x++;
            for (int j=0; j<20; j++) {
                x++;
                for (int k=0; k<30; k++) {
                    if (k>=5)
                        continue et;
                    x++;
                }
            }
        }
        return x;
    }
}

```

```

public static void main(String[] args) {
    A x = new A("9");
    A y = new A("9");
    System.out.println(x==y);
    System.out.println(x.i + y.i);

    A z = new A("7");
    double d = method(z);
    System.out.println(d);
    System.out.println(z.i != 15);

    A r = new A();
    A w = new A();
    System.out.println(r.t == w.t);
    System.out.println(r.t == "TESTUL"+"1");

    r.setI(33);
    w.setI(22);
    System.out.println(r.i + w.i);

    System.out.println(x.t);
    System.out.println(A.s);
    System.out.println(A.count());
}
}

```

Grila de răspunsuri:

01	
02	
03	
04	
05	
06	
07	
08	
09	
10	

2. (6p = 20x0.3p) Completăți în grila de mai jos conținutul care lipsește din chenarele numerotate de la 01 la 20 din programul JAVA scris mai jos astfel încât să respectați paradigma de programare orientată pe obiecte, iar programul să afișeze în urma execuției sale următoarele rezultate:

```
Cluj-Napoca->Bucuresti 8h30min
Cluj-Napoca->Sibiu 4h0min
Alba Iulia->Timisoara 3h0min
null
null
S-au creat 4 trenuri
Din Cluj-Napoca pleaca 2 trenuri
```

```
public class Tren {
    |01| sursa;
    |01| destinatie;
    private float durata;
    private |02| NR_TRENURI;

    |03| (String sursa, String d, float o) {
        |04|=sursa;
        destinatie=d;
        durata=o;
        |05|;
    }

    |06| getNrTrenuri() {
        return NR_TRENURI;
    }

    |06| getNrTrenuriSursa(|07|) {
        int count=0;
        for (|08|:t)
            if (x!=null && |09|)
                count++;
        return count;
    }

    public String |10| {
        return sursa+"->"+destinatie+" "
            +oreSiMinute();
    }

    |11| {
        int ore = |12|;
        int minute = (int)((durata-ore)*60);
        return |13|;
    }
}
```

```
public class MainClass {
    public static void main(String[] args) {
        Tren[] t = |14|;
        t[0] = new Tren ("Cluj-Napoca", "Bucuresti", |15|);
        t[1] = new Tren ("Cluj-Napoca", "Sibiu", 4);
        t[2] = new Tren ("Alba Iulia", "Timisoara", 3);
        |16|("Constanta", "Bucuresti", 2);

        for(int i=0; i<|17|; i++)
            System.out.println(|18|);

        System.out.println("S-au creat" +
            |19|.getNrTrenuri() + " trenuri");

        |20| s = new |20|("Cluj-Napoca");
        System.out.println("Din "+s+" pleaca " +
            |19|.getNrTrenuriSursa(t, s)+" trenuri");
    }
}
```

Grila de răspunsuri:

01		11	
02		12	
03		13	
04		14	
05		15	
06		16	
07		17	
08		18	
09		19	
10		20	