

# Grile POO → P2 (06.02.2025)

Considerând următorul cod Java, care din afirmațiile enumerate mai jos sunt ADEVĂRATE?

```
//A.java
package p1;
public class A {
    protected int x;
}

//B.java
package p1;
public class B extends A {}

//C.java
package p1;
public class C {}
class D {} → doar în interiorul aceluiași pachet

//E.java
package p1.p2;
import p1.A;
import p1.B;
public class E extends B {
```

**poate fi accesată de**  
**oarecare din pachet p1**  
**din subclase, pe**  
**this**

class D {}

```
//E.java
package p1.p2;
import p1.A;
import p1.B;
public class E extends B {
    E() {} → constructor nu-crește
    public void m(A a) {
        //1 a.x
    }
    public void n(C c) {} //2
}
```

Selectați unul sau mai multe:

- Clasa E poate fi instantiată în orice pachet X
- Linia marcată cu //1 conține o eroare de compilare
- La linia marcată cu //2 putem accesa câmpul x al obiectului referit de a
- Clasele A și B pot fi implementate în aceeași unitate de compilare, fără a fi necesare alte modificări a programului
- Putem declara referințe de tip D doar în cadrul pachetului p1

class B

A a = new A();

a.x asta nu merge

void printX () {

System.out.println ( this.x ); asta funcționează

}



import p1.\*;

Nu a primește răspuns încă  
Marcat din 1,00  
Pentru întrebarea corespunzătoare

Considerând următorul cod Java, care din fragmentele de cod enumurate mai jos vor conduce la o eroare de EXECUȚIE a programului când apar la poziția indicată cu /\*AICI\*/?

```
class A {
    public void m() {}
    public static void main(String args[]) {
        A[] a = new A[2];
        /*AICI*/
    }
}
```

Selectați unul sau mai multe:

- a[0] = new A(); ✓
- a[0].m();

a[0] = new A(); ✓

```

a[0].m();
a[2] = new A();
A[] x = new A[3];
x = a;
x[2] = new A();

```

a[1] = new A();

În metodele statice se pot apela doar metode statice

$a[0]$   $a[1]$   $\leftarrow A[] = \text{new } A[2]$

$a[0] = \text{new } A();$  ✓  
 $a[0].m();$  → nullpointer

dacă era initializat nu mai arăta eroare

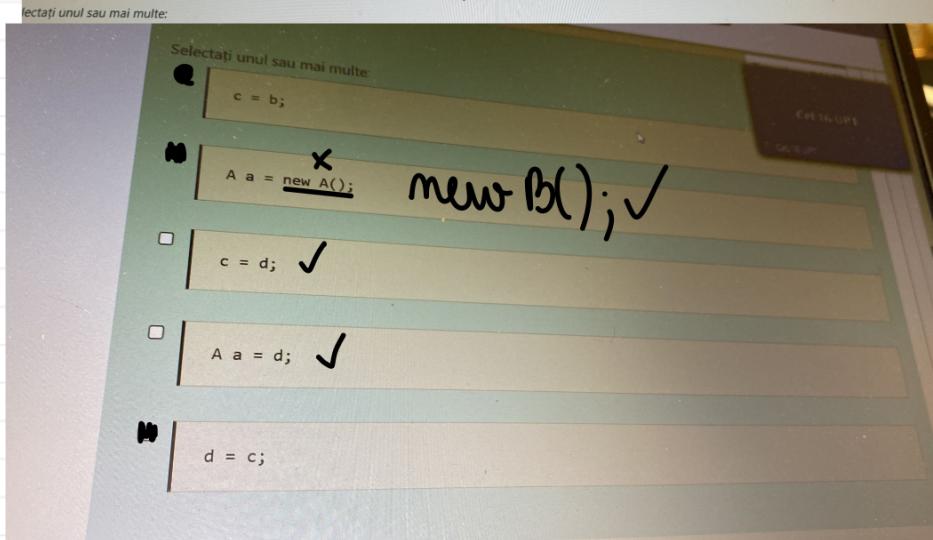
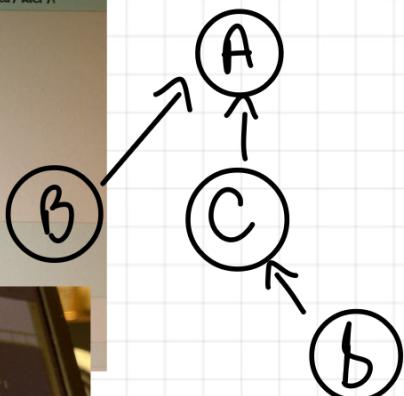
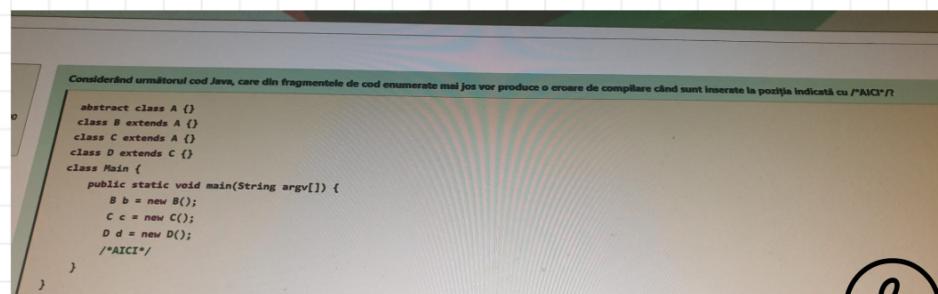
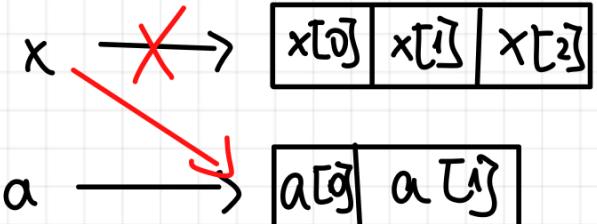
$a[2] = \text{new } A();$  → outofbounds

•  $A[] x = \text{new } A[3];$

$x = a;$

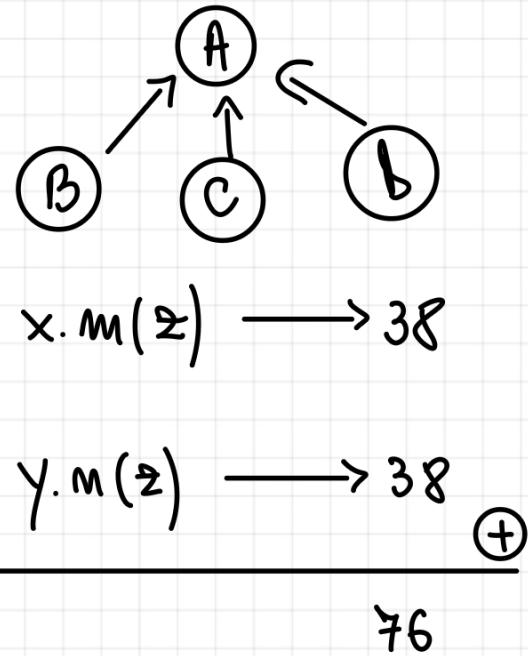
$x[2] = \text{new } A();$

→ outOfBounds



Ce valoare întreagă se va tipări la linia marcată cu /\* în urma executiei programului de mai jos?

```
class A {  
    public int m(A p) { return 12; }  
    public int n(A p) { return 38; }  
}  
class B extends A {  
    public int m(B p) { return 3; }  
    public int n(A p) { return 7; }  
}  
class C extends A {  
    public int m(A p) { return 38; }  
    public int n(A p) { return 4; }  
}  
class D extends A {  
    public int m(A p) { return 1; }  
    public int n(D p) { return 2; }  
}  
class Main {  
    public static void main(String argv[]) {  
        A x = new C();  
        A y = new D();  
        D z = new D();  
        System.out.println(x.m(z) + y.n(z));/*  
    }  
}
```



Întrebare  
a primit  
spuns încă  
arcat din 1,00  
Întrebare cu  
rg

Considerând următorul cod Java, care din fragmentele de cod enumerate mai  
jos să selectezi?

```
class A {  
    private int a = 0;  
    private static int b = 0;  
  
    public int m() {  
        return a;  
    }  
  
    public static int n(A x) {  
        /*ACII*/  
    }  
}
```

Selectați unul sau mai multe:

- $m \rightarrow$  statică /  $x.m()$  ar fi mers ✓
- $return b;$  ✓
- $return x.a;$
- $return this.a;$  nu știm la care obiect se referă  
→ nu e static

```

Considerând următorul cod Java, care din fragmentele de cod enumerate mai jos vor produce o eroare la poziția indicată cu /*AICI*/?

```

```

class G<T> {
    /* ... */
    void met(T t) /* ... */}
}

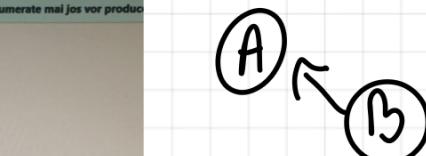
class A {/* ... */}
class B extends A /* ... */

public class Main {
    public static void test(G<? extends A> param) {
        G<A> x = new G<A>();
        G<B> y = new G<B>();
        G<? extends A> z = param;
        /*AICI*/
    }
}

```

Selectați unul sau mai multe:

- z.met(new A()); ✓
- z.met(new B()); ✓
- z = x; ✓
- x = y; ✗ nici x=y
- x.met(new B()); ✓
- z = y; ✓



G<? extends A>?

nu stie ce e?

z.met(null); → asta ar fi mers

G<B> nu e un subtip al lui G<A>  
chiar dacă B este un subtip al lui A

ArrayList a<Integer>  
a.add("Data");

↳ b<String>

```

Considerând următorul cod Java, care din fragmentele de cod enumerate mai jos vor produce o eroare la poziția indicată cu /*AICI*/?

```

```

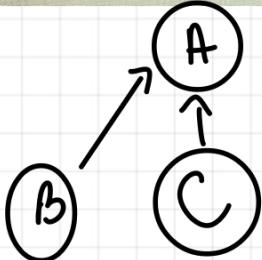
abstract class A {
    public void m(int p) {}
    public void r(int p) {}
}

class B extends A {
    public void m(int p) {}
    public void p(int p) {}
    public void q(int p) {}
}

class C extends A {
    public void m(int p) {}
    public void p(int p) {}
    public void r(int p, int t) {}
}

class Main {
    public static void main(String argv[]) {
        A a = new B();
        C c = new C();
        /*AICI*/
    }
}

```



Selectați unul sau mai multe:

- c.equals(a); ✓
- c.r(1); ✓
- a.q(1); ✗
- a.p(1); ✗
- a.m(1); ✓

Ce valoare întreagă se va tipări la linia marcată cu /\* în urma execuției programului de mai jos?

```

class Ex extends RuntimeException {}
class Main {
    public static void m(int x) throws Ex {
        if(x == 2) throw new Ex();
    }
    public static void main(String argv[]) {
        int a = 2;
        try {
            for(int i = 0; i < 4; i++) {
                m(i);
                a = a + 2;
            }
        } catch(Ex e) {
            a++;
        } finally {
            a++;
        }
        System.out.println(a);/*
    }
}

```

Răspuns: 8

$a=2$

try [  $i=0$   
 $m(0)$   
 $a=4$   
 $i=1$   
 $m(1)$   
 $a=6$   
 $i=2$   
 $m(2)$   
 $\rightarrow \text{catch}$

$a=7$   
 $a=8$  (finally)

1,00  
e cu

Ce valoare întreagă se va tipări la linia marcată cu /\* în urma execuției programului de mai jos?

```

class A {
    private static int x;
    private int y;
    public A(int x1, int y1) { x = x1; y = y1; }
    public void setXY(int x1, int y1) { x = x1; y = y1; }
    public int getX() { return x; }
    public int getY() { return y; }
}
class Main {
    public static void m(A z, A t) {
        z.setXY(3,20); *
        z = new A(2,17); *
    }
    public static void main(String[] args) {
        A a = new A(9,50);
        A b = new A(7,28); *
        m(a,b);
        System.out.println(a.getY() + b.getX());/*
    }
}

```

Răspuns:  

$20 + 2 = 22$

~~$x \rightarrow 8$~~   ~~$+ 32$~~

$a \rightarrow \cancel{50} \quad 20$

$b \rightarrow 28 \quad + \quad 17$

$m(a, b)$

$a : \quad 2 \mid 20$

$b : \quad 2 \mid 28$

✓ rezolvare  
înțeles  
despre lucru  
Marcat de 1.00  
P. întrebare cu flag

Considerând următorul cod Java, care din fragmentele de cod enumerate mai jos vor produce o eroare de compilare când sunt inserate la poziția indicată cu /\*AICI\*/?

```
class A {  
    protected int a;  
    public A(int a) {  
        this.a = a;  
    }  
}  
class B extends A {  
    private int b;  
    /*AICI*/  
}
```

Selectați unul sau mai multe:

public B(int a, int b) {  
 super(a);  
 this.b = b;  
}



public B(int a, int b) {  
 this.a = a;  
 this.b = b;  
}

ar fi corect doar dacă  
în loc de m aveam B și să lipsească și void

public B(int b) {  
 super(b);  
 this.b = b;  
}



public B(int a, int b) {  
 this.b = b;  
 super(a);  
}

→ trebuie să fie primul apelat

Considerând următorul cod Java, care din fragmentele de cod enumerate mai jos vor produce o eroare de compilare când sunt inserate la poziția indicată cu /\*AICI\*/?

```
class A {  
    public A m(int a) {  
        return new A();  
    }  
}  
class B extends A {  
    public void n(A a) {  
        return;  
    }  
    /*AICI*/  
}
```

Selectați unul sau mai multe:

public int n(A a) {  
 return 0;  
}

diferă prim tip  
public int m (A a, B b) ✓



public int n(B a) {  
 return 0;  
}

✓ overload

public A m(B a) throws Exception {  
 throw new Exception("Message");  
}

→ excepție verificată

public A m(int p) {  
 return new B();  
}

→ override

→ ar fi corect pentru o excepție nverificată (Runtime Exception)

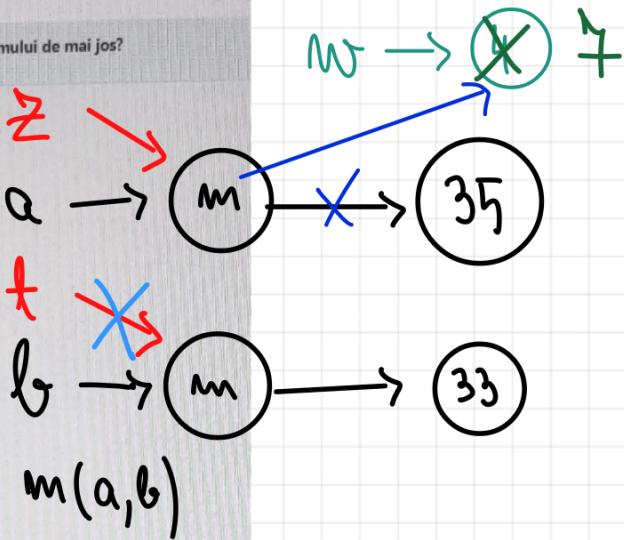
ebare  
primit  
ns încă  
at din 1.00  
zrebare cu

Ce valoare întreagă se va tipări la linia marcată cu /\* în urma executiei programului de mai jos?

```
class A {  
    private int x;  
    public A(int x) { this.x = x; }  
    public void setX(int x) { this.x = x; }  
    public int getX() { return x; }  
}  
class B {  
    private int y;  
    public B(A y) { this.y = y; }  
    public void setY(A y) { this.y = y; }  
    public A getY() { return y; }  
}  
class Main {  
    public static void m(B z, B t) {  
        A w = new A(4);  
        z.setY(w);  
        w.setX(7);  
        t = new B(new A(3));  
    }  
    public static void main(String[] args) {  
        B a = new B(new A(35));  
        B b = new B(new A(33));  
        m(a,b);  
        System.out.println(a.getY().getX() + b.getY().getX());/*  
    }  
}
```

$$7 + 33 = 40$$

Răspuns:



re  
nit  
ncă  
in 1.00  
are cu

Ce valoare întreagă se va tipări la linia marcată cu /\* în urma executiei programului de mai jos?

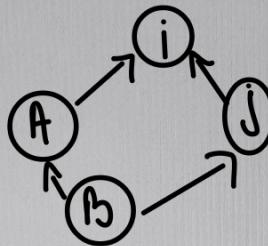
```
class Ex1 extends Exception {}  
class Ex2 extends Ex1 {}  
class Main {  
    public static void m(int x) throws Ex1 {  
        if(x == 1) throw new Ex2();  
    }  
    public static void main(String argv[]) {  
        int k = 0;  
        int a = 1;  
        while (k < 2) {  
            try {  
                k++;  
                m(k);  
                a++;  
            } catch(Ex2 e) {  
                a += 2;  
            } catch(Ex1 e) {  
                a += 3;  
            } finally {  
                a++;  
            }  
        }  
        System.out.println(a);/*  
    }  
}
```

Răspuns: 6

$b = 0$   
 $a = 1$   
—  $b = 0 \& 2$   
try —  $b = 1$   
 $m(1)$   
 $a = 3$   
→ finally  
 $a = 4$   
 $b = 1 < 2$   
try —  $b = 2$   
 $m(2)$   
 $a = 5$   
→ finally  
 $a = 6$

Considerând următorul cod Java, care din fragmentele de cod enumerate mai jos vor produce o eroare de compilare când sunt inserate la poziția indicată cu /\*AICI\*/?

```
interface I {
    public void g();
}
interface J extends I {
    public void h();
}
abstract class A implements I {
    public void m() {}
}
class B extends A implements J {
    public void p() {}
    public void g() {}
    public void h() {}
}
class Main {
    public static void main(String argv[]) {
        J j = new B();
        A a = new B();
        /*AICI*/
    }
}
```



*newB();*

Selectați unul sau mai multe:

j.g();

a.p();

a.g();

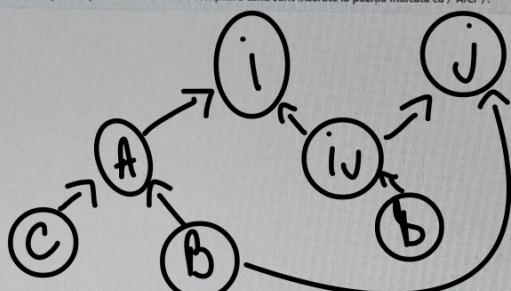
j.m();

a.h();

11°C Sunny

Considerând următorul cod Java, care din fragmentele de cod enumerate mai jos vor produce o eroare de compilare când sunt inserate la poziția indicată cu /\*AICI\*/?

```
interface I {}
interface J {}
interface IJ extends I,J {}
class A implements I {}
class B extends A implements J {}
class C extends A {}
class D implements IJ {}
class Main {
    public void doSomething(I i, J j, IJ ij) {
        A a = new A();
        B b = new B();
        C c = new C();
        D d = new D();
        /*AICI*/
    }
}
```



*i=c* ✓

Selectați unul sau mai multe:

j = c;

b = d;

i = c; ✓

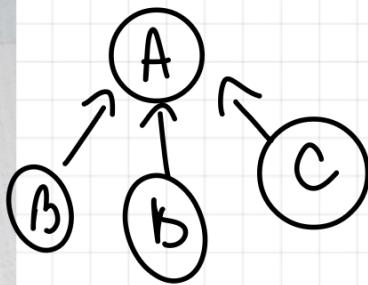
ij = j;

i = ij;

*downcast*

Ce valoare întreagă se va tipări la linia marcată cu // în urma execuției programului de mai jos?

```
class A {  
    public int m(A p) { return 8; }  
    public int n(A p) { return 7; }  
}  
class B extends A {  
    public int m(B p) { return 2; }  
    public int n(A p) { return 44; }  
}  
class C extends A {  
    public int m(A p) { return 4; }  
    public int n(A p) { return 4; }  
}  
class D extends A {  
    public int m(A p) { return 11; }  
    public int n(D p) { return 30; }  
}  
class Main {  
    public static void main(String argv[]) {  
        A x = new C();  
        A y = new D();  
        D z = new D();  
        System.out.println(x.n(z) + y.m(z));/*  
    }  
}
```



$$x.m(z) \rightarrow 4$$

$$y.m(z) \rightarrow 11$$

15

Răspuns: 15

Considerând următorul cod Java, care din expresiile enumerate mai jos vor avea valoarea de adevar FALSE când sunt evaluate la poziția indicată cu /\*AICI\*/?

```
class A {}  
class B extends A {}  
class Main {  
    public static void main(String argv[]) {  
        A a = new B();  
        B b = new B();  
        Object o = new A();  
        System.out.println(/*AICI*/);  
    }  
}
```



Selectați unul sau mai multe:

- A.class.equals(a.getClass()) → false
- A.class.isInstance(b) → true
- o.getClass() equals(Object.class) → false
- a.getClass().equal(b.getClass()) B → true
- B.class.isInstance(a) → true

Considerând următorul cod Java, care din fragmentele de cod enumerate mai jos vor produce o eroare de compilare când sunt inserate la poziția indicată cu /

Timp rămas 0:21:08

```
//A.java  
package p1;  
public class A {  
    private void m() {}  
    void n() {}  
    protected void p() {}  
    public void q() {}  
}
```

```
//B.java  
package p2;  
import p1.A;  
public class B extends A {  
    private void r() {}  
    void s() {}  
    protected void t() {}  
    public void k() {}  
}
```

```
//C.java  
package p2;  
import p1.A;  
public class C {  
    private void z() {}  
    public void v(A a, B b, C c) {  
        /*AICI*/  
    }  
}
```

Selectați unul sau mai multe:

- b.s(); ✓
- c.z(); ✓
- a.p();
- a.n();
- b.t(); ✓

2 întrebări

Nu a primit

răspuns încă

Mai puțin decât

1% înregistrare

cum

Înregistrare 1 (page 2 din 10) | CV - Mozilla Firefox

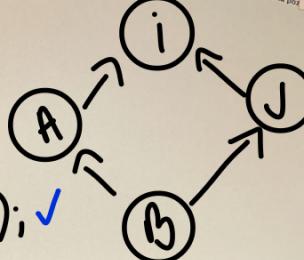
Săpt 20 Ian, 09:11

Timp rămas 0:27:13

```
Considerând următorul cod Java, care din fragmentele de cod enumerate mai jos vor produce o eroare de compilare când sunt inserate la poz.
```

Selectați unul sau mai multe:

- b.m(); ✓
- if (i instanceof B) {  
 i.m();  
}
- ((A)i).p(); ✓
- ((B)j).m();
- ((J)a).h(); ✓



answered  
Marked out of 3.00  
Flag question

```
class Pair {  
    private int v;  
    private int t[];  
    public Pair(int x, int[] y) { v = x; t = y; }  
    public void setV(int x, int[] y) { v = x; t = y; }  
    public void setT(int x, int i, int y) { v = x; t[i] = y; }  
    public int getV() { return v; }  
    public int[] getT() { return t; }  
}  
  
class Main {  
    public static Pair call(Pair p, Pair q) {  
        p.setV(76, q.getT());  
        p = q; //  
        p.setV(87, 0, 74);  
        int tz[] = {97, 93};  
        return new Pair(19, tz);  
    }  
  
    public static void main(String[] args) {  
        int ta[] = {29, 35};  
        int tb[] = {6, 87};  
        Pair a = new Pair(3, ta);  
        Pair b = new Pair(50, tb);  
        Pair c = b;  
        b = call(a, c);  
        System.out.println(c.getV()); /*  
    }  
  
După apelare:  
a={v=76, t=[74, 87]}  
b={v=19, t=[97, 93]}  
c={v=87, t=[74, 87]}
```

