

UNIVERSIDAD DEL PAPALOAPAN

EXAMEN 2DO PARCIAL

EQUPO:

YAXOL LEON GOMEZ

ARELI GABRIELA OSORIO CHAVEZ

M.C. JOSÉ DOMINGO JUAREZ HERNANDEZ

2 DE DICIEMBRE DE 2024

GLOSARIO

PKG	Define un paquete
CLASS	Define una clase
CLASSR	Define el cuerpo de la clase que está envuelto en llaves (OPESP)
CLASSB	El cuerpo de la clase contiene el método principal (MAIN).
MAIN	Define el método principal.
METHOD	El método está delimitado por paréntesis (OPESP)
ARGS	Define los argumentos del método principal.
MAINB	El cuerpo principal contiene un bucle do-while, seguido del cuerpo del bucle (LOOPB).
DO	La estructura do, que contiene el cuerpo del bucle y el comando de salida (SYSOUT), una expresión (EXPR), y finalmente la estructura while.
WHILE	Define la condición del bucle while con palabras reservadas (PALRE), operadores especiales (OPESP), y una condición (COND).
COND	Define una condición que compara un identificador con un número (NUM) usando operadores relacionales (OPRL).
LOOPB	El cuerpo del bucle que contiene la salida del sistema (SYSOUT) y una expresión (EXPR).
SYSOUT	La estructura para imprimir en la consola, seguido por operadores especiales y una cadena (STRINGR).
STRINGR	Define una cadena seguida por operadores aritméticos (OPARI) y un identificador (ID).
EXPR	Define una expresión que incrementa el valor de un identificador (ID) usando operadores de incremento (OPIND).
CRPCLASS	Define el cuerpo de la clase, empezando con public static void main para el método principal y los argumentos del método (ARGS), y las reglas adicionales para el cuerpo del método (CRPMN).
CRPMN	Define el contenido del método principal, con la asignación de un valor inicial a x y la estructura do-while
DO	Define el bucle do-while donde do contiene el cuerpo del bucle (PRINT y EXPR) y while define la condición para continuar el bucle.
PRINT	Define la impresión en la consola, utilizando System.out.println, seguido de una cadena y un identificador.
EXPR	Define la expresión que incrementa x utilizando el operador de incremento (++).
WHILE	Define la condición del bucle while, que compara x con 20 para determinar si el bucle debe continuar.
IDCLASS	Representa un identificador de clase.
CLASSR	Continúa la definición de la clase después del nombre de la clase. Esto incluye los delimitadores de bloque { y } que encierran el cuerpo de la

	clase, así como el contenido dentro de ese bloque, como variables, métodos, etc.
ID	Es un símbolo no terminal que representa un identificador. El identificador es un nombre utilizado para identificar variables, clases, métodos o cualquier otro elemento definido por el usuario.
METODR	Representa todo lo que viene después de la declaración de los parámetros del método, es decir, el cuerpo del método.

PRODUCCIONES

S -> PKG CLASS

PKG -> PALRE ID OPESP

CLASS -> PALRE ID CLASSR

CLASSR -> OPESP CLASSB OPESP

CLASSB -> MAIN

MAIN -> PALRE PALRE METHOD

METHOD -> OPESP ARGS MAINBODY

ARGS -> PALRE STRING OPESP

MAINBO -> DO WHILE LOOPB

DO -> PALRE OPESP SYSOUT EXPR OPESP WHILE

WHILE -> PALRE OPESP COND OPESP

COND -> ID OPRL NUM

LOOPB -> SYSOUT EXPR OPESP

SYSOUT -> SYSOUT OPESP STRINGR

STRINGR -> STRING OPARI ID

EXPR -> ID OPIND

PALRE -> package , public , class , static , void , int , do , while

OPESP -> { , } , [,] , (,) , ;

OPASI -> =

OPARI -> +

OPIND -> ++

OPREL -> <

NUM -> 0 ... 9

LTR -> a ... z , A ... Z

ID -> UNPA.Win.Form.Compiladores , DoWhile_Project , x

String -> String , "Valor de x: "

SYSOUT -> System.out.println

CODIGO

```
package UNPA.Win.Form.Compiladores;
```

```
public class DoWhile_Project {
```

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

```
        int x = 2;
```

```
        do
```

```
        {
```

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

```
            x++;
```

```
        }
```

```
        while (x < 20);
```

```
    }
```

```
}
```

S -> <PKG> <CLASS>

PKG -> PALRE ID OPESP

-> package ID OPESP

-> package UNPA.Win.Form.Compiladores OPESP

-> package UNPA.Win.Form.Compiladores;

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

-> public PALRE IDCLASS

-> public class <IDCLASS>

-> public class <ID> <CLASSR>

-> public class DoWhile_Project <CLASSR>

-> public class DoWhile_Project <OPESP> <CRPCLASS> <OPESP>

-> public class DoWhile_Project { <CRPCLASS> <OPESP> <- queda pendiente '}'

<CRPCLASS> -> <PALRE> <MAIN> <OPESP>

-> public <MAIN> <OPESP>

-> public <PALRE> <PALRE> <METHOD> <OPESP>

-> public static <PALRE> <METHOD> <OPESP>

-> public static void <METHOD> <OPESP>

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

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

-> public static void main (<METODR>

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

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

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

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

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

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

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

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

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

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

<CRPMN> -> <ID> <OPASI> <NUM> <OPESP> <DO> <OPESP>

-> x <OPASI> <NUM> <OPESP> <DO> <OPESP>

-> x = <NUM> <OPESP> <DO> <OPESP>

-> x = 2 <OPESP> <DO> <OPESP>

-> x = 2 ; <DO> <OPESP>

<DO> -> <PALRE> <OPESP> <PRINT> <EXPR> <OPESP> <WHILE>

-> do <OPESP> <PRINT> <EXPR> <OPESP> <WHILE>

-> do { <PRINT> <EXPR> } <WHILE>

-> do { <PRINT> <EXPR> } while (<COND> <OPESP>

-> do { <PRINT> <EXPR> } while (<ID> <OPRL> <NUM> <OPESP>

-> do { <PRINT> <EXPR> } while (x < 20 <OPESP>

-> do { <PRINT> <EXPR> } while (x < 20 ;

<PRINT> -> <SYSOUT> <OPESP> <STRINGR>

-> System.out.println (<STRINGR> <OPESP>

-> System.out.println ("Valor de x:" <OPARI> <ID> <OPESP>

-> System.out.println ("Valor de x:" + <ID> <OPESP>

-> System.out.println ("Valor de x:" + x <OPESP>

-> System.out.println ("Valor de x:" + x) <OPESP>

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

<EXPR> -> <ID> <OPIND> <OPESP>

-> x ++ <OPESP>

-> x ++ ;

<WHILE> -> <PALRE> <OPESP> <COND> <OPESP>

-> while (<ID> <OPRL> <NUM> <OPESP>

-> while (x < 20 <OPESP>

-> while (x < 20 ;} OPESP OPESP

}

OPESP

