Universidad del Papaloapan

Campus Loma Bonita

Teoría de la Computación

Proyecto:

Integrantes del Equipo:

Edgar Yeshua Ramírez Villa

Ramiro Muñoz Martínez

Ubaldo Santiago Gonzales

Semestre:
3ro

Profesor:
Domingo

Fecha:
02/12/24

```
Package  WS.curso.Scrum.CTRL;
Import java.util.Scanner:
public class NewClass {

    public static void main(String[] args) {
      int n, i;

      System.out.print("Introduce un número: ");
      Scanner leer = new Scanner(System.in);
      n = leer.nextInt();

      for(i = 0; i <= n; i++) {
        System.out.println(i);
      }
    }
  }
```

Gramática

S -> <PKG> <IMPORT> <CLASS>

<PKG> -> package <ID> ;

<IMPORT> -> import <ID> ;

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

<IDCLASS> -> <LTR> (<LTR> | <NUM>)*

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

<CRPMN> -> <DECLS> <INSTRUCTIONS>

<DECLS> -> int <ID> , <ID> ;

<INSTRUCTIONS> -> <PRINTIN> <SCANNERDECL> <ASSIGN> <FOR>

<PRINTIN> -> System.out.print (<STRING>) ;

<SCANNERDECL> -> Scanner <ID> = new Scanner (System.in) ;

<ASSIGN> -> <ID> = <METHODCALL> ;

<METHODCALL> -> <ID> . nextInt ( )

<FOR> -> for (<INIC> ; <COND> ; <INCR>) { <FORBLOCK> }

<INIC> -> <ID> = <NUM>

<COND> -> <ID> <OPEREL> <ID>

<INCR> -> <ID> <OPIND>

<FORBLOCK> -> System.out.println (<ID>) ;

Terminales

<PALRE> -> public | class | static | void | int | for | String

<OPESP> -> ; | { | } | (|) | [|]

<OPEREL> -> = | <= | >= | <

<OPASI> -> =

<OPIND> -> ++ | --

<NUM> -> 0 | 1 | 2 | ... | 9

<LTR> -> a | b | ... | z | A | B | ... | Z

<ID> -> <LTR> (<LTR> | <NUM>)*

<STRING> -> ".*"

<PALRE> -> Palabra reservada del lenguaje (e.g., public, class, void, etc.)

<OPESP> -> Operador o símbolo especial (e.g., ;, {, }, (,), etc.)

<OPEREL -> Operador relacional (e.g., =, <=, <, >)

<OPASI> -> Operador de asignación (=)

<OPARI> -> Operador aritmético (e.g., +, -, *, /)

<OPIND> -> Operador de incremento o decremento (++, --)

<OPREL> -> Operador relacional extendido (e.g., ==, >=)

<NUM> -> Números (0–9)

<LTR> -> Letras (mayúsculas o minúsculas, A–Z, a–z)

<ID> -> Identificadores válidos (e.g., nombres de clases, variables, objetos)

<String> -> Literal de cadena entre comillas (e.g., "Introduce un número: ")

<CLASS> -> Estructura general de una clase

<MAIN> -> Método principal (public static void main(String[] args))

<DECLS> -> Declaraciones de variables (int n, i;)

<PRINTIN> -> Instrucción para imprimir (e.g., System.out.print(...))

<SCANNERDECL> -> Declaración y creación de un objeto Scanner

<ASSIGN> -> Asignación de valores a una variable

<FOR> -> Declaración de un ciclo for

<FORBLOCK> -> Instrucciones dentro del bloque del ciclo for

Ejemplo valido

```
Package  WS.curso.Scrum.CTRL;
Import java.util.Scanner:

public class NewClass {

    public static void main(String[] args) {

      int n, i;


      System.out.print("Introduce un número: ");
      Scanner leer = new Scanner(System.in);

      n = leer.nextInt();

      for(i = 0; i <= n; i++) {

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

S -> <PKG> <Import> <CLASS>

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

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

<PKG> -> package WS.curso.Scrum.CTRL <OPESP>

<PKG> -> package WS.curso.Scrum.CTRL;

<Import> -> <PALRE> <ID> <OPESP>

<Import> -> import <ID> <OPESP>

<Import> -> import java.util.Scanner <OPESP>

<Import> -> import java.util.Scanner;

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

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

<CLASS> -> public class <IDCLASS>

<CLASS> -> public class <ID> <CLASSR>

<CLASS> -> public class NewClass <CLASSR>

<CLASS> -> public class NewClass <OPESP> <CRPCLASS> <OPESP>

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

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

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

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

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

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

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

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

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

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

<CRPMN> -> int n, i; <OPESP> <OPESP>

<CRPMN> -> Scanner leer = new Scanner(System.in); <OPESP> <OPESP>

<CRPMN> -> n = leer.nextInt(); <OPESP> <OPESP>

<CRPMN> -> for (i = 0; i <= n; i++) { <PRINT> <OPESP> <OPESP>

<CRPMN> -> System.out.println(i); <OPESP> <OPESP>

<CRPMN> -> } <OPESP> <OPESP>

<CRPMN> -> } <OPESP> <OPESP>

<CRPMN> -> } <OPESP> <OPESP>

2 ejemplos no validos

Ejemplo 1

```
Package  WS.curso.Scrum.CTRL;

Import java.util.Scanner:


public class {      <- no tiene nombre la clase marcara error por esa falta.


   public static void main(String[] args) {


     int n, i;



     System.out.print("Introduce un número: ");

     Scanner leer = new Scanner(System.in);


     n = leer.nextInt();


     for(i = 0; i <= n; i++) {
      System.out.println(i);
    }
  }
}
```

S -> <PKG> <Import> <CLASS>

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

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

<PKG> -> package WS.curso.Scrum.CTRL <OPESP>

<PKG> -> package WS.curso.Scrum.CTRL;

<Import> -> <PALRE> <ID> <OPESP>

<Import> -> import <ID> <OPESP>

<Import> -> import java.util.Scanner <OPESP>

<Import> -> import java.util.Scanner;

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

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

<CLASS> -> public class <IDCLASS>

<CLASS> -> public class <CLASSR>   no tiene nombre la clase el programa marcara error.

<CLASS> -> public class <CLASSR>

<CLASS> -> public class <OPESP> <CRPCLASS> <OPESP>

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

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

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

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

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

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

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

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

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

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

<CRPMN> -> int n, i; <OPESP> <OPESP>

<CRPMN> -> Scanner leer = new Scanner(System.in); <OPESP> <OPESP>

<CRPMN> -> n = leer.nextInt(); <OPESP> <OPESP>

<CRPMN> -> for (i = 0; i <= n; i++) { <PRINT> <OPESP> <OPESP>

<CRPMN> -> System.out.println(i); <OPESP> <OPESP>

<CRPMN> -> } <OPESP> <OPESP>

<CRPMN> -> } <OPESP> <OPESP>

<CRPMN> -> } <OPESP> <OPESP>

Ejemplo 2 de la derivación no valida

```
Package WS.curso.Scrum.CTRL;
Import java.util.Scanner:

public class NewClass {

    public static void main(String[] args {  // Falta el paréntesis de cierre ')'.

      int n, i;

      System.out.print("Introduce un número: ");
      Scanner leer = new Scanner(System.in);

      n = leer.nextInt();

      for(i = 0; i <= n; i++) {
        System.out.println(i);
      }
    }
  }
```

S -> <PKG> <Import> <CLASS>

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

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

<PKG> -> package WS.curso.Scrum.CTRL <OPESP>

<PKG> -> package WS.curso.Scrum.CTRL;

<Import> -> <PALRE> <ID> <OPESP>

<Import> -> import <ID> <OPESP>

<Import> -> import java.util.Scanner <OPESP>

<Import> -> import java.util.Scanner;

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

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

<CLASS> -> public class <IDCLASS>

<CLASS> -> public class <CLASSR>

<CLASS> -> public class <OPESP> <CRPCLASS> <OPESP>

<CLASS> -> public class NewClass <OPESP> <CRPCLASS> <OPESP>

<CLASS> -> public class NewClass { <CRPCLASS> <OPESP>

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

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

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

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

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

<CRPCLASS> -> public static void main (<METODR> <OPESP> <OPESP> <- Falta el cierre del paréntesis en la declaración.

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

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

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

<CRPMN> -> int n, i; <OPESP> <OPESP>

<CRPMN> -> Scanner leer = new Scanner(System.in); <OPESP> <OPESP>

<CRPMN> -> n = leer.nextInt(); <OPESP> <OPESP>

<CRPMN> -> for (i = 0; i <= n; i++) { <PRINT> <OPESP> <OPESP>

<CRPMN> -> System.out.println(i); <OPESP> <OPESP>

<CRPMN> -> } <OPESP> <OPESP>

<CRPMN> -> } <OPESP> <OPESP>

<CRPMN> -> } <OPESP> <OPESP>