

UNIVERSIDAD DEL PAPALOPAN

EXAMEN II PARCIAL

EQUIPO:

CARLOS T. GARCÍA CALDERÓN

CRISTEL R. GONZALEZ BARRIOS VERÓNICA

MALPICA MARTINEZ

M.C J. DOMINGO JUÁREZ HERNÁNDEZ

**GLOSARIO**

PKG	Paquete
CLASS	Clase
IDCLASS	Producción para nombrar la clase / identificador de clase.
CLASSR	Producción para iniciar el resto de las producciones del código / resto de la clase.
CRPCLASS	Me da la primera estructura para el cuerpo de la clase / cuerpo de clase.
DCVRL	Producción para la declaración de variable / declarar variable.
FNLDEC	Me ayuda a terminar la declaración de mi variable / fin de la declaración.
CRPCLASSR	Producción que me ayuda a continuar con el código después de DCVRL / resto de cuerpo de clase.
MAIN	Me da comienzo y continuación del 'main' de la clase / Main.
METHOD	Me da continuación a la producción del 'main' para después producirme el resto del método / método.
METHODR	Resto del método.
CRPMAIN	Producción para la estructura dentro del 'main' / cuerpo de main.
ARGS	Me da el nombre del método de 'main', produciendo el identificador dentro de él / argumento.
ARGSR	Resto de la producción ARGS
ARGSF	Cierra lo que inicio ARGS, produce los OPESP para ello / final de ARGS.
CRPMNR	Resto de producción para el cuerpo del 'main'
FOR	Da comiendo al ciclo for, produce las PALRE para ello/ for.
CRPFOR	Cuerpo del for.
INIC	Inicialización para el ciclo for.
INICR	Resto de la inicialización / ayuda a cerrar la inicialización.
COND	Condición para el ciclo FOR.
CONDR	Resto de la condición / ayuda a cerrar la condición.
INCR	Incremento para el ciclo for.
INCRR	Resto del incremento / ayuda a cerrar el incremento e inicializar la impresión.
PRINT	Imprimir.
PRINTR	Me ayuda a darle cuerpo a la estructura dentro del PRINT. / resto de PRINT.
PRINTF	Me ayuda a cerrar el print/ final de PRINT.

PRGCR	Produce el ultimo OPESP para el programa / cierre de programa.
-------	--

## PRODUCCIONES

S -> <PKG> <CLASS>      <PKG> -> <PALRE> <ID>  
 <OPESP>                    <CLASS>            -> <PALRE>  
 <PALRE> <IDCLASS>                    <IDCLASS> -> <ID> <  
 CLASSR>                    <CLASSR> -> <OPESP>  
 <CRPCLASS> <OPESP>                    <CRPCLASS> ->  
 <PALRE> <DCVRL> <CRPCLASSR>  
 <CRPCLASSR> -> <PALRE> <MAIN> <OPESP>  
 <DECVRL> -> <PALRE> <FNLDEC>                    <FNLDEC> ->  
 <ID> <OPESP>    <MAIN> -> <PALRE> <PALRE> <METOD>  
 <METOD> -> <PALRE> <OPESP> <METODR>  
 <METODR> -> <ARGS> <CRPMN>                    <ARGS> ->  
 <PALRE> <OPESP> <ARGSR>                    <ARGSR> -> <OPESP>  
 <ID> <ARGSF>    <ARGSF> -> <OPESP> <OPESP>  
 <CRPMN> -> <ID> <OPASI> <CRPMNR>    <CRPMNR> ->  
 <NUM> <OPESP> <FOR>                    <FOR> -> <PALRE>  
 <OPESP> <CRPFOR>    <CRPFOR> -> <INIC> <COND>  
 <INCR> <INIC>-> <ID> <OPAI> <INICR>    <INICR>->  
 <NUM> <OPESP>                    <COND>-> <ID>  
 <OPERL> <CONDR>    <CONDR> -> <NUM> <OPESP>  
 <INCR> -> <ID> <OPIND> <INCRR>                    <INCRR> ->  
 <OPESP> <OPESP> <PRINT>    <PRINT> -> <PALRE>  
 <OPESP> <PRINTR>                    <PRINTR> -> <STRING>  
 <OPARI> <PRINTF>                    <PRINTF> -> <ID>  
 <OPESP> <PRGCR>                    <PRGCR> -> <OPESP>  
 <OPESP>

<PALRE> -> package | public | class | static | void | string | println  
 ...  
 <OPESP> -> { | } | ( | ) | [ | ] | . | ? | , | / | \* | \* / | : | ;  
 <OPEREL> -> = | <= | < | >  
 <OPASI> -> = | \*= | /= | %= | += | -=  
 <OPARI> -> + | - | \* | / | %  
 <OPIND> -> ++ | --  
 <OPREL> -> == | < | > | >= | <= | !=  
 <NUM> -> 0...9  
 <ID> -> Compiladores | For\_Project | args | x  
 <String> -> "Valor de x: "

## CÓDIGO (CASO DE ÉXITO)

```
package Compiladores; public class
For_Project{      static int x;      public
static void main (String[] args){
    x=0;
    for (x=2; x<=4; x++){
        System.out.println("Valor de x: "+x);
    }
}
}
```

S -> <PKG> <CLASS>

<PKG> -> <PALRE> <ID> <OPESP>

-> package <ID> <OPESP>

-> package compiladores <OPESP>

-> package compiladores;

<CLASS> -> <PALRE> <PALRE> <IDCLASS>

-> public <PALRE> <IDCLASS>

-> public class <IDCLASS>

-> public class <ID> <CLASSR>

-> public class For\_Project <CLASSR>

-> public class For\_Project <OPESP> <CRPCLASS> <OPESP>

-> public class For\_Project { <CRPCLASS> <OPESP> <- queda pendiente '}'

<CRPCLASS> -> <PALRE> <DCVRL> <CRPOCLASSR> <OPESP>

-> static <PALRE> <FNLDEC> <CRPOCLASSR> <OPESP>

-> static int <FNLDEC> <CRPOCLASSR> <OPESP>

-> static int <ID> <OPESP> <CRPOCLASSR> <OPESP>

-> static int x <OPESP> <CRPOCLASSR> <OPESP>

-> static int x ; <CRPOCLASSR> <OPESP>

<CRPOCLASSR> -> <PALRE> <MAIN> <OPESP> <OPESP>

-> public <MAIN> <OPESP> <OPESP> <-queda pendiente el segundo '}'

-> public <PALRE> <PALRE> <METOD> <OPESP> <OPESP>

-> public static <PALRE> <METOD> <OPESP> <OPESP>

-> public static void <METOD> <OPESP> <OPESP>

-> public static void <PALRE> <OPESP> <METODR> <OPESP> <OPESP>

-> public static void main <OPESP> <METODR> <OPESP> <OPESP>

-> public static void main ( <METODR> <OPESP> <OPESP>

-> public static void main ( <ARGS> <CRPMN> <OPESP> <OPESP>

-> public static void main ( <PALRE> <OPESP> <ARGSR> <CRPMN> <OPESP> <OPESP>

-> public static void main ( String <OPESP> <ARGSR> <CRPMN> <OPESP> <OPESP>

-> public static void main ( String [ <ARGSR> <CRPMN> <OPESP> <OPESP>

-> public static void main ( String [ <OPESP> <ID> <ARGSF> <CRPMN> <OPESP> <OPESP>

-> public static void main ( String [ ] <ID> <ARGSF> <CRPMN> <OPESP> <OPESP>

-> public static void main ( String [ ] args <ARGSF> <CRPMN> <OPESP> <OPESP>

-> public static void main ( String [ ] args <OPESP> <OPESP> <CRPMN> <OPESP> <OPESP>

-> public static void main ( String [ ] args ) <OPESP> <CRPMN> <OPESP> <OPESP>

-> public static void main ( String [ ] args ) { <CRPMN> <OPESP> <OPESP>

<CRPMN> -> <ID> <OPASI> <CRPMNR> <OPESP> <OPESP>

-> x <OPASI> <CRPMNR> <OPESP> <OPESP>

-> x = <CRPMNR> <OPESP> <OPESP>

-> x = <NUM> <OPESP> <FOR> <OPESP> <OPESP>

-> x = 0 <OPESP> <FOR> <OPESP> <OPESP>

-> x = 0 ; <FOR> <OPESP> <OPESP>

```

<FOR> -> <PALRE> <OPESP> <CRPFOR> <OPESP> <OPESP>
-> for <OPESP> <CRPFOR> <OPESP> <OPESP>
-> for ( <CRPFOR> <OPESP> <OPESP>
-> for ( <INIC> <COND> <INCR> <OPESP> <OPESP>
-> for ( <INIC> <COND> <INCR> <OPESP> <OPESP>
-> for ( <ID> <OPAI> <INICR> <COND> <INCR> <OPESP> <OPESP>
-> for ( x <OPAI> <INICR> <COND> <INCR> <OPESP> <OPESP>
-> for ( x = <INICR> <COND> <INCR> <OPESP> <OPESP>
-> for ( x = <INICR> <COND> <INCR> <OPESP> <OPESP>
-> for ( x = <NUM> <OPESP> <COND> <INCR> <OPESP> <OPESP>
-> for ( x = 2 <OPESP> <COND> <INCR> <OPESP> <OPESP>
-> for ( x = 2 ; <COND> <INCR> <OPESP> <OPESP>
-> for ( x = 2 ; <ID> <OPERL> <CONDR> <INCR> <OPESP> <OPESP>
-> for ( x = 2 ; x <OPERL> <CONDR> <INCR> <OPESP> <OPESP>
-> for ( x = 2 ; x <= <CONDR> <INCR> <OPESP> <OPESP>
-> for ( x = 2 ; x <= <NUM> <OPESP> <INCR> <OPESP> <OPESP>
-> for ( x = 2 ; x <= 4 <OPESP> <INCR> <OPESP> <OPESP>
-> for ( x = 2 ; x <= 4 ; <INCR> <OPESP> <OPESP>
-> for ( x = 2 ; x <= 4 ; <ID> <OPIND> <INCRR> <OPESP> <OPESP>
-> for ( x = 2 ; x <= 4 ; x <OPIND> <INCRR> <OPESP> <OPESP>
-> for ( x = 2 ; x <= 4 ; x ++ <INCRR> <OPESP> <OPESP>
-> for ( x = 2 ; x <= 4 ; x ++ <OPESP> <OPESP> <PRINT> <OPESP> <OPESP>
-> for ( x = 2 ; x <= 4 ; x ++ ) <OPESP> <PRINT> <OPESP> <OPESP>
-> for ( x = 2 ; x <= 4 ; x ++ ) { <PRINT> <OPESP> <OPESP>

<PRINT> -> <PALRE> <OPESP> <PRINTR> <OPESP> <OPESP>
-> println <OPESP> <PRINTR> <OPESP> <OPESP>
-> println ( <PRINTR> <OPESP> <OPESP>
-> println ( <STRING> <OPARI> <PRINTF> <OPESP> <OPESP>

```

```

-> println ( "Valor de x: " <OPARI> <PRINTF> <OPESP> <OPESP>
-> println ( "Valor de x: " + <PRINTF> <OPESP> <OPESP>
-> println ( "Valor de x: " + <ID> <OPESP> <OPESP> <OPESP> <OPESP>
-> println ( "Valor de x: " + x <OPESP> <PRGCR> <OPESP> <OPESP>
-> println ( "Valor de x: " + x ) <PRGCR> <OPESP> <OPESP>
-> println ( "Valor de x: " + x ) <OPESP> <OPESP> <OPESP> <OPESP>
-> println ( "Valor de x: " + x ); <OPESP> <OPESP> <OPESP>
-> println ( "Valor de x: " + x ); <OPESP> <OPESP> <OPESP>
-> println ( "Valor de x: " + x ); } <OPESP> <OPESP>
                                } <OPESP>
                                }

```

## CÓDIGO (CASO DE ERROR 1)

```

package Compiladores; public
    class For_Project {
        int estatic x = 0 ;           public
static void main ( String [ ] args {
para (int i = 0; i < 10; i++) {       x = x +
i;
                                }
        }
    }
}

```

S -> <PKG> <CLASS>

<PKG> -> <PALRE> <ID> <OPESP>  
-> package <ID> <OPESP>  
-> package compiladores <OPESP>  
-> package compiladores;

<CLASS> -> <PALRE> <PALRE> <IDCLASS>  
-> public <PALRE> <IDCLASS>  
-> public class <IDCLASS>  
-> public class <ID> <CLASSR>  
-> public class For\_Project <CLASSR>  
-> public class For\_Project <OPESP> <CRPCLASS> <OPESP>

-> public class For\_Project { <CRPCLASS> <OPESP>

<CRPCLASS> -> <PALRE> <DCVRL> <CRPOCLASSR> <OPESP>

-> static <PALRE> <FNL> <CRPOCLASSR> <OPESP>

-> static int <FNL> <CRPOCLASSR> <OPESP>

-> static int <ID> <OPESP> <CRPOCLASSR> <OPESP>

-> static int x <OPESP> <CRPOCLASSR> <OPESP>

-> static int x <OPESP> <CRPOCLASSR> <OPESP>

Falla porque mi gramática me indica que lo siguiente debe ser un <OPESP> y no un <OPASI>.

## CÓDIGO (CASO DE ERROR 2)

Package Compiladores;

Public class Para\_Proyecto {

Static Float x; // Declarar la variable estática correctamente

Public static void main(String[] args) { // Corrección del tipo de método principal

X = 0.0f; // Inicializar con un valor flotante

// Corregir el bucle for para que sea compatible con el tipo Float

For (int i = 2; i <= 4; i++) {

X = (float) i; // Convertir i a Float para asignarla a x

System.out.println("Valor de x: " + x);

}

}

}

S -> <PKG> <CLASS>

<PKG> -> package <ID> <OPESP>

<CLASS> -> public class <ID> { <CRPCLASS> }

<CRPCLASS> -> <PALRE> <FNL> <CRPOCLASSR> <OPESP>

<FNL> -> static <PALRE> <TYPE> <ID> <OPASI> <NUM> <OPESP>

<TYPE> -> Float

<CRPOCLASSR> -> public static void <METOD> ( <ARGS> ) { <CRPMN> }

<METOD> -> main



<ARGS> -> String <OPESP> <ID> <OPESP>  
<ID> -> args  
<CRPMN> -> <FOR> <OPESP>  
<FOR> -> for ( <INIC> ; <COND> ; <INCR> ) { <PRINT> }  
<INIC> -> <ID> <OPAI> <NUM> <OPESP>  
<ID> -> i  
<NUM> -> 2  
<COND> -> <ID> <OPERL> <NUM> <OPESP>  
<OPERL> -> <=  
<NUM> -> 4  
<INCR> -> <ID> <OPIND> <NUM> <OPESP>  
<ID> -> i  
<NUM> -> 1  
<PRINT> -> System.out.println ( <STRING> <OPARI> <ID> ) <OPESP>  
<STRING> -> "Valor de x: "  
<OPARI> -> +  
<ID> -> x