Elaborar la tabla de símbolos

```java
package UNPA.WinForm.Compiladores;

public class Varargs_Project {

    public static void main(String[] args) {
        // TODO code application logic here

        vaTest(10); //1 arg
        vaTest(1,2,3); //3arg
        vaTest(); //sin arg
    }

    //vaTest() usa un vararg
    static void vaTest(int ... v){
        System.out.println("Número de args: "+v.length);
        System.out.println("Contiene: ");
        for (int i=0; i<v.length;i++)
            System.out.println(" arg "+i+": "+v[i]);

        System.out.println();
    }
}
```

S -> <PKG> <CLASS>
<PKG> -> <PALRE> <IDE> <OPESP>
<CLASS> -> <PALRE> <PALRE> <CLASS2>
<PALRE> -> package | public | class | static | void | int | for | string | <PALRE> <PALRE> | <PALRE> <ID> | <PALRE> <OPASG>

<ID> -> main | vaTest | compiladores | Varargs_Proyect | args | println | length | <ID> <OPESP>
<OPESP> -> { | } | (|) | [|] | ; | : | , | .| … | " | ε | <OPESP> <PALRE> | <OPESP> <OPESP> | <OPESP> <ID> | <NUM> | <LETRA> | <CADENA> <OPESP> | <OPASG> | <OPESP> <OPARI> | <OPREL>
<OPASG> -> = | <OPASG> <OPESP> | <LETRA> <OPASG> | <OPASG> <NUMERO> | <OPASG> >LETRA>
<OPARI> -> + | <OPARI> <OPESP> | <OPARI> <LETRA>
<OPIND> -> ++ | <OPIND> <OPESP>
<OPREL> -> < |<OPREL> LETRA
<LETRA> -> { a, …, z} | LETRA <OPESP> | <LETRA> <OPASG> | <OPASG> <OPASG> | <LETRA> OPREL | <LETRA> OPIND | <LETRA> <OPARI>
<NUM> -> {0, …, 9} | NUM NUM | NUM <OPESP>
<CLASS2> -> IDE OPESP DESAROLLO

DESAROLLO -> PALRE METODO
METODO -> IDE NUM PALRE
CADENA -> OPESP CADENA| Número de args: | Contiene | arg

S-> PKG CLASS

          PALRE ID OPESP CLASS
          package compiladores ; CLASS
          package compiladores ; PALRE PALRE CLASS2
          package compiladores ; public class CLASS2
          package compiladores ; public class ID OPESP DESARROLLO
          package compiladores ; public class Varargs_Project { DESARROLLO
          package compiladores ; public class Varargs_Project { PALRE METODO
          package compiladores ; public class Varargs_Project { PARLE PALRE METODO
          package compiladores ; public class Varargs_Project { public PALRE METODO
          package compiladores ; public class Varargs_Project { public PALRE PALRE METODO
          package compiladores ; public class Varargs_Project { public static PALRE METODO
          package compiladores ; public class Varargs_Project { public static PALRE ID METODO
          package compiladores ; public class Varargs_Project { public static void ID METODO
          package compiladores ; public class Varargs_Project { public static void ID OPESP METODO
          package compiladores ; public class Varargs_Project { public static void main OPESP ID OPESP METODO
          package compiladores ; public class Varargs_Project { public static void main (String OPESP OPESP METODO
          package compiladores ; public class Varargs_Project { public static void main (String [OPESP ID METODO
          package compiladores ; public class Varargs_Project { public static void main (String [ ] ID OPESP METODO
          package compiladores ; public class Varargs_Project { public static void main (String [ ] args OPESP OPESP METODO
          package compiladores ; public class Varargs_Project { public static void main (String [ ] args) { METODO
          package compiladores ; public class Varargs_Project { public static void main (String [ ] args) { ID NUM
          package compiladores ; public class Varargs_Project { public static void main (String [ ] args) { ID OPESP NUM
          package compiladores ; public class Varargs_Project { public static void main (String [ ] args) { VaTest (NUM
          package compiladores ; public class Varargs_Project { public static void main (String [ ] args) { VaTest (NUM NUM
          package compiladores ; public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 NUM OPESP
          package compiladores ; public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0 OPESP OPESP
          package compiladores ; public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0) OPESP ID
          package compiladores ; public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0) ; ID OPESP
          package compiladores ; public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0) ; VaTest OPESP NUM
          package compiladores ; public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (NUM OPESP
          package compiladores ; public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 OPESP OPESP
          package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , NUM
          package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , NUM OPESP
          package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 OPESP OPESP
          package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , NUM
          package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , NUM OPESP
          package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3 OPESP OPESP
          package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) OPESP
          package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) OPESP ID
          package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; ID OPESP
          package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest OPESP OPESP

          package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /(OPESP

          package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /(OPESP OPESP

          package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) OPESP

          package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) OPESP OPESP

          package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; OPESP

          package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; OPESP PALRE

          package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; } PALRE

          package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ;} PALRE PALRE

          package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ;} static PARLE ID

          package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ;} static void ID OPESP

          package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ;} static void vaTest OPESP

          package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; }static void vaTest OPESP PALRE

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ;} static void vaTest (PALRE OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; }static void vaTest (int OPESP OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ;} static void vaTest (int ... OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; }static void vaTest (int ... LETRA

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; } static void vaTest (int ... LETRA OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; } static void vaTest (int ... v OPESP OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; } static void vaTest (int ... v) OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; } static void vaTest (int ... v) OPESP ID

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; } static void vaTest (int ... v) { ID OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; } static void vaTest (int ... v) { println OPESP OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; } static void vaTest (int ... v) { println (CADENA OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; }static void vaTest (int ... v) { println (OPESP CADENA OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: OPESP OPARI ***

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; }static void vaTest (int ... v) { println (" Número de args: " OPARI ***

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " OPARI OPESP ****

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + OPESP **

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + LETRA

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " v OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v OPESP ID

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . ID OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length OPESP OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) OPESP ID

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; ID OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println OPESP OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (OPESP OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; }static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" CADENA OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; }static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: OPESP OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; }static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: " OPESP OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") OPESP ID

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; ID OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for OPESP PALRE

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (PALRE OPASG

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int LETRA OPASG

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i OPASG NUM

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3); VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i = NUM OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3); VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i = 0 OPESP OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3); VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i = 0 ; LETRA

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3); VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i = 0 ; LETRA OPREL

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3); VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i = 0 ; i OPREL

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3); VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i = 0 ; i OPREL LETRA

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3); VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i = 0 ; i < LETRA

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3); VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i = 0 ; i < LETRA OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3); VaTest /( ) ; }static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i = 0 ; i < v OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3); VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i = 0 ; i < v OPESP ID

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3); VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i = 0 ; i < v. ID

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3); VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i = 0 ; i < v. ID OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3); VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i = 0 ; i < v. length OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3); VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i = 0 ; i < v . length OPESP OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3); VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i = 0 ; i < v . length ; OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3); VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i = 0 ; i < v . length ; LETRA

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3); VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i = 0 ; i < v . length ; LETRA OPIND

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3); VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i = 0 ; i < v . length ; i OPIND

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3); VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i = 0 ; i < v . length ; i OPIND OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3); VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i = 0 ; i < v . length ; i ++ OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3); VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i = 0 ; i < v . length ; i ++ OPESP ID

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3); VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i = 0 ; i < v . length ; i ++) ID **

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3); VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i = 0 ; i < v . length ; i ++) ID OPESP
package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3); VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i = 0 ; i < v . length ; i ++) println OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3); VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i = 0 ; i < v . length ; i ++) println OPESP   OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3); VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i = 0 ; i < v . length ; i + +) println (OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3); VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i = 0 ; i < v . length ; i + +) println (OPES   OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3); VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i = 0 ; i < v . length ; i + +) println (" OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3); VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i = 0 ; i < v . length ; i + +) println (" CADENA OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3); VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i = 0 ; i < v . length ; i + +) println (" arg OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3); VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i = 0 ; i < v . length ; i + +) println (" arg OPESP OPARI

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3); VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i = 0 ; i < v . length ; i + +) println (" arg "   OPARI

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3); VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i = 0 ; i < v . length ; i + +) println (" arg "   OPARI LETRA

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3); VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i = 0 ; i < v . length ; i + +) println (" arg " + LETRA

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i = 0 ; i < v . length ; i + +) println (" arg " + LETRA OPARI

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i = 0 ; i < v . length ; i + +) println (" arg " + i OPARI

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i = 0 ; i < v . length ; i + +) println (" arg " + i   OPARI OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i = 0 ; i < v . length ; i + +) println (" arg " + i + OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; }static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i = 0 ; i < v . length ; i + +) println (" arg " + i + OPESP OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i = 0 ; i < v . length ; i + +) println (" arg " + i + "   OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i = 0 ; i < v . length ; i + +) println (" arg " + i +   OPESP OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i = 0 ; i < v . length ; i + +) println (" arg " + i + " : OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i = 0 ; i < v . length ; i + +) println (" arg " + i + " : OPESP OPARI

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i = 0 ; i < v . length ; i + +) println (" arg " + i + " : OPARI

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i = 0 ; i < v . length ; i + +) println (" arg " + i + " :   " OPARI LETRA

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i = 0 ; i < v . length ; i + +) println (" arg " + i + " : " + LETRA

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i = 0 ; i < v . length ; i + +) println (" arg " + i + " :   " + LETRA OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i = 0 ; i < v . length ; i + +) println (" arg " + i + " :   " + v OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i = 0 ; i < v . length ; i + +) println (" arg " + i + " :   " + v OPESP OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i = 0 ; i < v . length ; i + +) println (" arg " + i + " :   " + v [OPESP **

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i = 0 ; i < v . length ; i + +) println (" arg " + i + " :   " + v [LETRA

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i = 0 ; i < v . length ; i + +) println (" arg " + i + " :   " + v [LETRA OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i = 0 ; i < v . length ; i + +) println (" arg " + i + " :   " + v [i OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i = 0 ; i < v . length ; i + +) println (" arg " + i + " :   " + v [i OPESP OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i = 0 ; i < v . length ; i + +) println (" arg " + i + " :   " + v [i] OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i = 0 ; i < v . length ; i + +) println (" arg " + i + " :   " + v [i] OPESP OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i = 0 ; i < v . length ; i + +) println (" arg " + i + " :   " + v [i]) OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i = 0 ; i < v . length ; i + +) println (" arg " + i + " :   " + v [i]) OPESP ID

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i = 0 ; i < v . length ; i + +) println (" arg " + i + " :   " + v [i]) ; ID

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i = 0 ; i < v . length ; i + +) println (" arg " + i + " :   " + v [i]) ; ID OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i = 0 ; i < v . length ; i + +) println (" arg " + i + " :   " + v [i]) ; println OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i = 0 ; i < v . length ; i + +) println (" arg " + i + " : " + v [i]) ; println OPESP OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i = 0 ; i < v . length ; i + +) println (" arg " + i + " : " + v [i]) ; println (OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i = 0 ; i < v . length ; i + +) println (" arg " + i + " :   " + v [i]) ; println (OPESP OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i = 0 ; i < v . length ; i + +) println (" arg " + i + " :   " + v [i]) ; println ( ) OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i = 0 ; i < v . length ; i + +) println (" arg " + i + " :   " + v [i]) ; println ( ) OPESP OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i = 0 ; i < v . length ; i + +) println (" arg " + i + " :   " + v [i]) ; println ( ) ; OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i = 0 ; i < v . length ; i + +) println (" arg " + i + " :   " + v [i]) ; println ( ) ; OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i = 0 ; i < v. length ; i + +) println (" arg " + i + " : " + v [i]) ; println ( ) ; } OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i = 0 ; i < v . length ; i + +) println (" arg " + i + " : " + v [i]) ; println ( ) ; } OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i = 0 ; i < v . length ; i + +) println (" arg " + i + " : " + v [i]) ; println ( ) ; } } OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; } static void vaTest (int ... v) { println (" Número de args: " + v . length) ; println (" Contiene: ") ; for (int i = 0 ; i < v . length ; i + +) println (" arg " + i + " :   " + v [i]) ; println ( ) ; } }

Caso 1 que falla

```
package compiladores;

    public class Varargs_Project{


        public static void main (String [] args){

            vaTest(10);
            vaTest(1,2,3);
            vaTest();
        }

        staticc void vaTest(int ... v){

            println(Número de args:  "+ v.length);
            println(Contiene: ");
            for(int i=0; i<v.length;i++){
                println(" arg "+i+": "+v[i]);

            println();

        }

    }
```

```
S-> PKG CLASS
            PALRE ID OPESP CLASS
            package compiladores ; CLASS
            package compiladores ; PALRE PALRE CLASS2
            package compiladores ; public class CLASS2
            package compiladores ; public class ID OPESP DESARROLLO
            package compiladores ; public class Varargs_Project { DESARROLLO
            package compiladores ; public class Varargs_Project { PALRE METODO
            package compiladores ; public class Varargs_Project { PARLE PALRE METODO
            package compiladores ; public class Varargs_Project { public PALRE METODO
            package compiladores ; public class Varargs_Project { public PALRE PALRE METODO
            package compiladores ; public class Varargs_Project { public static PALRE METODO
            package compiladores ; public class Varargs_Project { public static PALRE ID METODO
            package compiladores ; public class Varargs_Project { public static void ID METODO
            package compiladores ; public class Varargs_Project { public static void ID OPESP METODO
            package compiladores ; public class Varargs_Project { public static void main OPESP ID OPESP METODO
            package compiladores ; public class Varargs_Project { public static void main ( String OPESP OPESP METODO
            package compiladores ; public class Varargs_Project { public static void main ( String [ OPESP ID METODO
            package compiladores ; public class Varargs_Project { public static void main ( String [ ] ID OPESP METODO
            package compiladores ; public class Varargs_Project { public static void main ( String [ ] args OPESP OPESP METODO
            package compiladores ; public class Varargs_Project { public static void main ( String [ ] args ) { METODO
            package compiladores ; public class Varargs_Project { public static void main ( String [ ] args ) { ID NUM
            package compiladores ; public class Varargs_Project { public static void main ( String [ ] args ) { ID OPESP NUM
            package compiladores ; public class Varargs_Project { public static void main ( String [ ] args ) { VaTest ( NUM
            package compiladores ; public class Varargs_Project { public static void main ( String [ ] args ) { VaTest ( NUM NUM
            package compiladores ; public class Varargs_Project { public static void main ( String [ ] args ) { VaTest ( 1 NUM OPESP
            package compiladores ; public class Varargs_Project { public static void main ( String [ ] args ) { VaTest ( 1 0 OPESP OPESP
            package compiladores ; public class Varargs_Project { public static void main ( String [ ] args ) { VaTest ( 1 0 ) OPESP ID
            package compiladores ; public class Varargs_Project { public static void main ( String [ ] args ) { VaTest ( 1 0 ) ; ID OPESP
            package compiladores ; public class Varargs_Project { public static void main ( String [ ] args ) { VaTest ( 1 0 ) ; VaTest OPESP NUM
            package compiladores ; public class Varargs_Project { public static void main ( String [ ] args ) { VaTest ( 1 0 ); VaTest ( NUM OPESP
            package compiladores ; public class Varargs_Project { public static void main ( String [ ] args ) { VaTest ( 1 0 ); VaTest ( 1 OPESP OPESP
            package compiladores : public class Varargs_Project { public static void main ( String [ ] args ) { VaTest ( 1 0 ); VaTest ( 1 , NUM
            package compiladores : public class Varargs_Project { public static void main ( String [ ] args ) { VaTest ( 1 0 ); VaTest ( 1 , NUM OPESP
            package compiladores : public class Varargs_Project { public static void main ( String [ ] args ) { VaTest ( 1 0 ); VaTest ( 1 , 2 OPESP OPESP
            package compiladores : public class Varargs_Project { public static void main ( String [ ] args ) { VaTest ( 1 0 ); VaTest ( 1 , 2 , NUM
            package compiladores : public class Varargs_Project { public static void main ( String [ ] args ) { VaTest ( 1 0 ); VaTest ( 1 , 2 , NUM OPESP
            package compiladores : public class Varargs_Project { public static void main ( String [ ] args ) { VaTest ( 1 0 ); VaTest ( 1 , 2 , 3 OPESP OPESP
            package compiladores : public class Varargs_Project { public static void main ( String [ ] args ) { VaTest ( 1 0 ); VaTest ( 1 , 2 , 3 ) OPESP
            package compiladores : public class Varargs_Project { public static void main ( String [ ] args ) { VaTest ( 1 0 ); VaTest ( 1 , 2 , 3 ) OPESP ID
            package compiladores : public class Varargs_Project { public static void main ( String [ ] args ) { VaTest ( 1 0 ); VaTest ( 1 , 2 , 3 ) ; ID OPESP
            package compiladores : public class Varargs_Project { public static void main ( String [ ] args ) { VaTest ( 1 0 ); VaTest ( 1 , 2 , 3 ) ; VaTest OPESP OPESP
```

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /(OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /(OPESP OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) OPESP OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; OPESP PALRE

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; } PALRE

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ;} PALRE PALRE

Aquí es donde para el programa y falla, porque no existe la palabra staticc solo existe la palabra static, como se muestra aquí:

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ;} static PARLE ID

Caso 2 que falla
package compiladores;

```java
public class Varargs_Project{


    public static void main (String [] args){

        vaTest(10);
        vaTest(1,2,3);
        vaTest();
    }

    static void vaTestS(int ... v){

        println(Número de args:  "+ v.length);
        println(Contiene: ");
        for(int i=0; i<v.length;i++){
            println(" arg "+i+": "+v[i]);

        println();

     }

}
```

```
S-> PKG CLASS
PALRE ID OPESP CLASS
package compiladores ; CLASS
package compiladores ; PALRE PALRE CLASS2
package compiladores ; public class CLASS2
package compiladores ; public class ID OPESP DESARROLLO
package compiladores ; public class Varargs_Project { DESARROLLO
package compiladores ; public class Varargs_Project { PALRE METODO
package compiladores ; public class Varargs_Project { PARLE PALRE METODO
package compiladores ; public class Varargs_Project { public PALRE METODO
package compiladores ; public class Varargs_Project { public PALRE PALRE METODO
package compiladores ; public class Varargs_Project { public static PALRE METODO
package compiladores ; public class Varargs_Project { public static PALRE ID METODO
package compiladores ; public class Varargs_Project { public static void ID METODO
package compiladores ; public class Varargs_Project { public static void ID OPESP METODO
package compiladores ; public class Varargs_Project { public static void main OPESP ID OPESP METODO
package compiladores ; public class Varargs_Project { public static void main ( String OPESP OPESP METODO
package compiladores ; public class Varargs_Project { public static void main ( String [ OPESP ID METODO
package compiladores ; public class Varargs_Project { public static void main ( String [ ] ID OPESP METODO
package compiladores ; public class Varargs_Project { public static void main ( String [ ] args OPESP OPESP METODO
package compiladores ; public class Varargs_Project { public static void main ( String [ ] args ) { METODO
package compiladores ; public class Varargs_Project { public static void main ( String [ ] args ) { ID NUM
package compiladores ; public class Varargs_Project { public static void main ( String [ ] args ) { ID OPESP NUM
package compiladores ; public class Varargs_Project { public static void main ( String [ ] args ) { VaTest ( NUM
package compiladores ; public class Varargs_Project { public static void main ( String [ ] args ) { VaTest ( NUM NUM
package compiladores ; public class Varargs_Project { public static void main ( String [ ] args ) { VaTest ( 1 NUM OPESP
package compiladores ; public class Varargs_Project { public static void main ( String [ ] args ) { VaTest ( 1 0 OPESP OPESP
package compiladores ; public class Varargs_Project { public static void main ( String [ ] args ) { VaTest ( 1 0 ) OPESP ID
package compiladores ; public class Varargs_Project { public static void main ( String [ ] args ) { VaTest ( 1 0 ) ; ID OPESP
package compiladores ; public class Varargs_Project { public static void main ( String [ ] args ) { VaTest ( 1 0 ) ; VaTest OPESP NUM
package compiladores ; public class Varargs_Project { public static void main ( String [ ] args ) { VaTest ( 1 0 ); VaTest ( NUM OPESP
package compiladores ; public class Varargs_Project { public static void main ( String [ ] args ) { VaTest ( 1 0 ); VaTest ( 1 OPESP OPESP
package compiladores ; public class Varargs_Project { public static void main ( String [ ] args ) { VaTest ( 1 0 ); VaTest ( 1 , NUM
package compiladores ; public class Varargs_Project { public static void main ( String [ ] args ) { VaTest ( 1 0 ); VaTest ( 1 , NUM OPESP
package compiladores : public class Varargs_Project { public static void main ( String [ ] args ) { VaTest ( 1 0 ); VaTest ( 1 , 2 OPESP OPESP
package compiladores ; public class Varargs_Project { public static void main ( String [ ] args ) { VaTest ( 1 0 ); VaTest ( 1 , 2 , NUM
package compiladores : public class Varargs_Project { public static void main ( String [ ] args ) { VaTest ( 1 0 ); VaTest ( 1 , 2 , NUM OPESP
package compiladores : public class Varargs_Project { public static void main ( String [ ] args ) { VaTest ( 1 0 ); VaTest ( 1 , 2 , 3 OPESP OPESP
package compiladores : public class Varargs_Project { public static void main ( String [ ] args ) { VaTest ( 1 0 ); VaTest ( 1 , 2 , 3 ) OPESP
package compiladores : public class Varargs_Project { public static void main ( String [ ] args ) { VaTest ( 1 0 ); VaTest ( 1 , 2 , 3 ) OPESP ID
package compiladores : public class Varargs_Project { public static void main ( String [ ] args ) { VaTest ( 1 0 ); VaTest ( 1 , 2 , 3 ) ; ID OPESP
package compiladores : public class Varargs_Project { public static void main ( String [ ] args ) { VaTest ( 1 0 ); VaTest ( 1 , 2 , 3 ) ; VaTest OPESP OPESP
```

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /(OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /(OPESP OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) OPESP OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; OPESP

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; OPESP PALRE

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; } PALRE

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ;} PALRE PALRE

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ;} static PARLE ID

package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ;} static void ID OPESP

Aquí es donde para el programa y falla, porque no existe la palabra vaTestS solo existe la palabra vaTest, como se muestra aquí:
package compiladores : public class Varargs_Project { public static void main (String [ ] args) { VaTest (1 0); VaTest (1 , 2 , 3) ; VaTest /( ) ; } static void vaTest OPESP