

POO - testul 2

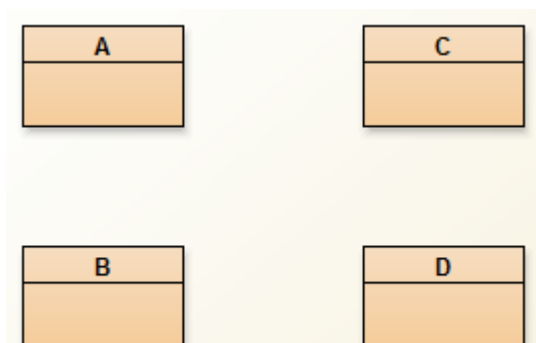
1. (4.5p = 10x0.3p + 1.5p) Completați în grila alăturată, în ordine, cele 10 valori afișate pe rânduri separate în urma execuției următorului program JAVA (3p). Desenați pe diagrama UML simplificată de mai jos toate relațiile care există între cele patru clase (1.5p).

```
public class A {
    protected String x = "Test";
    int y = 10;
    public A() {
        x+="Curs";
        y-=2;
    }
    public String getX() {
        return x;
    }
    public String info() {
        return "Clasa A";
    }
    public String toString() {
        return info();
    }
}
```

```
public class B extends A {
    String x = "Studenti";
    public B() {
        x+="POO";
        y*=3;
    }
    public String info() {
        return "Clasa B";
    }
}
```

```
public class C {
    private A x;
    public C (A x) {
        this.x=x;
    }
    public String toString() {
        return x.toString();
    }
}
```

Diagrama UML de clase (simplificată):



```
public class D {
    public static void main(String[] args) {
        B b = new B();
        A a = b;
        C c = new C(a);
        System.out.println(b instanceof A);
        System.out.println(c.getClass().getSuperclass().getName());
        System.out.println(a.x);
        System.out.println(b.x);
        System.out.println(a.y);
        System.out.println(b.y);
        System.out.println(a.getX());
        System.out.println(b.getX());
        System.out.println(a.info());
        System.out.println(c);
    }
}
```

Grila de răspunsuri:

01	
02	
03	
04	
05	
06	
07	
08	
09	
10	

2. (4.5p = 15x0.3p) Completați în grila de mai jos conținutul care lipsește din chenarele numerotate de la 01 la 15 din programul JAVA scris mai jos astfel încât să respectați paradigma de programare orientată pe obiecte, iar programul să afișeze în urma execuției următoarele rezultate:

```
OK: Nota a fost actualizata pentru studentul Popescu Alex!
OK: Nota a fost actualizata pentru studentul Ionescu Andreea!
EROARE: Nota trebuie sa fie intre 1 si 10 pentru studentul Popescu Alex!
PERSOANELE IN ORDINE ALFABETICA:
Bora Dan-1741215172475-cadru didactic cu biroul in sala 45
Ionescu Andreea-2960113322389-student cu nota 5.47
Moldovan Cristian-1860715141397-cadru didactic cu biroul in sala D2
Popescu Alex-1970318021187-student cu nota 8.54
```

```
public abstract class Persoana |01| {
    private String nume;
    private String CNP;
    public Persoana(String nume, String CNP) {
        this.nume=nume;
        this.CNP=CNP;
    }
    public String getNume() {
        return nume;
    }
    public String toString() {
        return nume+"-"+CNP+"-";
    }
    public int compareTo(Object obj) {
        return |02|;
    }
}
```

```
public class Student |03| {
    private float nota;
    public Student(String nume, String CNP) {
        |04|;
    }
    public String toString() {
        return |05| + "student cu nota "+nota;
    }
    public void primesteNota(float x) |06| {
        if (|07|)
            |08| Exception("EROARE: Nota trebuie
            sa fie intre 1 si 10 pentru
            studentul "+ |09| + "!");
        nota = x;
        System.out.println("OK: Nota a fost
        actualizata pentru studentul " +
        |09| + "!");
    }
}
```

```
public class CadruDidactic |03| {
    private String birou;
    public CadruDidactic(String nume,
        String CNP,
        String birou) {
        |04|;
        this.birou=birou;
    }
    public String toString() {
        return |05| + "cadru didactic cu
        biroul in "+birou;
    }
}
```

```
import java.util.*;
public class MainClass {
    public static void main (String [] args) {
        ArrayList<Persoana> persoane = |10|;
        persoane.add(new Student("Popescu Alex",
            "1970318021187"));
        persoane.add(new Student("Ionescu Andreea",
            "2960113322389"));
        persoane.add(new CadruDidactic("Moldovan Cristian",
            "1860715141397","sala D2"));
        persoane.add(new CadruDidactic("Bora Dan",
            "1741215172475","sala 45"));
        |11|{
            (|12|.get(0)).primesteNota(8.54f);
            (|12|.get(1)).primesteNota(5.47f);
            (|12|.get(0)).primesteNota(10.78f);
        }|13| {
            System.out.println(e.getMessage());
        }
        Collections.sort(persoane);
        System.out.println("PERSOANELE IN ORDINE
            ALFABETICA:");
        Iterator<Persoana> t = |14|;
        while (t.hasNext())
            System.out.println(|15|);
    }
}
```

Grila de răspunsuri:

01	
02	
03	
04	
05	
06	
07	
08	
09	
10	
11	
12	
13	
14	
15	