

//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

package p2;

import p1.A;

public class C {

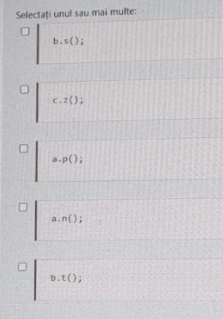
private void z() {}

public void v(A a, B b, C c){

//AICI

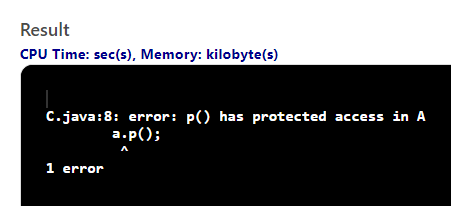
}

}

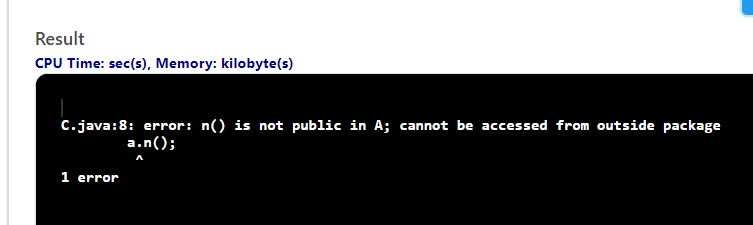


b.s(); -> compileaza

c.z(); -> compileaza

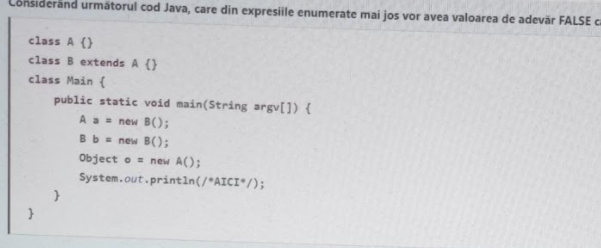
a.p(); -> EROARE

a.n(); -> EROARE



b.t(); -> compileaza

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



class A {}

class B extends A {}

public class Main {

public static void main(String args[]) {

A a = new B();

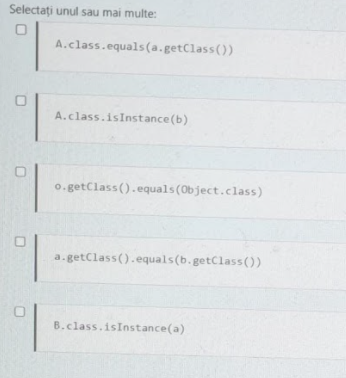
B b = new B();

Object o = new A();

System.out.println(/\*AICI\*/);

}

}



A.class.equals(a.getClass()) -> FALSE

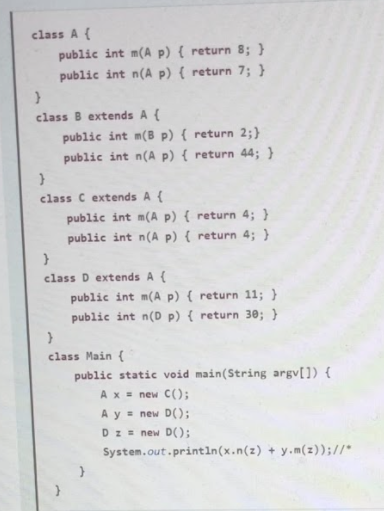
A.class.isInstance(b) -> TRUE

o.getClass().equals(Object.class) -> FALSE

a.getClass().equals(b.getClass()) -> TRUE

B.class.isInstance(a) -> TRUE

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



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;}

}

public class Main {

public static void main(String args[]) {

A x = new C();

A y = new D();

D z = new D();

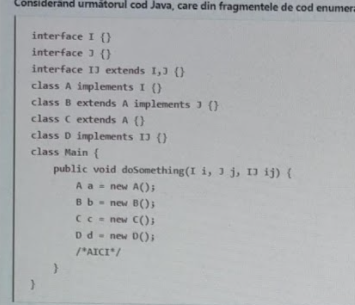
System.out.println(x.n(z) + y.m(z));

}

}



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



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 {}

public class Main {

public void doSomething(I i, J j, IJ ij){

A a = new A();

B b = new B();

C c = new C();

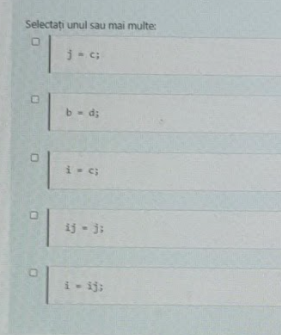
D d = new D();

//AICI

}

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

}



j = c; -> EROARE

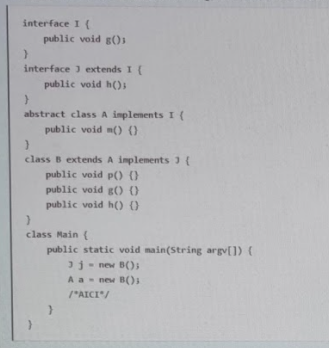
b = d; -> EROARE

i = c; -> compileaza

ij = j; -> EROARE

i = ij; -> compileaza

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



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() {}

}

public class Main {

public static void main(String args[]) {

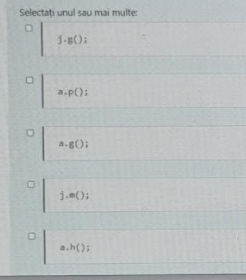
J j = new B();

A a = new B();

//AICI

}

}



j.g(); -> compileaza

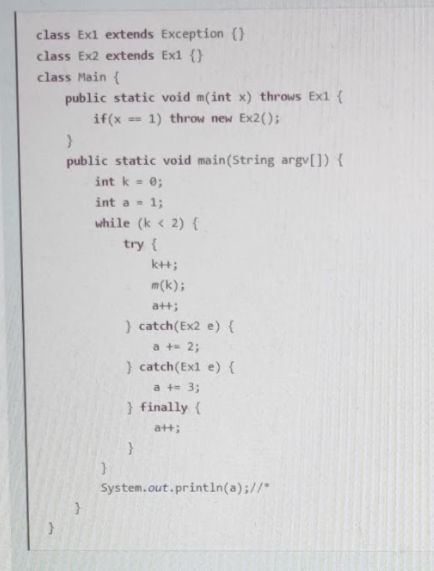
a.p(); -> EROARE

a.g(); -> compileaza

j.m(); -> EROARE

a.h(); -> EROARE

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



class Ex1 extends Exception {}

class Ex2 extends Ex1 {}

public class Main {

public static void m(int x) throws Ex1{

if(x==1) throw new Ex2();

}

public static void main(String args[]) {

int a = 1;

int k = 0;

while(k<2){

try{

k++;

m(k);

a++;

} catch (Ex2 e){

a += 2;

} catch (Ex1 e){

a += 3;

} finally {

a++;

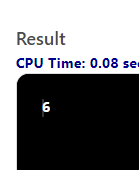
}

}

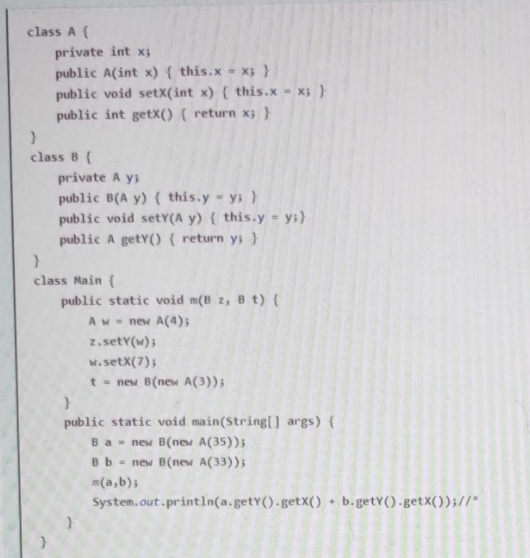
System.out.println(a);

}

}



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



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 A y;

public B(A y){

this.y = y;

}

public void setY(A y){

this.y = y;

}

public A getY(){

return y;

}

}

public 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));

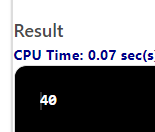
B b = new B(new A(33));

m(a, b);

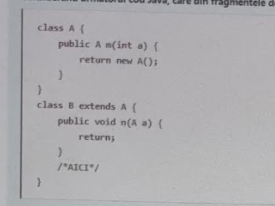
System.out.println(a.getY().getX() + b.getY().getX());

}

}



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



class A{

public A m(int a){

return new A();

}

}

class B extends A{

public void n(A a){

return;

}

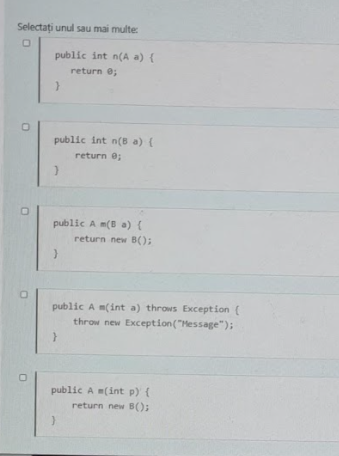
//AICI

}

public class Main {

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

}



public int n(A a){

return 0;

} -> EROARE

public int n(B a){

return 0;

} -> compileaza

public A m(B a){

return new B();

} -> compileaza

public A m(int a) throws Exception{

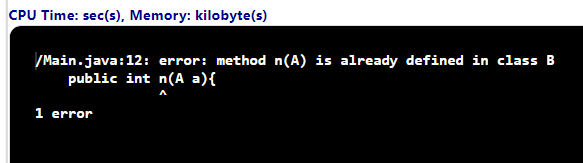
throw new Exception(“Message”);

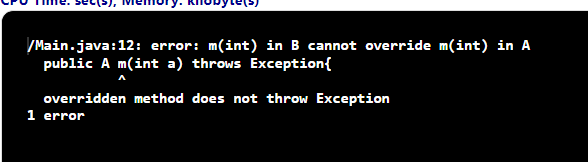
} -> EROARE

public A m(int p){

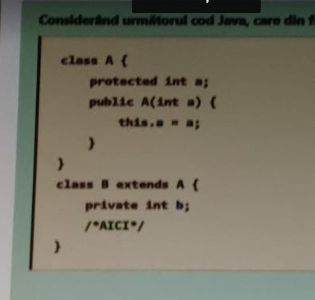
return new B();

} -> compileaza





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



class A{

protected int a;

public A(int a){

this.a = a;

}

}

class B extends A{

private int b;

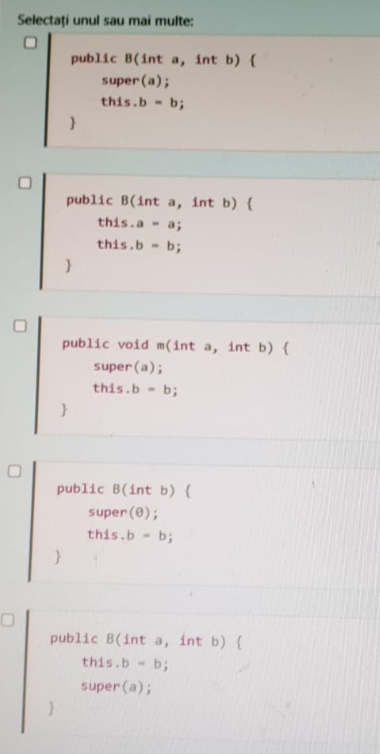
//AICI

}

public class Main {

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

}



public B(int a, int b){

super(a);

this.b = b;

} -> compileaza

public B(int a, int b){

this.a = a;

this.b = b;

} -> EROARE

public void m(int a, int b){

super(a);

this.b = b;

} -> EROARE

public B(int b){

super(0);

this.b = b;

} -> compileaza

public B(int a, int b){

this.b = b;

super(a);

} -> EROARE

