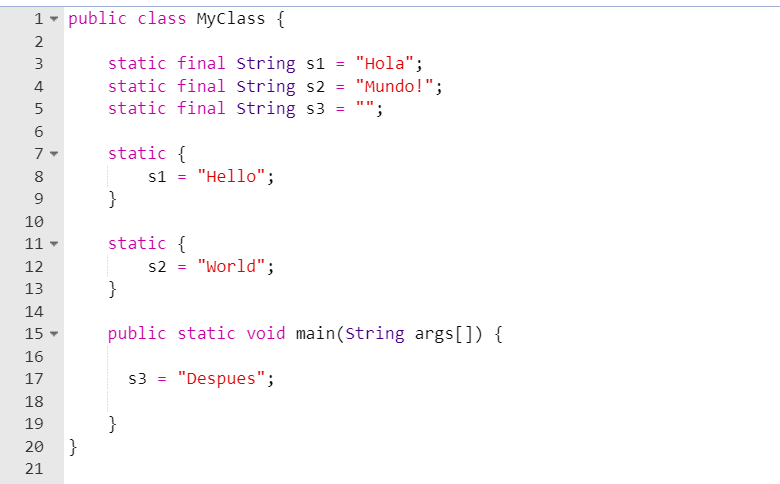
¿Cuantos errores tiene el siguiente código?



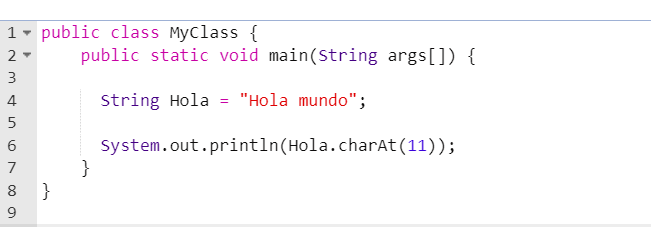
respuestas posibles:

-0

-1

-3

2)

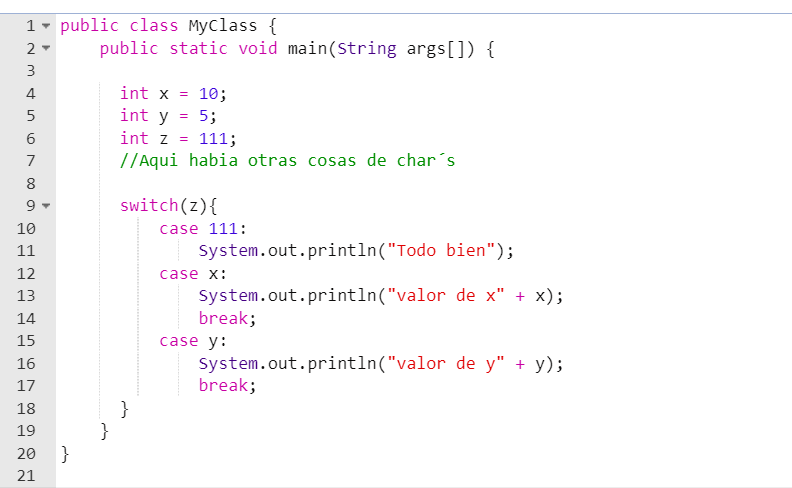




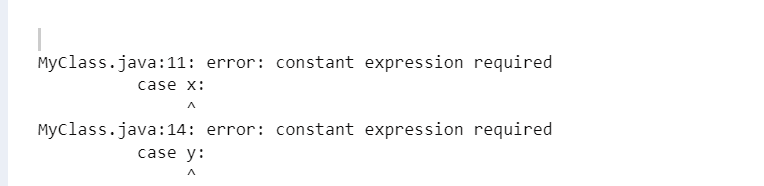
La trampa está en que también te ponen:

java.lang.StringrIndexOutOfBoundsException

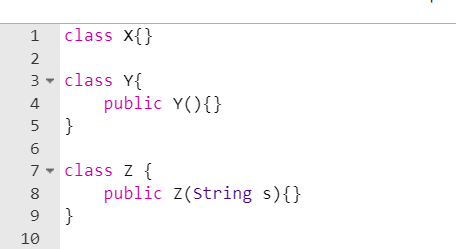
3)



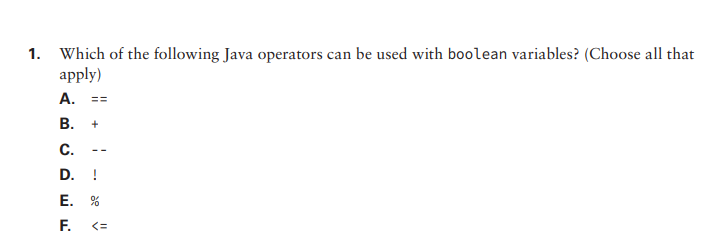
No pueden ir variables en los CASE

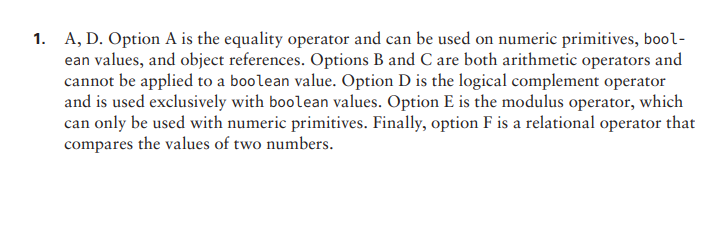


4) ¿Cuáles de las siguientes clases tienen un constructor por default?

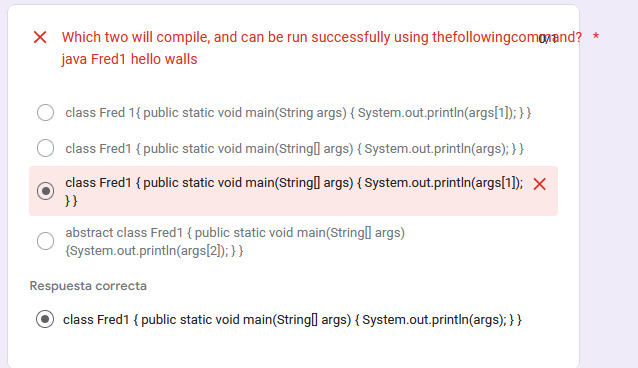


X,Y

5)  


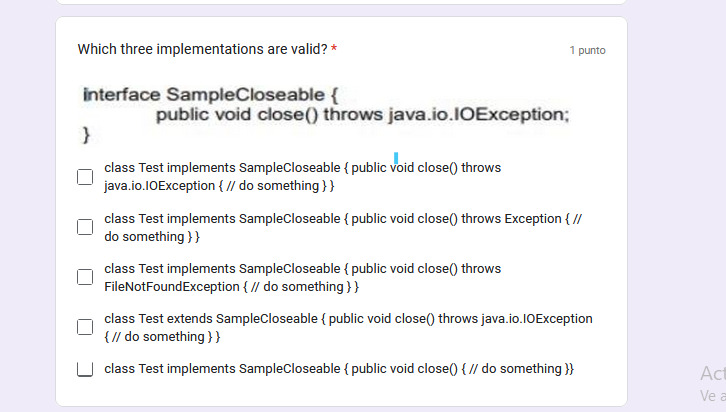


6)

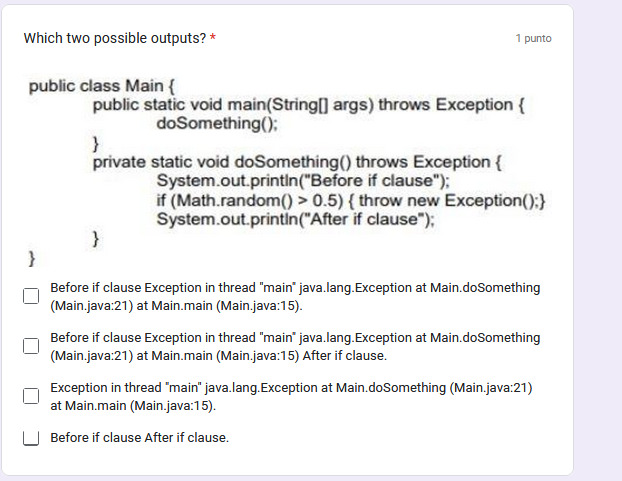


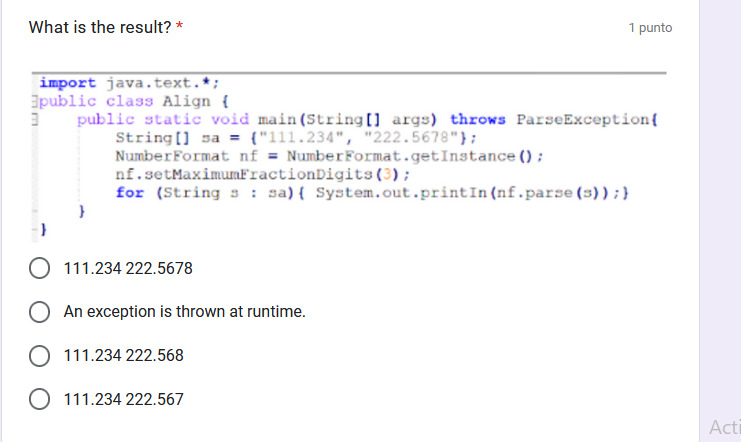
hay dos correctas

6)

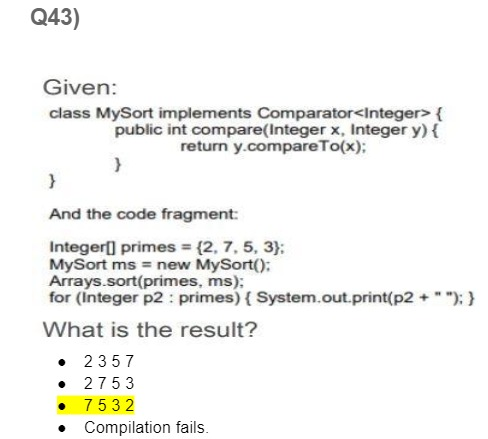


7)

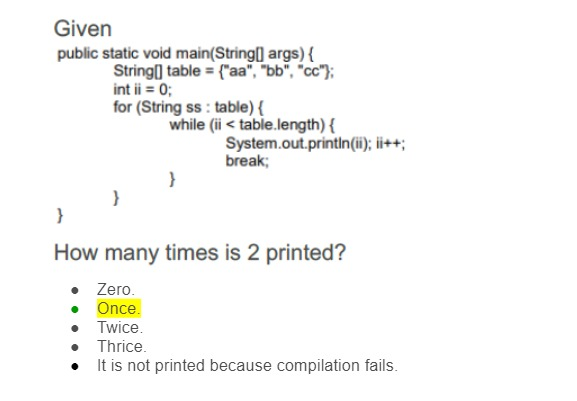


8)

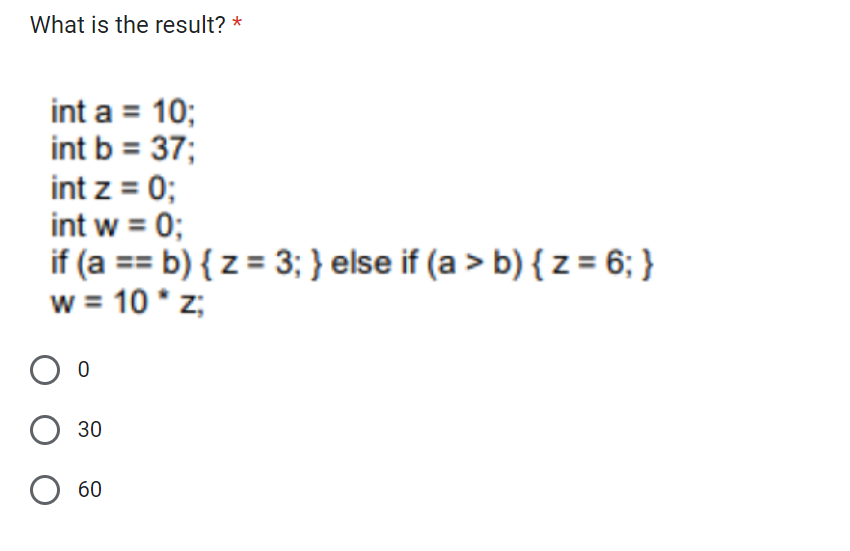
9)



10)



11)



Respuesta 0



public static void main(String[] args) {

System.out.println("Hola Mundo");

// int [] arr = new int[]; // no compila porque no se indico la longitud

//int [][] d = new int[];// no compila

int a [][] = new int[7][6];

Object [][][] objects = new Object[4][0][5];

}

public static void main(String[] args) {

int a=0;

System.out.println(a++ + 2);

System.out.println(a);

}

en el primer print a la variable “a” se le suma 2 e imprime 2 después de hacer ese proceso

a la variable “a” se incrementa en 1 entonces en el segundo print imprimer 1

Resultado 2 1

Qué se va a imprimir al ejecutar el método main()

public class Spider {

void spider(int a){

System.out.print("Spider");

}

}

public class Genero {

void genero(double b){

System.out.print("Genero");

}

}

public class SpiderMain {

public static void main(String[] args) {

Spider spider = new Spider();

Genero genero = new Genero();

spider.spider(4);

genero.genero(9.0);

}

}

Respuesta SpiderGenero

Con cuál de línea imprimere Equals

//boolean r1 = a.equals(b);

//boolean r2 = a==b;

//boolean r3 = a.equalsIgnoreCase(b);

String a = "Hola";

String b = "hola";

boolean resultado = false;

if (.........){

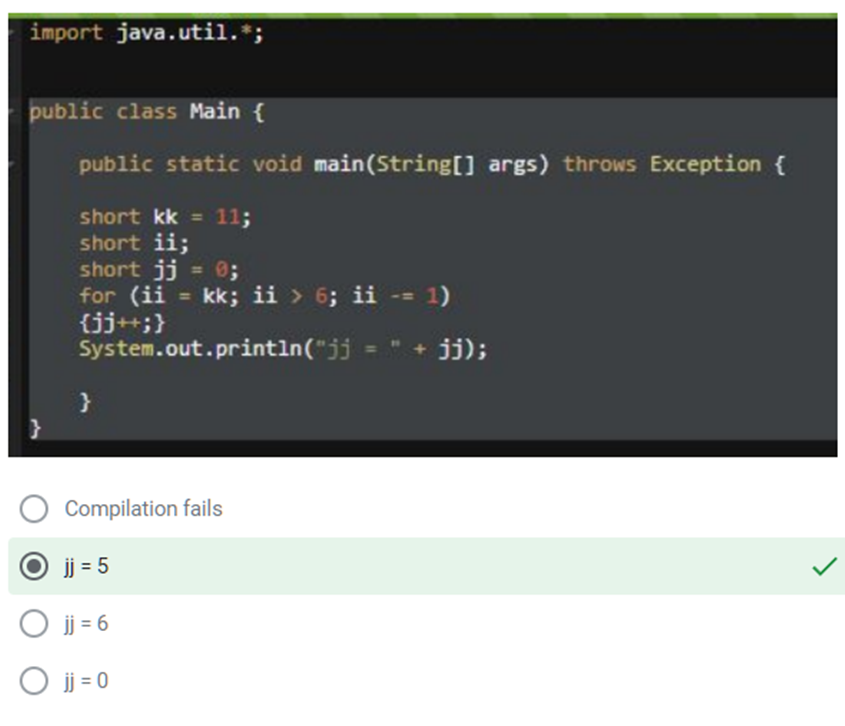
System.out.println("Equals");

}else {

System.out.println("Not Equals");

}

Respuesta boolean r3 = a.equalsIgnoreCase(b);



¿Qué se imprime al ejecutar el código?

public class EjemploArray {

public static void main(String[] args) {

List p= new ArrayList();

p.add(7);

p.add(1);

p.add(5);

p.add(1);

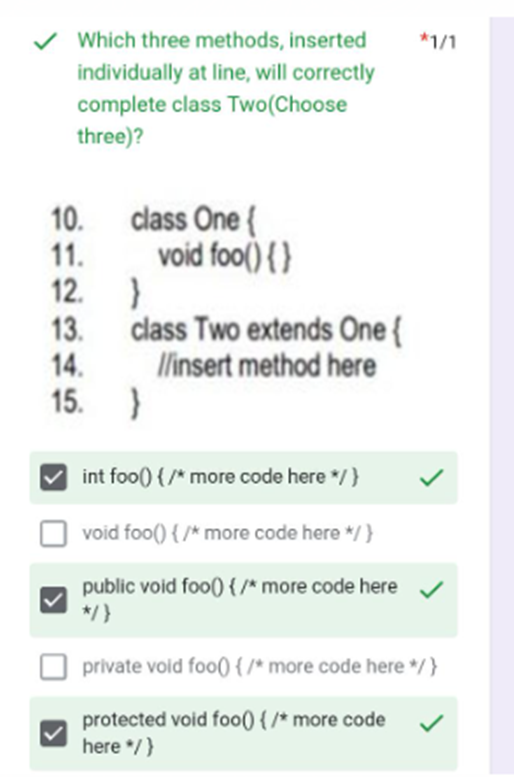
p.remove(1);

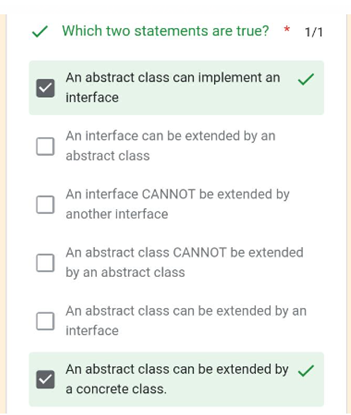
System.out.println(p);

}

}

Respuesta [7, 5, 1]





a este ejercicio le cambiaron la redacción

Varags

public void suma(int...g,int []q){} //no compila porque los varargs debe ir al final de la lista de parametros

public void div(int[], ...int r){} //no compila porque los varargs van después del tipo de dato

public int suma(int..d,float...o){

return d+o;

}//no compila porque no sabe que es la variable d de los parametros si se desea agregar varargs deben ser tres puntos\*/

public void t (int...h, double...t){}//no compila porque solo se permite una vez usar varargs

public void ad(int b,double s, String...a){}// si funciona

public class VaribaleFinal {

public class Persona extends Carro{

final String nombre;

public Persona() {

Carro carro = new Carro();

nombre = "andres";

}

}

public class Carro{

}

public static void main(String[] args) {

Persona persona = new Persona(); // error aqui

}

}

Respuesta no se puede referenciar desde un contexto static

public static void main(String[] args) {

boolean x = false;

boolean z = false;

int y= 20;

System.out.println(x);

//true false |

x = (y!=10) ^ (z!=false);

System.out.println(x + " " + y + " " + z);

}

Imprime: true 20 false