

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;

}

}

public 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);

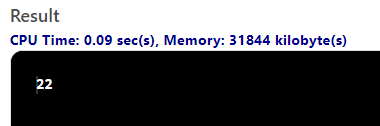
A b = new A(7, 28);

m(a, b);

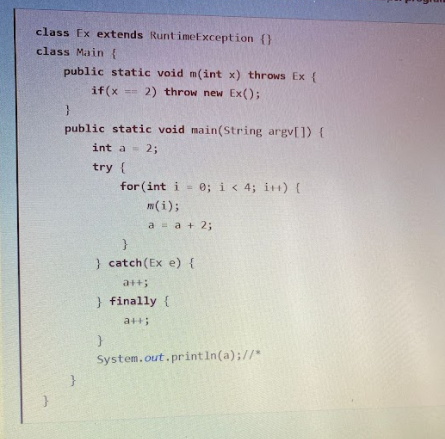
System.out.println(a.getY() + b.getX());

}

}



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



class Ex extends RuntimeException{}

public class Main {

public static void m(int x) throws Ex{

if(x==2) throw new Ex();

}

public static void main(String args[]) {

int a = 2;

try{

for(int i=0; i<4; i++){

m(i);

a = a + 2;

}

} catch (Ex e){

a++;

} finally {

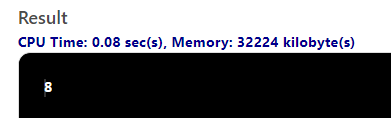
a++;

}

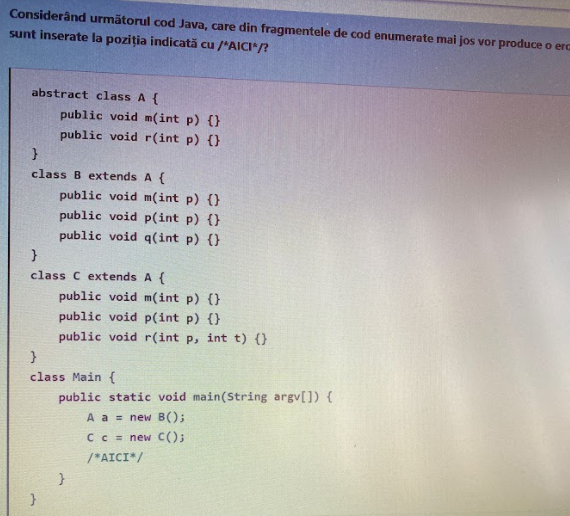
System.out.println(a);

}

}



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



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) {}

}

public class Main {

public static void main(String args[]) {

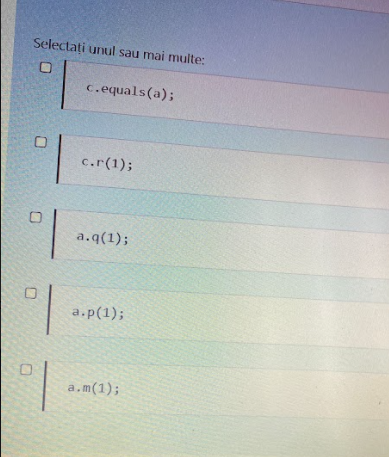
A a = new B();

C c = new C();

//AICI

}

}



c.equals(a); -> compileaza

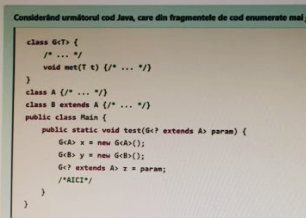
c.r(1); -> compileaza

a.q(1); -> EROARE

a.p(1); -> EROARE

a.m(1); -> compileaza

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



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>();

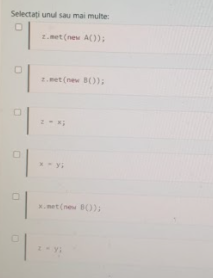
G<? extends A> z = param;

//AICI

}

public static void main(String args[]) {}

}



z.met(new A()); -> EROARE

z.met(new B()); -> EROARE

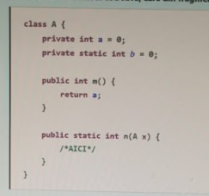
z = x; -> compileaza

x = y; -> EROARE

x.met(new B()); -> compileaza

z = y; -> compileaza

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



class A{

private int a = 0;

private static int b = 0;

public int m(){

return a;

}

public static int m(A x){

//AICI

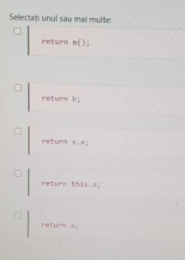
}

}

public class Main {

public static void main(String args[]) {}

}



return m(); -> EROARE

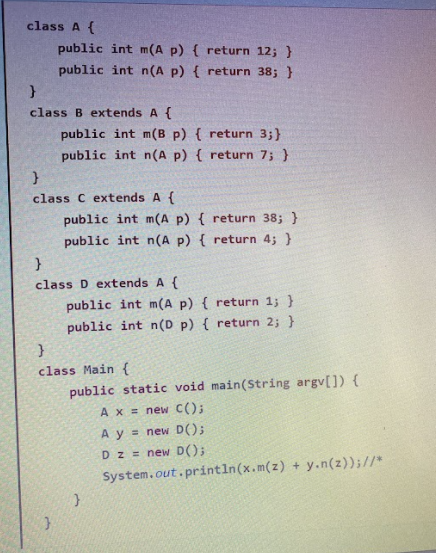
return b; -> compileaza

return x.a; -> compileaza

return this.a; -> EROARE

return a; -> EROARE

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



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;}

}

public class Main {

public static void main(String args[]) {

A x = new C();

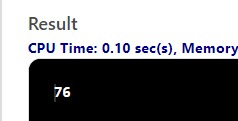
A y = new D();

D z = new D();

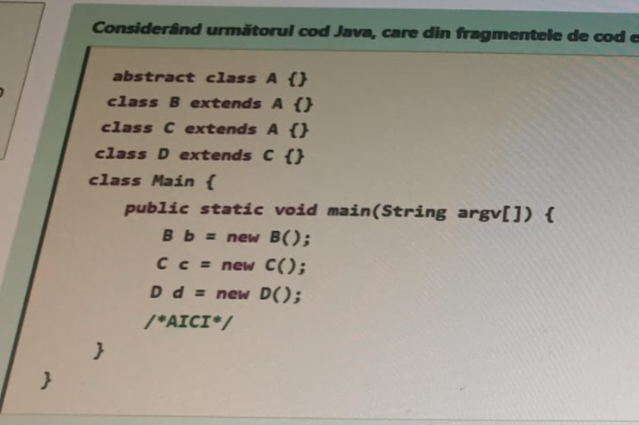
System.out.println(x.m(z) + y.n(z));

}

}



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



abstract class A {}

class B extends A {}

class C extends A{}

class D extends C {}

public class MyClass {

public static void main(String args[]) {

B b = new B();

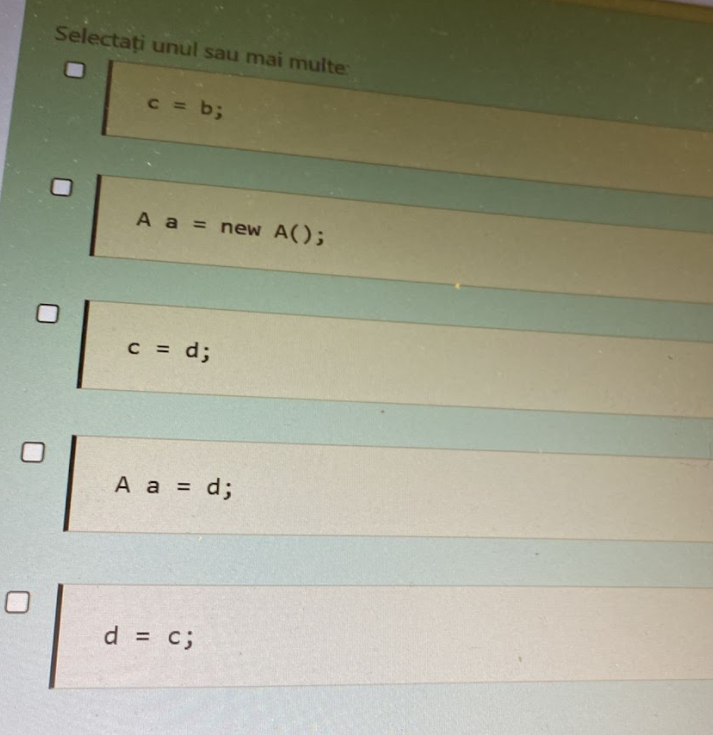
C c = new C();

D d = new D();

//AICI

}

}



c = b; -> EROARE

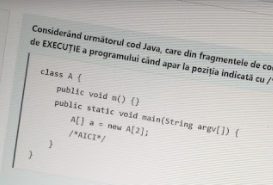
A a = new A(); -> EROARE

c = d; -> compileaza

A a = d; -> compileaza

d = c; -> EROARE

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



public class A {

public void m() {}

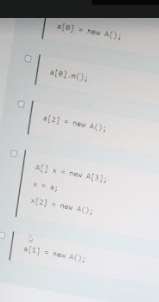
public static void main(String args[]) {

A[] a = new A[2];

//AICI

}

}



a[0] = new A(); -> compileaza

a[0].m(); -> EROARE

a[2] = new A(); -> EROARE

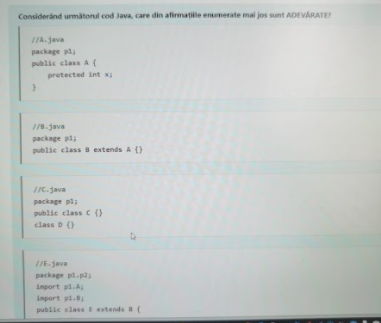
A[] x = new A[3];

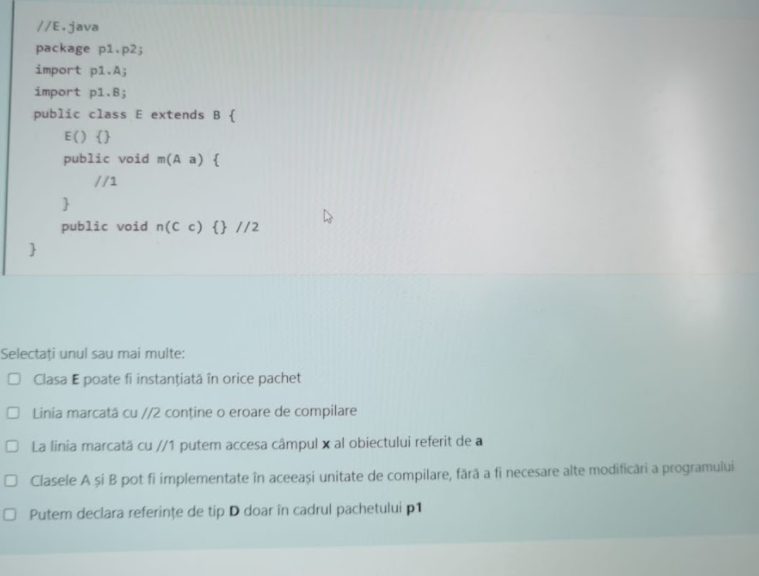
x = a;

x[2] = new A(); -> EROARE

a[1] = new A(); -> compileaza

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_





//A.java

package p1;

public class A{

protected int x;

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

//B.java

package p1;

public class B extends A{}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

//C

package p1;

public class C {}

class D {}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

//E.java

package p1.p2;

import p1.A;

import p1.B;

public class E extends B{

E() {}

public void m(A a) {

//1

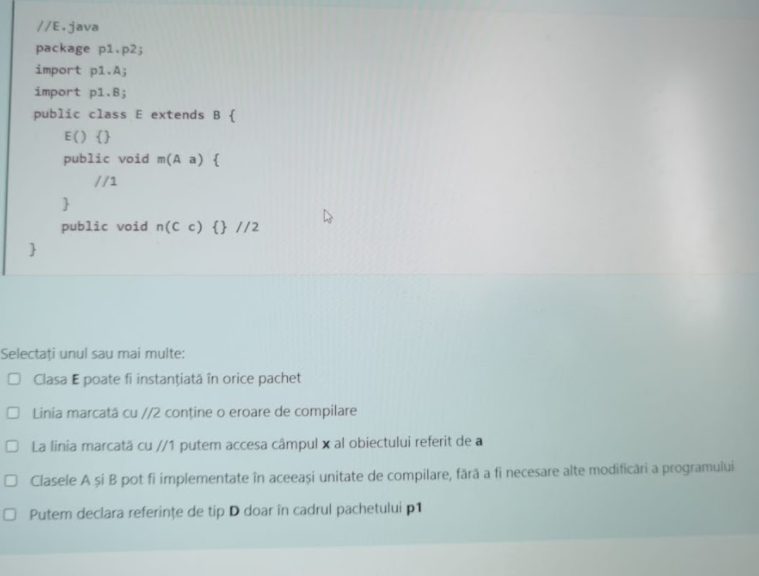
}

public void n(C c) {

//2

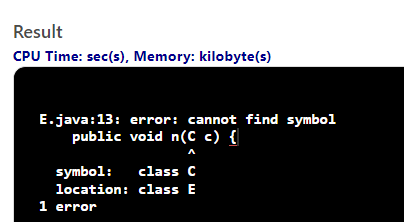
}

}



*Clasa E poate fi instantiata in orice pachet* -> FALS, E este de tip package => E poate fi instantiata doar in pachetul p1.p2;

*Linia marcata cu //2 contine o eroare de compilare* -> ADEVARAT;



*La linia marcata cu //1 putem accesa campul x al obiectului referit de a* -> FALS, se poate accesa doar cu this.a, nu si cu x.a;

*Clasele A si B pot fi implementate in aceeasi unitate de compilare, fara a fi necesare alte modificari a programului* ->FALS, A si B sunt amandoua public, iar intr-un fisier/unitate de compilare poate exista o singura functie publica!

*Putem declara referinte de tip D doar in cadrul pachetului p1* -> ADEVARAT