

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

```
abstract class A
{
    public int proc(A p)
    {
        return 98;
    }
}

class B extends A
{
    public int proc(A p)
    {
        return 17;
    }
}

class C extends A
{
    public int proc(C p)
    {
        return 65;
    }
}

public class Main
{
    public static void main(String argv[])
    {
        C x = new C();
        A y = new B();
        C z = new C();
        System.out.println(y.proc(x) + z.proc(x)); /*
    }
}
```

A handwritten addition problem is shown. It consists of two numbers, 17 and 65, aligned vertically under a plus sign (+). A horizontal line is drawn below the numbers, indicating the sum. The number 17 is written with a large '1' and a smaller '7'. The number 65 is written with a large '6' and a smaller '5'.

$$\begin{array}{r} 17 \\ + 65 \\ \hline \end{array}$$

Răspuns: $17 + 65 = 82$

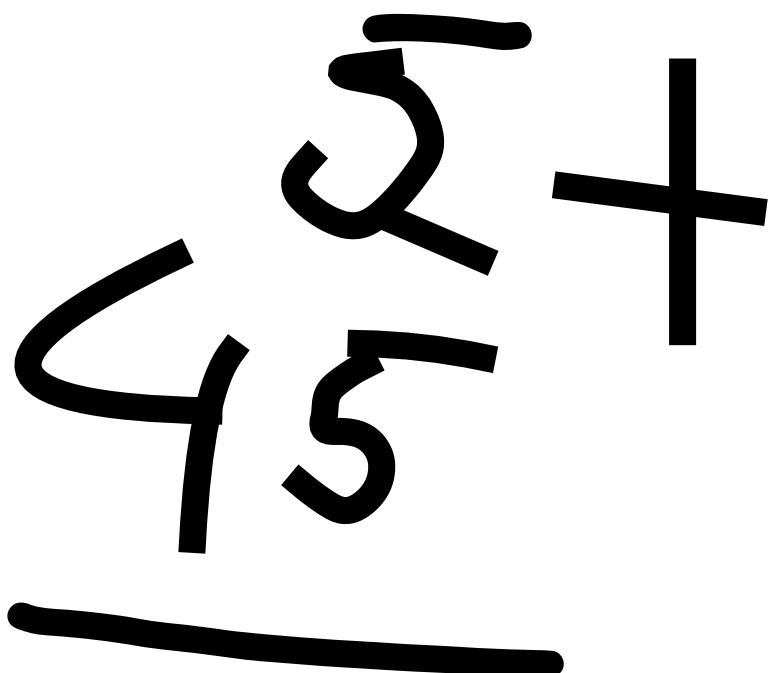
Ce valoare întreagă se va tipări la linia marcată cu /* în urma executiei programului de mai jos:

```
abstract class A
{
    public int proc(B p)
    {
        return 5;
    }
}

class B extends A
{
    public int proc(C p)
    {
        return 21;
    }
}

class C extends B
{
    public int proc(C p)
    {
        return 45;
    }
}

public class Main
{
    public static void main(String argv[])
    {
        A x = new C();
        C y = new C();
        C z = new C();
        System.out.println(x.proc(z) + y.proc(z)); /*
    }
}
```



Răspuns: $5 + 45 = 50$

4

yet

med

old out of

question

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

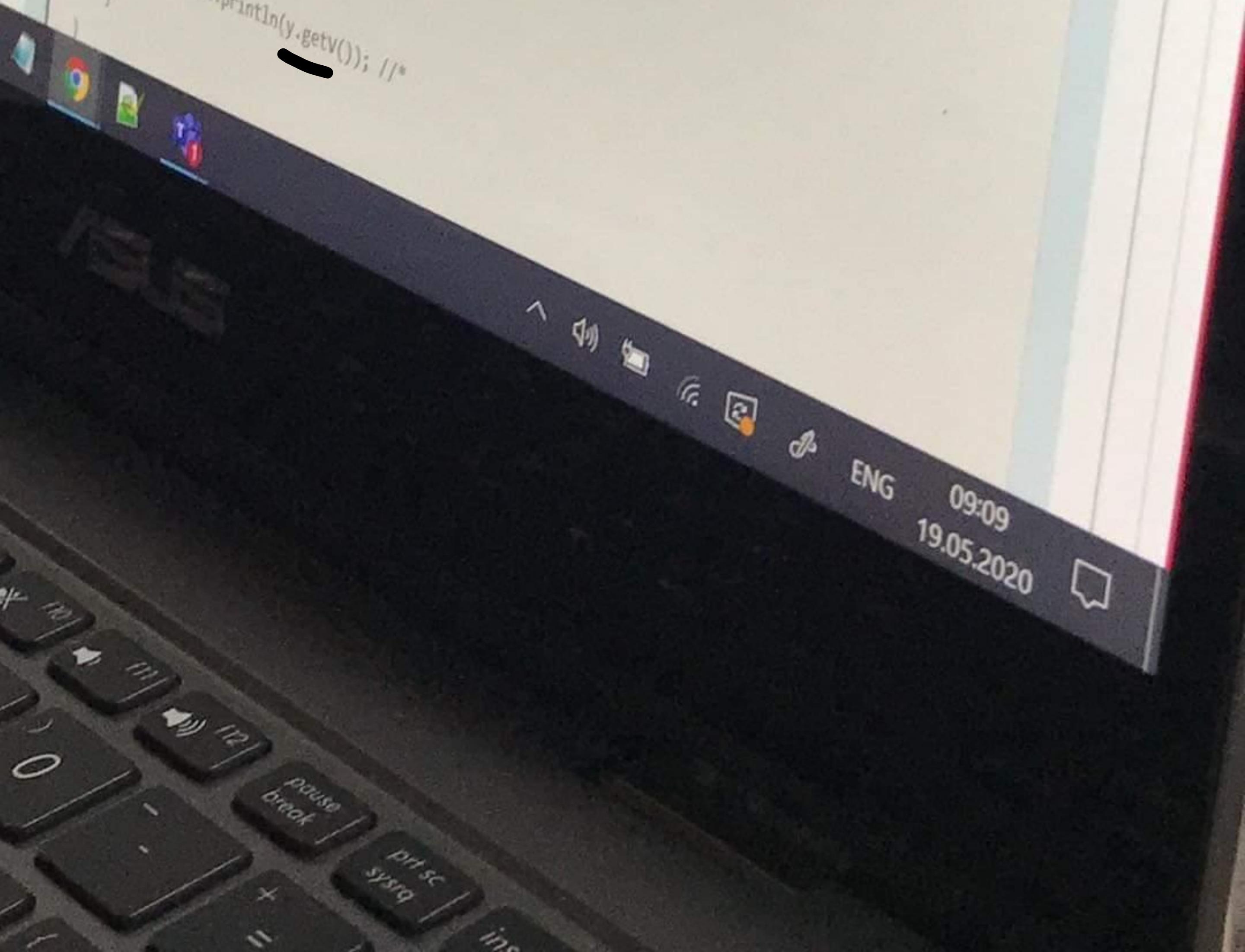
```
class X {  
    protected int v = 0;  
    public X() { v += 43; }  
}  
  
class Y extends X {  
    public Y() { v -= 24; }  
    public int getv() { return v; }  
}  
  
class Z extends Y {  
    public Z() { v += 100; }  
}  
  
class Main {  
    public static void main(String argv[]) {  
        X x = new Z();  
        Y y = new Y();  
        Z z = new Z();  
        System.out.println(y.getv()); /*  
    }
```



43 -
24



ENG 09:09
19.05.2020



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

```
abstract class A {
    public int proc(B p) { return 5; }
}

class B extends A {
    public int proc(C p) { return 21; }
}

class C extends B {
    public int proc(C p) { return 45; }
}

public class Main {
    public static void main(String argv[]) {
        A x = new C();
        C y = new C();
        C z = new C();
        System.out.println(x.proc(z) + y.proc(z)); /**
    }
}
```

5 +
45

Answer:

answered

Marked out of
3.00

Flag question

74

```
class Pair {  
    private int v;  
    private int t[];  
    public Pair(int x, int[] y) { v = x; t = y; }  
    public void setVT(int x, int[] y) { v = x; t = y; }  
    public void setVIT(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.setVT(76, q.getT());  
        p = q;  
        p.setVIT(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}}

19 97 93

}

Question 1

Not yet
answered

Marked out of
3.00

Flag question

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

```
abstract class A {  
    public int proc(A p) { return 20; }  
}  
  
class B extends A {  
    public int proc(A p) { return 100; }  
}  
  
class C extends A {  
    public int proc(C p) { return 75; }  
}  
  
public class Main {  
    public static void main(String argv[]) {  
        C x = new C();  
        A y = new B();  
        C z = new C();  
        System.out.println(y.proc(x) + z.proc(x)); /*  
    }  
}
```

$$\begin{array}{r} \cancel{20} \\ \cancel{100} \\ + 75 \\ \hline \end{array}$$

Answer: 175 ✓

5.1.5a) Se dau clasele care urmează:

```
class Card {  
    public void deschideVisaCard();  
    String beneficiar, pozitieSume();  
}
```

```
class OrdinFizic {  
    public void platesteVisaCard(  
        String beneficiar, double suma);  
}
```

- Făcă modificări în cadrul celei două clase, implementând într-o clasă Executor o metodă realizată în pafi ce primește ca parametru un tablou de beneficiari (string-uri), un tablou de sume de plată (double-uri) și un tablou ce conține referințe la instanțe a claselor anterioare reprezentând modalități de efectuare a unei plăți (obiecturi tablouri). Metoda platește fiecărui beneficiar de pe poziția i din primul tablou sumă de pe poziția i din al doilea tablou. Implementarea de plată: [din al treilea tablou]. Implementarea se face încălcând principiul open-closed în report cu adăugarea de noi funcții de efectuare a platelor (dacă se consideră că cele trei tablouri sunt date în timpul cu același număr de elemente).
- Implementări metoda de la punctul a) dar respectând principiul open-closed în report cu adăugarea de noi feluri de efectuare a unei plăți. Se implementă ocazional modificări considerate necesare (inclusiv asupra claselor înțelese care trebuie să continue să existe cu aceeași funcționalitate).

6.1.6a) În report cu codul Java altădată, răspunsul la întrebări și motivatii răspunsul în 4-5 propozitii, explicând casă cu pas cu care au luat la rând valoarea.

a. Ce se va întâlni la linia marcată cu (1)?

b. Ce se va întâlni la linia marcată cu (2)?

```
class Tuplu {  
    private int v;  
    private Tuplu nr;  
    public Tuplu(int v, Tuplu nr) { this.v = v; this.nr = nr; }  
    public void saluTuplu(Tuplu nr) { v = 2 * v + nr.v; }  
    public int getValoare() { return v; }  
    public Tuplu getNr() { return nr; }  
}  
  
class Main {  
    public static Tuplu raspetaTuplu(Tuplu t, Tuplu nr) {  
        if (nr == null) {  
            return t; }  
        else {  
            t.getNr().saluTuplu(nr);  
            return new Tuplu(t.getValoare(), new Tuplu(nr)); }  
    }  
  
    public static void main(String[] args) {  
        Tuplu x = new Tuplu(3, new Tuplu(2, null));  
        Tuplu y = new Tuplu(1, new Tuplu(1, null));  
        Tuplu z = x;  
        z = raspetaTuplu(y);  
        System.out.println(z.getVal());  
        System.out.println(z.getNr().getVal());  
    }  
}
```

NOTĂ: Unde se cere motivatii, dacă aceasta lipseste, ori nu e obiect, ori nu e corecta, ori contine greseli, ori e ambigua etc., raspunsul nu se puncteaza

R1C2 Probleme

Odă cu creșterea numărului de angajați, departamentul de management al unei companii din Timisoara are nevoie de un sistem pentru gestionarea training-urilor oferite angajaților. Elementul central al sistemului nostru va fi angajatul. Un angajat este caracterizat de: i) un atribut de tip long ce reprezintă numărul de identificare al acestuia. În cadrul departamentului; ii) un String ce reprezintă codul departamentului; iii) un double reprezentând numărul anilor de experiență și iv) de o secvență (tablică sau lista) numită lista de String-uri reprezentând tehnologile preferate / de interes ale angajantului. Toata această vor fi setată în momentul instantierii unui nou angajat. Dacă angajații sunt unuia, aceleași date numerice de cedulare și codurile departamentele coindice.

Sistemul este confinat cu toate felurile de training-uri. Ambula memorizează întreaga secvență lăsată de angajați înainte sau lista înscrisă în training-ul respectiv, precum și un double reprezentând numărul recomandării de ani de experiență pentru a participa la acel training (double). Experiența recomandată este setată la crearea training-ului și numărul maxim de participanți și devine de tipul efectiv al training-ului. De asemenea, toate fizurile de training-uri dispun și o metoda verificăRataDeSuccess care primește ca unic argument un angajat și întoarcă sub forma unei intervale în intervalul [0,100] gradul de satisfacție al angajatului după efectuarea acestuia training.

La training-urile ce soft skills pot participa maxim 10 angajați pentru că presupune multe interacțiuni și și-aferențe. Un astfel de training va pună la dispoziție o metodă adăugaAngajat care primește ca argument un angajat. Metoda verifica dacă angajatul nu a deținut și secvența de participanți astfel în care nu se întâmplă nimic. Altfel, dacă nu este loc, se adaugă angajatul la secvența de participanți. Altfel, metoda aruncă o excepție verificăLimităAlesă (Nota: exceptia se consideră definită și nu trebuie să aibă definită văză). Modul de calcul al ratiu de succes este următorul: dacă mediu și diferență dintre anii de experiență recomandării și anii de experiență al angajatului sunt în intervalul [0,1], metoda verificăRataDeSuccess întoarcă valoarea 100. Dacă acest valoare se află în intervalul [1,2] metoda va întoarce 80. Altfel se va întoarce 5.

Suntem rankin-g-urile tehnici și numărul limită de participanți și anume 20. Si un astfel de training va pună la dispoziție metoda adăugaAngajat care primește ca argument un angajat. Metoda verifica dacă angajatul nu a deținut și secvența de participanți astfel în care nu se întâmplă nimic. Altfel, dacă nu este loc, se adaugă angajatul la secvența de participanți. Altfel, metoda aruncă o excepție verificăLimităAlesă. În plus, la crearea unui training tehnic, pe langa anii recomandării de experiență, se specifică și o secvență (tablică sau lista cu orice elemente) ce conține elemente de tip String reprezentând tehnologile abordate la respectivul training. Pentru acest fel de training, rata de succes va fi chiar procentul de tehnologii preferate ale angajatului ce se regăsesc în lista cu tehnologii abordate la training (procentul se va rotunji la întregul intervalul [0,100]).

Pentru a responsabiliza de managementul training-urilor și al angajaților este manager-ului de resurse umane. Un astfel de obiect va memora întreaga căde și secvența numită lista de angajați, respectiv ce training-uri, ambele simple la crearea managerului. Managerul va avea și o metodă distribuieAngajat prin care se adaugă la fiecare training multe angajați. Adăugarea la fiecare training are loc astfel: se adaugă pe rand angajații a căror interes pentru acel training e mai mare sau egal cu 100. Apoi se adaugă angajații a căror interes pentru acel training e de minim 99. Si astăzi nu există, până când se adaugă, și cel cu interesul de minim 1 (acezând pragul cu un punct de lecție dată). Dacă în un moment dat nu mai este loc la acel training, procedura se oprește (ramânând inserarea angajaților care au fost deja adăugăți și ignorându-i pe restul). Metoda implementează inserarea angajaților care au fost adăugăți și ignorându-i pe restul. Metoda

implementează în Java clasa deosebită, imprimându-și clasele considerate necesare întreaga numărul de inserări efectuate cu succes.

S. C. E. B. S. G. M. C. P. C. C.

John B. Kelly
Dartmouth
General Engineering Department

2017-01-
XBT-CV23
2017-01-20 09:00:00

11.5pt în raport cu colectivitatea și respectul legăturilor la trăbou și motivatii sprijinire în 45 procent, în ceea ce priveste pas în pas cu

- a. Ce se va întâmpla în
mercău în 2012?
 - b. Ce se va întâmpla în
mercău în 2013?

```
class Person {
    private int id;
    private String name;
    public Person(int id, String name) {
        this.id = id;
        this.name = name;
    }
    public int getId() {
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
}
```

```
    new(Ma,3)  
    let L = Ma.create(spine,apex,Plane(Ma,  
        Box(Ma,  
        1,  
        1))  
    new(Perched,4,1)
```

```
lizenzen von man2010@...  
PorscheX -> Porsche2, NewPorsche12, NewPorsche13  
PorscheY -> Porsche11, NewPorsche12, NewPorsche13  
PorscheZ -> X,  
X -> P1,P2,P3,P4,P5  
System will terminieren
```

RIC-1 Prob 2

On can see that the mean age of patients in the study is 50 years old. The mean age of patients in the control group is 50.1 years old. The mean age of patients in the intervention group is 50.2 years old. The difference between the two groups is not statistically significant.

Concurența între partile politice și între ele, dar și situația unei programe politice care să nu obțină un rezultat în spatele voturilor, va determina o situație similară de dezbatere rezolvată și în același sens, la unii să fie reprezentate unele voci mai puternice, cum sunt cele ale lui Gheorghe Hagi, sau ale lui Ion Iliescu, care îl consideră un "bun președinte".

A continuación se presentan los resultados de la encuesta realizada a los participantes que respondieron a la encuesta. Los datos se presentan en forma de tablas y se describen las principales tendencias observadas.

Dicho concepto más característico de la teoría es la **comunicación**. Jacob Lewin consideraba que el manejo de los elementos y sus soluciones de problema. La comunicación es el elemento central de un sistema jerárquico ya sea estableciendo relaciones de problema o no de él. La comunicación entre las unidades determinante de memoria y las unidades de respuesta. La memoria ya tiene conocimiento, ya tiene malos problemas resueltos. Toda la comunicación se basa en la memoria, porque todo lo que se dice se va a considerar y se va a aplicar con base en lo que ya se ha memorizado. Varias interacciones entre las unidades de memoria y las unidades de respuesta. Una persona ya no piensa lo que ya piensa. Una persona ya no piensa lo que ya piensa. Una persona ya no piensa lo que ya piensa.

Concurs regional este caracterizat ca o acțiune cu sub-concursuri de orice tip, recunoscute specifică într-o anumită perioadă (ca, de exemplu, concursul național de elevi din cadrul Liceului). Într-un astfel de concurs se poate da și premiu de la "ceea ce nu este concursul propriu-zis". Apărarea creșării banchetelor și a pregiulor de concursuri voluntară îl pot aduce la un nivel de concursuri deosebit de scăzut. În final, nu va exista concursuri regionale și nici chiar și urmărește.

Su su imprimado si dice cada director, figura tal de los dos videntes que han comprendido la idea clara de su escena.

NOTA: Unde se poate observa, dacă rezultă lăptea, că nu e la doar, că nu e la doar, că nu e la doar.

PHOTO: Deonna

1. Cada vez que se realizó una actividad, respondió al director o directora encargado de la actividad y se le pidió que la describiera.

2. Cada actividad se realizó en un horario de 17:00 a 18:00 horas.

3. Cada actividad se realizó en el mismo horario de 17:00 a 18:00 horas.

```

class F {
protected: void f() { }
public: void g() { }
public: friend void h() { }
};

class D : public F {
public: void i() { }
};

class E : public D {
public: void j() { }
};

class G {
public: static void main() {
    D d;
    d.f();
    d.g();
    d.h();
    d.i();
    d.j();
}
};

```

[View Details](#) | [Edit](#) | [Delete](#) | [Check off](#) | [Import](#) | [Export](#) | [Print](#)

- Un nou instrument de politica intelectuală în
cadrul Administrației publice.

```
class A {
    public void m1() { System.out.println("A"); }
}

class B extends A {
    public void m2() { System.out.println("B"); }
    public void m3() { m1(); }
}

class C {
    public static void main(String args[]) {
        A a = new B();
        a.m3();
    }
}
```

It is about as easy to find them
when they are in the middle
of a cluster like this as it is to
find them when they are
scattered.

2012. These items include, but are not limited to, the cost of software, on-going phone, or a telephone number to be purchased.

to 12 feet in diameter, has smooth, yellowish

- Suntori pe 4K va trebui să fie menținute în 10 și moșteniri disponibile în 4.5 proiecte, exprimându-se în numerele următoare de la secvențe.
 - Suntori dacă elementele sunt urmărite și corectă și înțeleasă și modul disponibil în 4.5 proiecte. Deosebiti un proiect cu 10 secvențe nu este doar ca să se scrie în secvență ca 101, ci poate ar fi o combinație de secvențe deosebite care să fie urmărite separat.

www.François-XavierDesPuy.com

pink is second, the first is light blue.

```
class B extends A {
    public int add(int m) throws Exception {
        if (m < 0)
            throw new Exception("m<0");
        else
            return m + 1;
    }
}
```

```
class Node {
    public int val;
    public Node left;
    public Node right;
}
```

12. *Polypodium* 12.

卷之三

Public 100%
Public 100% + Friends
Public 100% - 0

Einzelne Z-Orte
z. verhindern die
Spiele der
Akteure

4. (3.3) In week ten during long distance competition, the number of minutes completed in 45 seconds

1. Există o îmbunătățire a i
potențialului săptămânii și
potrivită următoarelor săptămâni
în cadrul unui anual de lucru în Zile
Naturale este de la 10 la 20%
 2. În cadrul anului de lucru naturale se
împart în trei perioade de cinci săptămâni
cadastrale și în trei perioade de cinci săptămâni
cadastrale.

R2C2 Prob

Gigel este pasionat de programare și urmărește diariile concurenții de algoritmici și ar fi putut să poată simula un astfel de concurs și să ceră nouă titlu. Un diarist în modulă prezintă problema următoare: concursul este problema. O problemă are ca atunci un diarist să scrie numărul ce reprezintă dificultatea. Aceasta este un număr aleator în intervalul $[0,1]$. Rândom din clasa Math generează un număr întreag ce reprezintă numărul de concurenți. Orice concurrent este caracterizat printr-un interval în care au fost realizate prezentările (initial 0), un întreg ce reprezintă timpul total în care concurenții au prezentat proiectele lor. De asemenea, orice concurrent dispune de un parametru o problemă. Există două feluri de concurenți: unii cu dificultatea problemelor rezolvată și unii cu probleme nerezolvate. În funcție de cum sunt rezolvate problemele, începatorul poate rezolva sau nu problemele nerezolvate. În funcție de cum sunt rezolvate problemele rezolvate, începatorul poate rezolva sau nu problemele rezolvate.

cu codul Java alăturat:

```

class MyExceptA extends Exception { }
class MyExceptB extends MyExceptA { }

class A {
    public int make(int x) throws MyExceptA {
        return 1;
    }
}

class B extends A {
    public int make(int a) throws MyExceptA {
        for(int k = 0, k < 2, k++) {
            if(a==0) throw new MyExceptA();
            a = a - 3;
        }
        return a;
    }
}

class Main {
    public static void main(String args[]) {
        B b = new B();
        int x = 0;
        for(int i = 0 ; i < 2, i++) {
            try {
                x = b.make(x);
            } catch(MyExceptA e) {
                System.out.println(e);
            } finally {
                x++;
            }
        }
        System.out.println(x); //?
    }
}

```

Concursul său este mai deosebit de interesant, deoarece adaugă la tehnica de programare o dimensiune socială și motivată.

"Dacă la poziții: "Dacă la sterge clauza "compilabil" și se va tipări la linia și motivată răspunsul explicând pas cu pas aceea valoare.

Concursul său este mai deosebit de interesant, deoarece adaugă la tehnica de programare o dimensiune socială și motivată.

"Dacă la poziții: "Dacă la sterge clauza "compilabil" și se va tipări la linia și motivată răspunsul explicând pas cu pas aceea valoare.

Se încearcă să se rezolve problema de după:

Propriile clase

Se dă o clasă numeUnu și se cere să se scrie o altă clasă numeDoi care să extindă numeUnu.

Se dă următoarea informație:

```

    package numeUnu;
    public class A {
        protected int x;
    }

    package numeUnu;
    public class B extends A {
        public void doSomething() {
            // ...
        }
    }
  
```

Se cere să se scrie o nouă clasă C care să extindă B și să adauge o metodă doSomething(A a) care să returneze valoarea lui x din obiectul a.

Se dă răspunsul:

```

    package numeUnu numeDoi;
    import numeUnu.A;
    import numeUnu.B;
    public class C extends B {
        public void doSomething(A a) {
            // ...
        }
    }
  
```

Se cere să se analizeze răspunsul și să se determine dacă este corect sau nu.

6. (1.5p) În raport cu codul Java a
în 4-5 propozitii

 - a. La linia marcată cu //1 putem
accesa câmpul x al obiectului
referit de referința a
 - b. Clasele A și
implementația

NOTĂ: Unde se cere motivatie, daca aceasta lipseste

R2C1 Teorie

1. (1.5p) Se dau clasele care urmează:

```
class Card {  
    public void platiesteViaCard(  
        String beneficiar, double suma) {  
    }  
}
```

```
class OranPlata {  
    public void platiesteViaOran(  
        String beneficiar, double suma) {  
    }  
}
```

- a. Fără a modifica în vreun fel cele două clase, implementați întări-o clasă Executor o metodă realizeazaPlati ce primește ca parametru un tablou de beneficiari (stringuri), un tablou de sume de plată (double-uri) și un tablou ce conține referințe la instante a claselor anterioare reprezentând modalitatea de plată (o poziție / din obligatoriu tablouri). Metoda plătește / (din al doilea tablou) efectuarea se face indicând principiul open-closed / (din al treilea tablou). Implementarea de pe pozitia / din al doilea tablou încărcând principiul open-closed în raport cu adăugareaa de noi tipuri de efectuare a platelor. Se implementează consideră că cele trei tablouri sunt date tot timpul cu același număr de elemente.
- b. În raport cu adăugareaa de noi feluri de efectuare a unei plăti. Se implementează în raport cu adăugarea de noi feluri de efectuare necesare (inclusiv funcționalitate), orice alte modificări considerate necesare. Considerați că clasele inițiale care trebuie să continute să existe cu aceeași funcționalitate.

2. (1.5p) În raport cu codul Java alăturat, răspundeți la întrebări și explicăți pas cu pas cum se ajunge la aceea valoare.

- a. Ce se va tipări la linia marcată cu l/l ?
- b. Ce se va tipări la linia marcată cu l/l ?

```
class A {  
    protected static int x = 0;  
    public A() {x = 10;}  
    public int exec(A p) {return 45;}  
}  
  
class B extends A {  
    public int exec(B p) {return 37;}  
}  
  
class C extends B {  
    public C() {x += 10;}  
    public int exec(C p) {return 69;}  
    public static int getX() {return x;}  
}  
  
class Main {  
    public static void main(String args[]) {  
        A a = new A();  
        A b = new B();  
        System.out.println(C.getX());  
        C c = new C();  
        System.out.println(c.getX());  
    }  
}
```

a. În codul alăturat, la linia marcată cu l/l, avem o eroare de compilare.

b. În raport cu codul alăturat, avem o referință "Object" extindere (fără de referință corectă) fără "astfel de inițializare" corectă. Înțeles că apelarea B(). este corectă la compilare.

3. (1.5p) În raport cu codul Java alăturat, răspundeți la întrebări și explicăți pas cu pas cum se ajunge la aceea valoare.

- a. Ce se va tipări la linia marcată cu l/l ?
- b. Ce se va tipări la linia marcată cu l/l ?

```
class L {  
    public int exec() {return 12;}  
}  
  
class B extends A {  
    public int exec() {return 1;}  
    public static void main(String args[]) {  
        L l = new L();  
        System.out.println(l.exec());  
    }  
}
```

3. (1.5p) În raport cu codul Java alăturat, răspundeți la întrebări și explicăți pas cu pas cum se ajunge la aceea valoare.

- a. Ce se va tipări la linia marcată cu l/l ?
- b. Ce se va tipări la linia marcată cu l/l ?

```
class Table {  
    protected int x;  
    public Table() {  
        public Table A = Table.this; // Line 1  
        public Table B = Table.this; // Line 2  
        public Table C = Table.this; // Line 3  
        public Table D = Table.this; // Line 4  
        public Table E = Table.this; // Line 5  
        public Table F = Table.this; // Line 6  
        public Table G = Table.this; // Line 7  
        public Table H = Table.this; // Line 8  
        public Table I = Table.this; // Line 9  
        public Table J = Table.this; // Line 10  
        public Table K = Table.this; // Line 11  
        public Table L = Table.this; // Line 12  
        public Table M = Table.this; // Line 13  
        public static Table Faza faza = Table.this; // Line 14  
        public Table N = Table.this; // Line 15  
        public Table O = Table.this; // Line 16  
        public Table P = Table.this; // Line 17  
        public Table Q = Table.this; // Line 18  
        public Table R = Table.this; // Line 19  
        public Table S = Table.this; // Line 20  
        public Table T = Table.this; // Line 21  
        public Table U = Table.this; // Line 22  
        public Table V = Table.this; // Line 23  
        public Table W = Table.this; // Line 24  
        public Table X = Table.this; // Line 25  
        public Table Y = Table.this; // Line 26  
        public Table Z = Table.this; // Line 27  
    }  
}
```

4. (1.5p) Spuneți dacă afirmația a cărui se respectă și motivatiți dumneata în 4.5 propozitii.

- a. În codul alăturat, la linia marcată cu l/l, avem o eroare de compilare.

```
class A {  
    public int exec() {return 12;}  
}  
  
class B extends A {  
    public int exec() {return 1;}  
    public static void main(String args[]) {  
        A a = new A();  
        System.out.println(a.exec());  
    }  
}
```

1. (1.5p) Se dă clasele care urmează:

```
class Some {
    public void sendString(String dest, String msg) {
        // ...
    }
}
```

```
class Fax {
    public void sendFax(String dest, String msg) {
        // ...
    }
}
```

- a. Pără a modifica în vreun fel cele două clase, implementați într-o clasă Executor o metodă `transmitMessage` ce primește ca parametru un `ArrayList` de destinatari (string-uri), un `ArrayList` de mesaje (string-uri) și un `ArrayList` ce conține referințe la instante a claselor anterioare, reprezentând fiecărui destinatar de pe poziția / din prima listă mesajul de pe punctul a doua listă) folosind principiul open-closed în raport cu adăugarea unei noi liste de destinatari. Implementarea se face încăcând principiul (Notă: se consideră că cele trei liste sunt date tot timpul și respectând principiul open-closed în raport cu adăugarea unei noi liste de destinatari). Se implementează așa cum este indicat în următoarele trepte.

b. În raport cu codul să răspundă și motivează răspunsul în 4-5 propoziții.

- a. Ce se va tipări la linia marcată cu //1?

- b. Ce se va tipări la linia marcată cu //2?

```
class Pereche {
    private int n;
    private Pereche p;
    public Pereche(int a, Pereche b) { n = a; p = b; }
    public void setN() { return n; }
    public int getN() { return n; }
    public Pereche getP() { return p; }
}
```

```
public class Main {
    public static Pereche apelata(Pereche e, Pereche f) {
        f.set(17, e.getP());
        f.set(33, new Pereche(37, null));
        f.set(19, new Pereche(29, null));
        return new Pereche(43, new Pereche(33, null));
    }
    public static void main(String[] args) {
        Pereche x = new Pereche(43, new Pereche(100, null));
        Pereche y = new Pereche(97, new Pereche(100, null));
        Pereche z = x;
        z = apelata(x, y);
        System.out.println(z.getN()); //?
        System.out.println(y.getN()); //?
    }
}
```

NOTA: Orice rezolvare care nu respectă criteriile de corectitudine sau coerență nu se va puncta.

3. (1.5p) În raport cu codul Java alăturat, răspundeți la întrebări și motivează răspunsul în 4-5 propoziții, explicând pas cu pas cum se ajunge la acea valoare.

- a. Ce se va tipări la linia marcată cu //1?
- b. Ce se va tipări la linia marcată cu //2?

```
class X {
    protected int v = 0;
    public X() { v += 10; }
    public void proc(X p) { System.out.println(91); }
}
class Y extends X {
    public Y() { v += 1; }
    public void proc(X p) { System.out.println(37); }
    public int getV() { return v; }
}
class Z extends Y {
    public Z() { v = 5; }
    public void proc(Z p) { System.out.println(14); }
}
class Main {
    public static void main(String argv[]) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        x.proc(z); //?
        System.out.println(y.getV()); //?
    }
}
```

4. (1.5p) Spuneți dacă afirmația este corectă ori incorrectă și motivează răspunsul în 4-5 propoziții.

a. În codul alăturat, la linia marcată cu //1, avem o eroare de compilare.

```
class X { public int get() {return 0;} }
class Y extends X { public int get() {return 1;} }
class Main {
    class A a = new Y();
    private X x = new A();
    public static void main(String argv[]) {
        a.get(); //?
    }
}
```

- b. În raport cu codul alăturat, o atribuire de forma "Obj<A> x = new Obj();" este corectă la compilare.

```
class A<T>
class B extends A<T>
class Obj<T> {
    private T c;
    public void set(T a) { c = a; }
}
```

NOTA: Orice rezolvare care nu respectă criteriile de corectitudine sau coerență nu se va puncta.

6. (1.5p) În raport cu codul Java alăturat, răspundeți la întrebări și motivați răspunsul în 4-5 propoziții, explicând pas cu pas cum se ajunge la acea valoare.

- a. Ce se va tipări la linia marcată cu //1 ?
- b. Ce se va tipări la linia marcată cu //2 ?

```
class Tuplu {  
    private int v;  
    private Tuplu obj;  
    public Tuplu(int a, Tuplu b) { v = a; obj = b; }  
    public void set(int a, Tuplu b) { v = a; obj = b; }  
    public int getV() {return v;}  
    public Tuplu getObj() {return obj; }  
}  
  
class Main {  
    public static Tuplu apelare(Tuplu e, Tuplu f) {  
        f.set(9, e.getObj());  
        f = e;  
        f.getObj().set(6, null);  
        return new Tuplu(4, new Tuplu(5,null));  
    }  
  
    public static void main(String[] args) {  
        Tuplu x = new Tuplu(3, new Tuplu(2, null));  
        Tuplu y = new Tuplu(1, new Tuplu(-1, null));  
        Tuplu z = y;  
        y = apelare(x, y);  
        System.out.println(z.getV());//1  
        System.out.println(z.getObj().getV());//2  
    }  
}
```

NOTĂ: Unde se cere motivatie, dacă aceasta lipseste, ori nu e la obiect, ori nu e corectă, ori conține greșeli, ori e ambiguă etc. răspunsul nu se punctează.

urmatorul: daca modulul diferenței dintre anii de exp angajatului se află in intervalul [0,1), metoda este Daca acea valoare se află in intervalul [1,2) metoda va Similar, training-urile tehnice au numar limitat de training va pune la dispozitie metoda adaugaAngaj Metoda verifica daca angajatul nu e deja in secvența intampla nimic. Altfel, daca mai este loc, se adauga metoda arunca o exceptie verificata LimitaAtinsa. In primele anii recomandati de experienta, se specifica si o secvență ce contine elemente de tip String reprezentand tehnologii. Pentru acest fel de training, rata de succes va fi deosebit de mare, deoarece majoritatea angajatilor vor fi in intervalul [0,100].

Persoana responsabila cu managementul training-urilor va avea nevoie de resurse umane. Un astfel de obiect va memora interacțiunile respectiv de training-uri, ambele setate la crearea unei distribuireAngajati prin care se adauga la fiecare training un nou obiect astfel: se adauga pe rand angajatii a căror numar de inscrieri este mai mic sau egal cu 100. Apoi se adauga angajatii a căror numar de inscrieri este mai mare decât 100, pana cand se adauga si cei cu numar de inscrieri cel mai mare (adică punct de fiecare data). Daca la un moment dat nu mai există locuri disponibile (ramanând inscriși angajatii care au fost deja înregistrati), se adaugă un număr de inscrieri efectuate cu succes.

Sa se implementeze in Java clasele descrise, implementarii functionalitatii descrise.

5. (1.5p) În raport cu codul Java alăturat:

- a. Spuneți ce se va tipări la linia marcată cu //1 și **motivați răspunsul în 4-5 propozitii**, explicând pas cu pas cum se ajunge la acea valoare.

- b. Spuneți dacă afirmația care urmează e corectă ori incorrectă și **motivați răspunsul în 4-5 propozitii**: "Dacă la linia marcată cu //2 am șterge clauza throws, codul nu ar mai fi compilabil"

```
class Ex extends Exception {  
    public static void main(String args[]) throws Ex { //2  
        int i = 0;  
        int z = 0;  
        while(i < 2){  
            try{  
                int k = 0;  
                while(k < 1){  
                    k++;  
                    z = z + 2;  
                    if(i == 0) throw new Ex();  
                }  
                z = z + 2;  
            } catch(Ex e){  
                System.out.println(e);  
            } finally{  
                z = z + 1;  
            }  
            i++;  
        }  
        System.out.println(z);  
    }  
}
```

6. (1.5p) În raport cu codul Java alăturat, răspundeți la întrebări și **motivați răspunsul în 4-5 propozitii**

- a. Clasa A nu poate fi extinsă de nici o clasă din afara pachetului *numePachet*, o astfel de încercare provoacă eroare de compilare

```
package numePachet;  
public class C {  
    public static A create()  
    }  
}
```

- b. Nu există nici o posibilitate de a referi la runtime obiecte ale clasei B în exteriorul pachetului *numePachet*

```
package numePachet;  
public class A {  
    A() {}  
}  
class B extends A {  
    B() {}  
}
```

NOTĂ: Unde se cere motivație, dacă aceasta lipsește, ori nu e la obiect, ori nu e corectă ori nu e

Question 3

Not yet
answered

Marked out of
3.00

Flag question

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

```
abstract class A {
    public int proc(A p) { return 74; }
}

class B extends A {
    public int proc(A p) { return 33; }
}

class C extends A {
    public int proc(C p) { return 71; }
}

public class Main {
    public static void main(String argv[]) {
        C x = new C();
        A y = new B();
        C z = new C();
        System.out.println(y.proc(x) + z.proc(x)); /*
    }
}
```

33 +
71

Answer: 104



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

```

class Pair {
    private int v;
    private static int t[];
    public Pair(int x, int[] y) { v = x; t = y; }
    public void setVT(int x, int[] y) { v = x; t = y; }
    public void setVIT(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.setVT(17, q.getT());
        p = q;
        p.setVIT(72, 1, 21);
        int tz[] = {89, 41};
        return new Pair(94, tz);
    }
    public static void main(String[] args) {
        int ta[] = {0, 13};
        int tb[] = {45, 99};
        Pair a = new Pair(66, ta);
        Pair b = new Pair(55, tb);
        Pair c = b;
        b = call(a, c);
        System.out.println(b.getT()[0]);
    }
}

```

(45) 99

Diagrama de date și urmăriri:

- Pair Class:** Acesta conține un membru privat `v` și un membru static privat `t`.
- Constructor:** `Pair(int x, int[] y)` setează `v = x` și `t = y`.
- Metode:**
 - `setVT(int x, int[] y)` setează `v = x` și `t = y`.
 - `setVIT(int x, int i, int y)` setează `v = x` și `t[i] = y`.
 - `getV()` returnează `v`.
 - `getT()` returnează `t`.
- Main Class:** Conține o metodă statică `call(Pair p, Pair q)`.
 - În interiorul acesteia, se apelă `p.setVT(17, q.getT())`. În urma acestei apeluri, `p` va avea `v = 17` și `t = {0, 13}`.
 - Apoi, se apelă `p = q`, astfel că `p` va avea `v = 55` și `t = {45, 99}`.
 - Se apelă `p.setVIT(72, 1, 21)`, astfel că `t` va fi modificată la `{72, 45, 21}`.
 - În final, se returnează un nou obiect `Pair(94, tz)`, unde `tz = {89, 41}`.
- Metoda main:** Crea obiectele `a` și `b` și le aplice metoda `call`. Apoi, se printează primul element din `t` al obiectului `b`.

Urmariri:

- Obiectul a:** `{66, 0, 13}`
- Obiectul b:** `{55, 45, 21}`
- Obiectul c:** `{72, 45, 21}`
- Obiectul returnat:** `{94, 89, 41}`

Answer:

45

Not yet
answered

Marked out of
3.00

Flag question

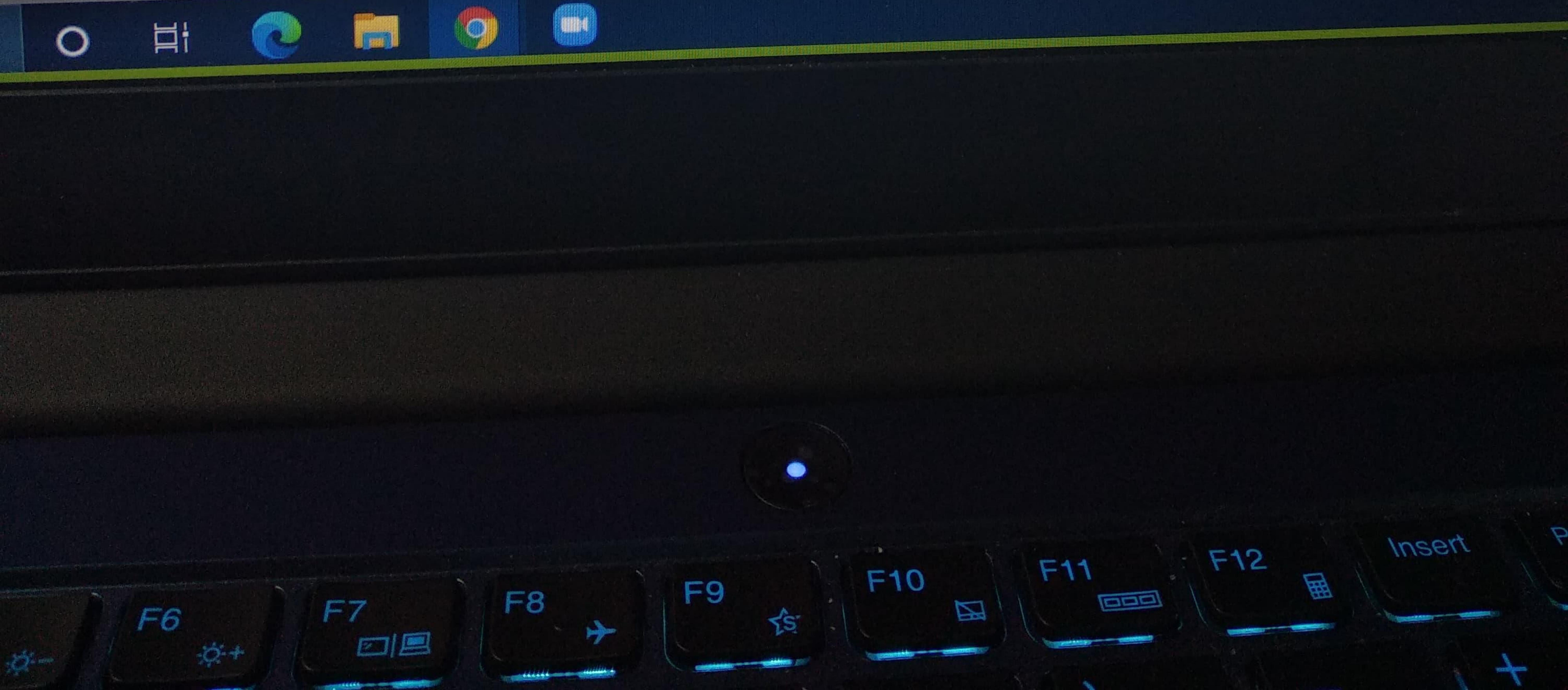
Ce valoare întreagă se va tipări la linia marcată cu /* în urma executiei programului?

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

Handwritten annotations:

- l Z K
- 0 C O
- 2 1
- 1 3
- 0
- 1
- 5
- 7 2
- 9 3
- 1 1
- 2 1 2

Answer:



Obiecte

Question 4

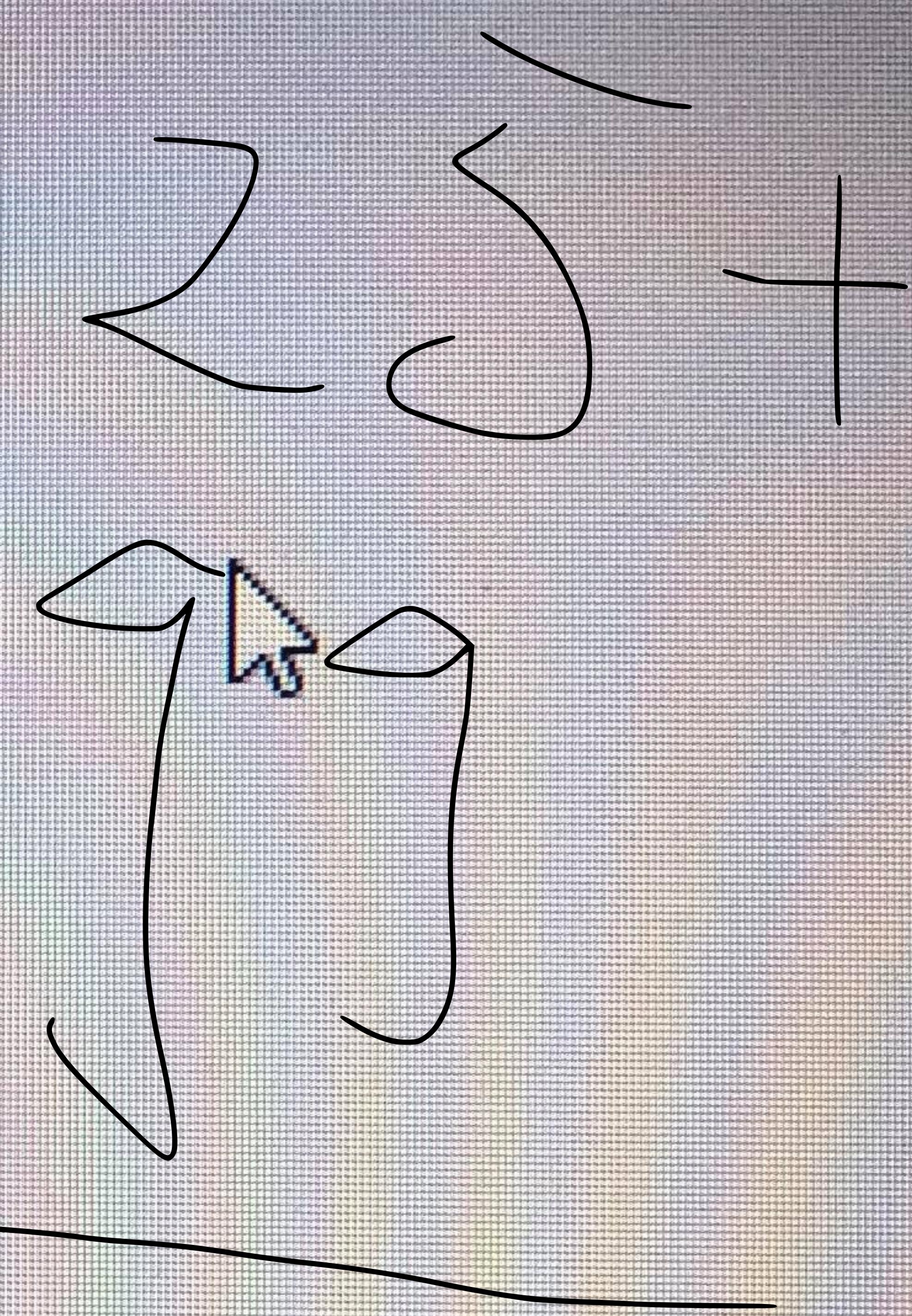
Not yet
answered

Marked out of
3.00

Flag question

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

```
abstract class A {  
    public int proc(B p) { return 25; }  
}  
  
class B extends A {  
    public int proc(C p) { return 96; }  
}  
  
class C extends B {  
    public int proc(C p) { return 99; }  
}  
  
public class Main {  
    public static void main(String argv[]) {  
        A x = new C();  
        C y = new C();  
        C z = new C();  
        System.out.println(x.proc(z) + y.proc(z)); /*  
    }  
}
```



Answer: 124

Previous page

Question 3

Not yet
answered

Marked out of
3.00

Flag question

Ce valoare întreagă se va tipări la linia marcată cu ///* în urma execuției pro-

```
class ExA extends Exception {}  
class ExB extends ExA {}  
  
class Main {  
  
    public static int make(int a) throws ExA {  
        for (int k = 0; k < 3; k++) {  
            a = a + 2;  
            if (a == 3) throw new ExB();  
        }  
        return a;  
    }  
  
    public static void main(String argv[]) throws ExA {  
        int x = 0;  
        for (int i = 0; i < 2; i++) {  
            try {  
                x++;  
                x = make(x);  
                x++;  
            } catch (ExB e) {  
                System.out.println(e);  
            } finally {  
                x++;  
            }  
        }  
        System.out.println(x); ///*
```

Answer

Not yet
answered

Marked out of
3.00

Flag question

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

```
class Pair {  
    private int v;  
    private int t[];  
  
    public Pair(int x, int [] y) { v = x; t = y; }  
    public void setVT(int x, int[] y) { v = x; t = y; }  
    public void setVIT(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) {  
        q.setVT(100, p.getT());  
        q = p;  
        q.setVIT(53, 1, 34);  
        int tz[] = {88, 81};  
        return new Pair(20, tz);  
    }  
  
    public static void main(String[] args) {  
        int ta[] = {71, 59};  
        int tb[] = {11, 61};  
        Pair a = new Pair(19, ta);  
        Pair b = new Pair(83, tb);  
        Pair c = b;  
        b = call(a, b);  
        System.out.println(c.getT()[1]); /*  
    }  
}
```

→

34

53, 71, 34
a { 19, 71, 59 }
b { 83, 71, 59 }
c { 100, 71, 59 }
100, 71, 59
(34)
b { 20, 88, 81 }



GAMING



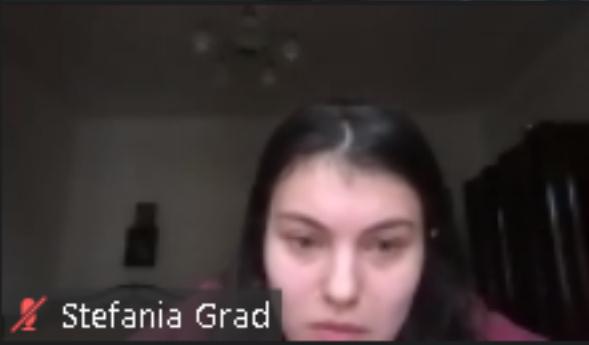
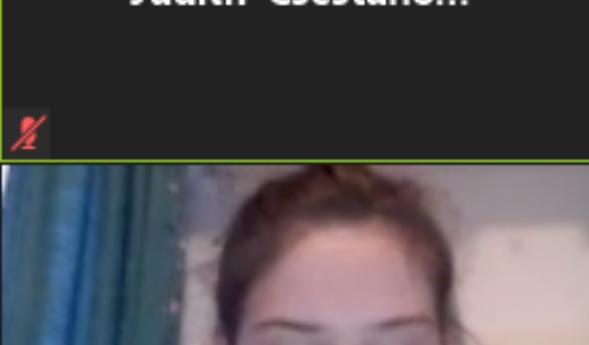
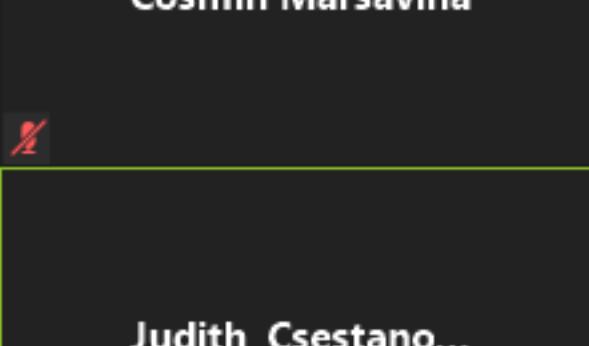
Programarea Orientata pe Obiecte

Quiz navigation

[Finish attempt ...](#)Time left **0:07:49****Question 4**Not yet
answeredMarked out of
3.00[Flag question](#)

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

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

Answer: **Cosmin Marsavina****Judith Cestano...**

1. (1.5p) Se dau clasele care urmează:

```
class Sms {
    public void sendSMS(String dest, String msg) {
        class Fax {
            public void sendFax(String dest, String msg) {
                // ...
            }
        }
    }
}
```

- a. Fără a modifica în vreun fel cele două clase, implementați la instante a claselor de mesaje (string-uri), un ArrayList de mesaje (parametru din prima listă) și un ArrayList reprezentând modalității de transmitere / (din a treia listă). Metoda transmite în report că cele trei liste sunt date tot împreună se face încălcând modalitatea de transmisie de pe pozitia a altă modificare a lui punctul a) dar respectând principiul open-closed să continuăm să există nevoie de transmitere a mesajelor (Notă: se va adăugarea a treia listă în report cu codul să fie să cum se ajunge la marcată cu //1 ?)

```
2. (1.5p) În report cu codul întrebări și motivează răspunsul la propoziții după cum se ajunge la marcată cu //1 ?
```

b. Ce se va lipări la linia menționată cu //2 ?

```
class Pereche {
    private int n;
    public Pereche p;
    public void set(int a, Pereche b) {n = a, p = b;}
    public int getN() {return n;}
    public Pereche getP() {return p;}
}
public class Main {
    t.set(17, b.getP());
    t = e;
    t.set(33, new Pereche(pereche.e, Pereche.e));
    return new Pereche(19, new Pereche());
}
public static void main(String[] args) {
    Pereche x = new Pereche(43, new Pereche(83, null));
    Pereche y = new Pereche(43, new Pereche(100, null));
    x = apelare(x, y);
    System.out.println(x.getN()); //1
    System.out.println(y.getN()); //2
}
```

NOTĂ: Vedeți ce este rezultatul de la scosă îl scrieți, ori nu e la obiect, ori nu e corectă, ori conține greșeli, ori e ambiguă etc. răspunsul nu se punctează

3. (1.5p) În raport cu codul Java alăturat cu pasul în 4-5 propoziții și motivati răspunsul la întrebări și motivează răspunsul în 4-5

```
a. Ce se va lipări la acea valoare.
```

b. Ce se va lipări la linia marcată cu //1 ?

c. Ce se va lipări la linia marcată cu //2 ?

```
class X {
    protected int v = 0;
    public X() {v += 10;}
}
class Y extends X {
    public void proc(X p) {System.out.println(91);}
}
class Z extends Y {
    public void proc(Y p) {System.out.println(92);}
}
class Main {
    X x = new X();
    Y y = new Y();
    Z z = new Z();
    x.proc(z); //1
    System.out.println(y.proc()); //2
}
```

a. În codul alăturat, la linia marcată cu //1, avem o eroare de compilare

```
class X {
    public int get() {return 0;}
}
class Main {
    public X a = new Y();
    a.get(); //1
}
class A {}
class B extends A {}
private T c;
public void set(T a) {c = a;}
```


Question 1

Not yet
answered

Marked out of
3.00

Flag question

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

```
class X {
    protected static int v = 0;
    public X() { v += 9; }
}
class Y extends X {
}
class Z extends Y {
    public Z() { v += 7; }
    public static int getV() { return v; }
}
class Main {
    public static void main(String argv[]) {
        X x = new Y();
        X y = new Z();
        Z z = new Z();
        System.out.println(Z.getV()); /* ←
    }
}
```

9
9
7
9
7
—
4 1

Answer:



DELL

ENG 9:49
ROP 1/2

Finish attempt ...

Time left 0:07:38

Not yet
answered
Marked out of
3.00
Flag question

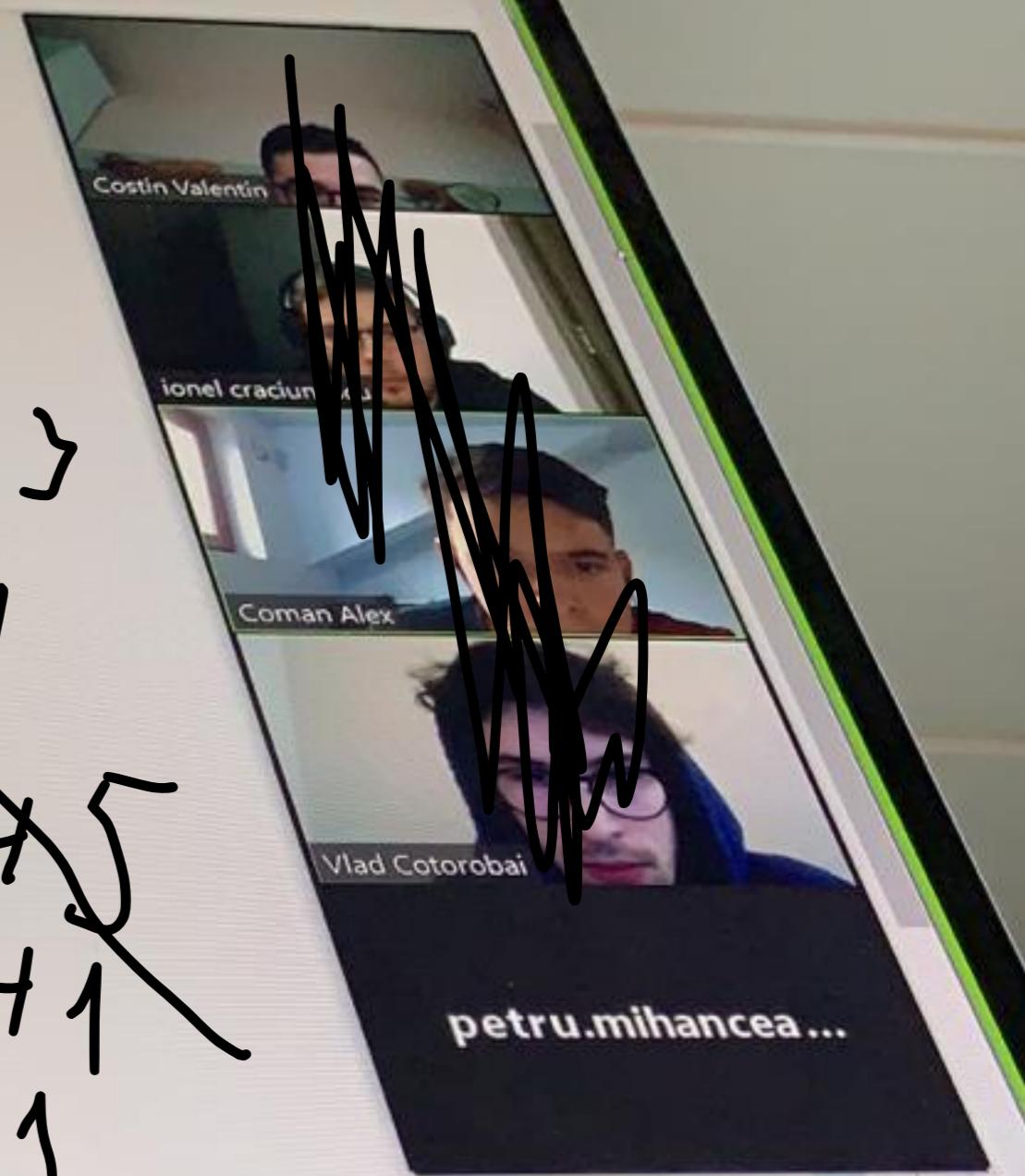
Ce valoare întreagă se va tipări la linia marcată cu /* în urma executiei programului de mai jos?

```
class Pair {  
    private int v;  
    private int t[];  
    public Pair(int x, int [] y) { v = x; t = y; }  
    public void setVT(int x, int[] y) { v = x; t = y; }  
    public void setVIT(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) {  
        q.setVT(48, p.getT());  
        q = p;  
        q.setVIT(87, 1, 41);  
        int tz[] = {93, 81};  
        return new Pair(38, tz);  
    }  
    public static void main(String[] args) {  
        int ta[] = {39, 45};  
        int tb[] = {11, 76};  
        Pair a = new Pair(55, ta);  
        Pair b = new Pair(35, tb);  
        Pair c = b;  
        b = call(a, b);  
        System.out.println(c.getV()); /*  
    }  
}
```

48



87, 39 41
n → a {55, 39, 45 }
b {35, 11, 76 }
c → 48, 39, 45
38, 93, 81 41



h ENG 9:51 AM
ROP 1/29/2021

DELL

Programarea Orientata pe Obiecte

Quiz navigation



Finish attempt ...

Time left 0:04:04

Question 4
Not yet answered
Marked out of 3.00
Flag question

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

```
abstract class A {
    public int proc(A p) { return 98; }
}
class B extends A {
    public int proc(A p) { return 17; }
}
class C extends A {
    public int proc(C p) { return 65; }
}
public class Main {
    public static void main(String argv[]) {
        C x = new C();
        A y = new B();
        C z = new C();
        System.out.println(y.proc(x) + z.proc(x)); /*
    }
}
```

17
65

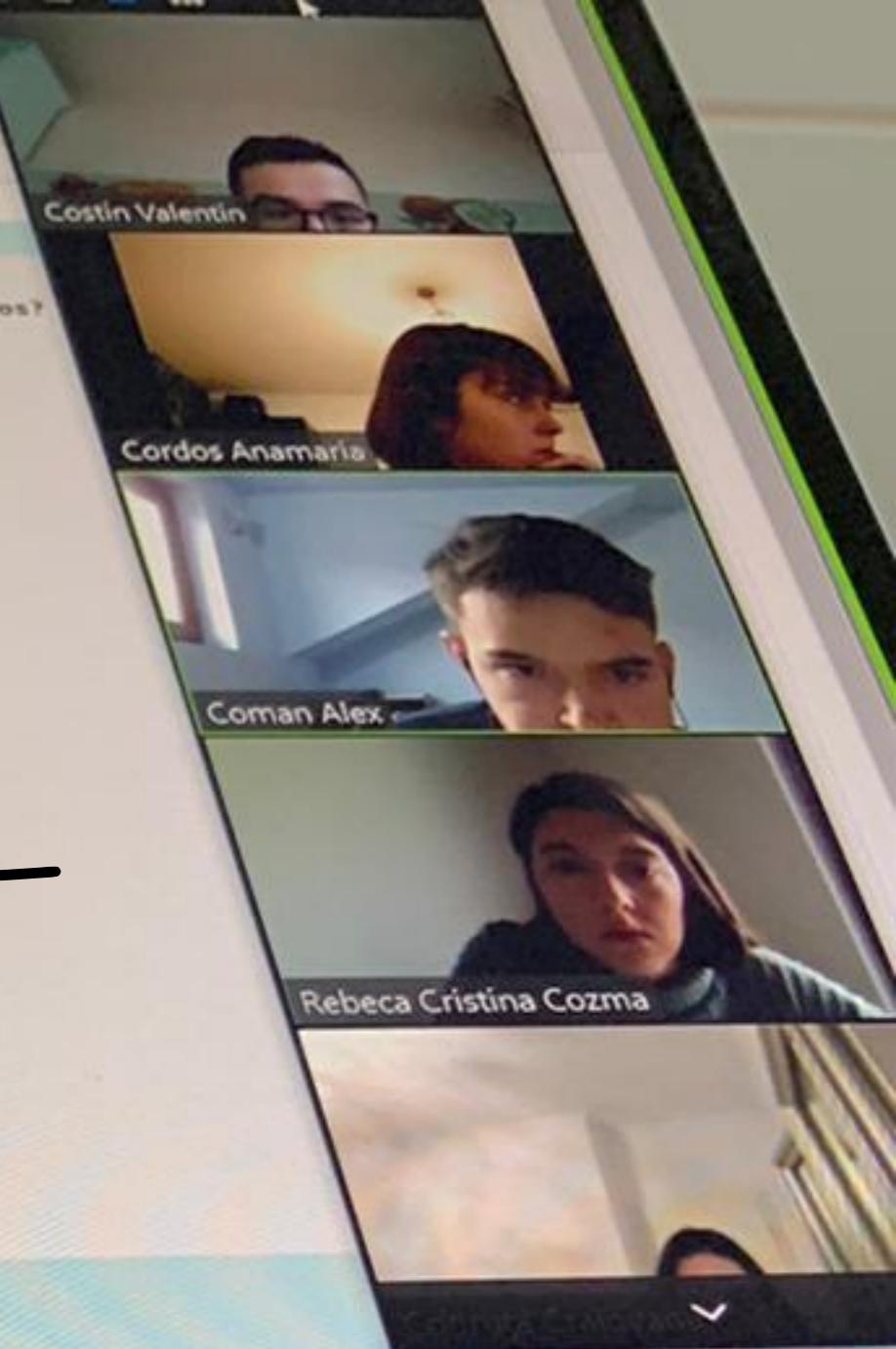
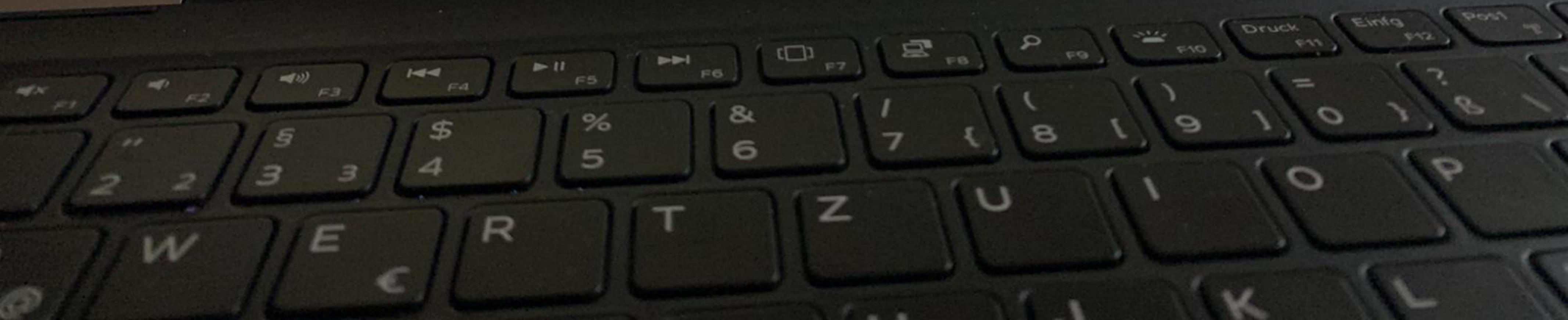
Answer:



here to search



DELL



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

```
class X {  
    protected int v = 0;  
    public X() { v += 87; }  
}  
  
class Y extends X {  
    public Y() { v += 94; }  
    public int getV() { return v; }  
}  
  
class Z extends Y {  
    public Z() { v += 46; }  
}  
  
class Main {  
    public static void main(String argv[]) {  
        X x = new Z();  
        Y y = new Y();  
        Z z = new Z();  
        System.out.println(y.getV()); /*  
    }  
}
```

87 +
94

181

Answer:

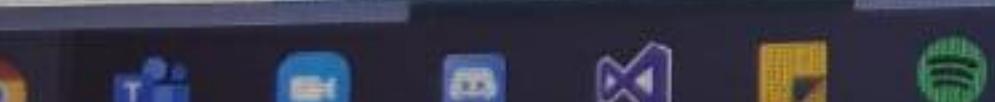
ASUS VivoBook



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

```
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 setVI(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(77, q.getT());
        p = q;
        p.setV(46, 1, 58);
        int tz[] = {80, 94};
        return new Pair(50, tz);
    }
    public static void main(String[] args) {
        int ta[] = {16, 55};
        int tb[] = {76, 54};
        Pair a = new Pair(86, ta);
        Pair b = new Pair(24, tb);
        Pair c = b;
        b = call(a, c);
        System.out.println(a.getT()[1]); /*
    }
}
```

Download on the



ASUS VivoBook



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

```
abstract class A {
    public int proc(A p) { return 56; }
}

class B extends A {
    public int proc(A p) { return 73; }
}

class C extends B {
    public int proc(C p) { return 61; }
}

public class Main {
    public static void main(String argv[]) {
        A x = new C();
        B y = new B();
        C z = new C();
        System.out.println(x.proc(z) + y.proc(z)); /* _____
    }
}
```

+
66
73

J G +
7 3

Answer:

[Previous page](#)

