



UNIVERSIDAD ESTATAL A DISTANCIA
VICERRECTORÍA ACADÉMICA
ESCUELA DE CIENCIAS EXACTAS Y NATURALES
Cátedra Tecnología de Sistemas



[Compiladores]

Código: [3307]

Proyecto 2

Temas de Estudio

Tema 3 Análisis léxico.

Tema 4 Análisis sintáctico.

Tema 5 Traducción orientada por la sintaxis.

Tema 6 Generación de código intermedio.

Objetivo

Poner en práctica los conocimientos adquiridos en la asignatura, en cada uno de los temas citados y relacionar los conceptos para madurar algunas de las principales ideas de un compilador.

Software de Desarrollo

Java, usando NetBeans en modo carácter. No se debe usar el modo gráfico.

Desarrollo

- El proyecto debe tener independencia física en la ejecución, lo que quiere decir que el programa debe poder ser ejecutado en cualquier máquina y en cualquier carpeta.
- Para el desarrollo de este Proyecto 2 se asume que se desarrolló el Proyecto 1 y que todos los puntos evaluados en él están correctos.
- Por lo tanto, se sigue el mismo enfoque: leer un archivo de texto, el cual va a contener un código Java; asumiendo que todo lo que se pidió en el Proyecto 1 está desarrollado el estudiante debe validar los puntos que se citan en la siguiente tabla y si hubiera errores se deben reportar en el archivo respectivo.
- En general, se puede asumir que toda línea trae solo un comando.

1	COMENTARIOS Validar que si una línea empieza con estos caracteres: // (sin espacio entre ellos) se debe interpretar como comentario y entonces se puede asumir que el resto de la línea es un comentario y por lo tanto no se debe validar nada más. Se puede asumir, también, que nunca vendrán comentarios de los delimitados por /* y */
2	IDENTIFICADORES Validar la correcta definición de identificadores para nombrar las variables tomando en cuenta que en Java los identificadores se definen siguiendo las siguientes reglas: <ul style="list-style-type: none"> • Deben comenzar con una letra. • Se permite el carácter subrayado (_) o el carácter de dollar (\$). • Pueden incluir dígitos, pero no comenzar con uno. • No pueden incluir el carácter espacio o blanco. • Son significativas las letras mayúsculas y minúsculas. • No se pueden utilizar las palabras reservadas (ver punto 10) como identificadores. • El identificador no debe existir ni siquiera con otro tipo de datos.
3	TIPOS DE DATOS Validar la correcta definición de los tipos de datos de los identificadores para nombrar las variables tomando en cuenta que en Java los tipos de datos pueden ser: <ul style="list-style-type: none"> • char • byte • short • int • long • float • double • boolean Se puede asumir que las variables se declaran siempre sin hacerles asignación.
4	ASIGNACIÓN Validar que la asignación de variables se haga mediante el operador relacional “=”. A las variables se les puede asignar constantes, variables o expresiones que sean del mismo tipo. Se debe validar que la variable a la izquierda del igual exista. Se puede asumir que todo lo que viene a la derecha del igual es correcto. Pero se deben validar que todos los tipos de datos de las variables que aparecen a la derecha del igual coinciden con el tipo de datos de la variable que aparece a la izquierda del igual.
5	PREPARAR LA ENTRADA DE DATOS Se debe validar que aparezca este comando escrito exactamente de esa manera: <pre>static Scanner sc = new Scanner(System.in);</pre> Además, se debe validar que aparezca después del main y que solo aparezca una vez.
6	ENTRADA DE DATOS Se debe validar que los comandos de entrada de datos sigan esta sintaxis: <pre>variable = método de lectura</pre> La variable debe existir.

	<p>El método de lectura debe ser congruente con el tipo de datos de la variable y puede ser uno de los siguientes; se debe validar que se escriban literalmente así:</p> <pre> sc.nextLine() (para char y strings) sc.nextByte() sc.nextShort() sc.nextInt() sc.nextLong() sc.nextFloat() sc.nextDouble() sc.nextBoolean() </pre>
7	<p>SALIDA DE DATOS</p> <p>Se debe validar que el comando para salida de datos siga esta sintaxis:</p> <pre>println ("lo-que-se-quiere-imprimir")</pre> <p>Siempre deben aparecer ambos paréntesis y comillas dobles tanto al inicio como al final de lo que se quiere imprimir. lo-que-se-quiere-imprimir no se debe validar.</p>
8	<p>DECISIÓN</p> <p>Se debe validar que el comando if cumpla la siguiente sintaxis:</p> <pre> if (condición) { comandos } else { comandos } </pre> <p>Se puede asumir que no vienen if anidados. Se puede asumir que la condición es correcta y por lo tanto no se debe validar. Si el if aparece entonces debe tener al menos un comando. El else es opcional; pero si aparece entonces debe tener al menos un comando. No puede aparecer un else sin if.</p>
9	<p>CICLO</p> <p>Se debe validar que el comando while cumpla la siguiente sintaxis:</p> <pre> while (condición) { comandos } </pre> <p>Se puede asumir que no vienen while anidados. Se puede asumir que la condición es correcta y por lo tanto no se debe validar. Si el while aparece entonces debe tener al menos un comando.</p>
10	<p>PALABRAS RESERVADAS</p> <p>Si el primer lexema de una línea no corresponde con alguno de los primeros 9 puntos se debe validar si es una palabra reservada de Java. Si lo fuera entonces simplemente se ignora el resto de la línea y no se debe validar nada más. Pero si no lo fuera entonces se debe enviar el mensaje: "Comando Java inválido".</p>

Los siguientes son ejemplos de programas en Java que podrían usarse para calificar el Proyecto 2. Sin embargo, se debe tomar en cuenta que pueden venir blancos redundantes o líneas en blanco redundantes, de manera que tu programa debe ser capaz de procesarlos de manera adecuada. Además, a propósito, el tutor introducirá errores entre las líneas para verificar que tu programa envíe los mensajes de error apropiados.

```
public class Operaciones {
    public static void main(String args[]) {
        // inicio del programa
        static Scanner sc = new Scanner(System.in);
        int num1;
        int num2;
        println("Primer numero:");
        num1 = sc.nextInt();
        println("Segundo numero:");
        num2 = sc.nextInt();
        println("El resultado de la suma es "+(num1+num2));
        println("El resultado de la resta es "+(num1-num2));
        println("El resultado de la multiplicación es "+(num1*num2));
        println("El resultado de la división es "+(num1/num2));
        // fin del programa
    }
}
```

```
public class DivisionSegura {
    public static void main(String args[]){
        // inicio del programa
        static Scanner sc = new Scanner(System.in);
        int x;
        int y;
        int z;
        x = 12;
        y = 2;
        z = 0;
        if( y !=0 ) {
            z = x / y;
            println("El resultado es : " + z);
        } else {
            println("Atención! se pretende dividir por 0");
        }
        // fin del programa
    }
}
```

```
public class Discriminante {
    public static void main (String args[]) {
        // inicio del programa
        static Scanner sc = new Scanner(System.in);
        double a;
        double b;
        double c;
        double discriminante;
        a = 1;
        b = 2;
        c = 1;
        discriminante = b*b - 4*a*c;
        if (discriminante > 0) {
            println("Hay dos soluciones reales distintas.");
        }
        if (discriminante == 0) {
            println("Hay una solución única.");
        }
        if (discriminante < 0) {
            println("No hay soluciones reales.");
        }
        // fin del programa
    }
}

public class MuestraNumWhile {
    public static void main(String args[] ) {
        // inicio del programa
        static Scanner sc = new Scanner(System.in);
        int num;
        num=1;
        while (num<=100){
            println(num);
            num=num+1;
        }
        // fin del programa
    }
}
```

Honestidad Académica



<https://audiovisuales.uned.ac.cr/play/player/23048>

Nota Importante

Cada estudiante es responsable del contenido que entrega, si no es el archivo correcto, no podrá entregarlo posterior a la fecha establecida.

Si el contenido del archivo coincide con algún otro estudiante, o se comprueba que no es de su autoría, se aplicaría lo indicado en la plataforma en el documento [Lineamientos ante casos de plagio](#)

Indicaciones Importantes

- Es obligatorio que incluya todo el directorio donde se encuentra el Proyecto 2.
- Este proyecto debe estar desarrollado en NETBEANS que es la herramienta oficial de la asignatura.
- El programa debe ser modular, utilizando de la mejor manera funciones definidas por usted.
- Los trabajos deben realizarse en forma individual. Dentro del código del programa debe de indicar la documentación que explique cómo fue realizado el programa.
- Si utiliza código de algún ejemplo del libro, o de otra fuente que no sea de su autoría, debe de indicarlo.
- Comprima todos los archivos en un solo archivo .zip o .rar.
- Nombre del archivo que envía: debe ser nombre y primer apellido del estudiante, y nombre del proyecto. Ejemplo: **JuanRojas-Proyecto2**
- La entrega de Proyecto 2 en las fechas establecidas en la plataforma de aprendizaje en línea Moodle en el apartado que se indique.
- Si no concluyó a tiempo este proyecto, debe entregar lo que pudo hacer e incluir una carta explicando las razones por las cuales no finalizó.

Rúbrica de Evaluación

Criterio	Cumple de manera excelente lo indicado en la evaluación	Cumple bastante bien lo indicado en la evaluación	Cumple medianamente lo indicado en la evaluación	Cumple en contenido y formato, pero los aportes no son significantes	No cumple o no presenta lo solicitado
Realización punto #1	10	7	5	3	1
Realización punto #2	10	7	5	3	1
Realización punto #3	10	7	5	3	1
Realización punto #4	10	7	5	3	1
Realización punto #5	10	7	5	3	1
Realización punto #6	10	7	5	3	1
Realización punto #7	10	7	5	3	1
Realización punto #8	10	7	5	3	1
Realización punto #9	10	7	5	3	1
Realización punto #10	10	7	5	3	1
TOTAL	100	70	50	30	10