



Universidad Autónoma de Zacatecas

Unidad Académica de Ingeniería Eléctrica

Programa Académico de Ingeniería de Software

Práctica 12

Datos generales:

Nombre de la Práctica	Pilas
Nombre de la carrera	Ingeniería de Software
Nombre de la materia	Laboratorio de Estructuras de Datos
Número y nombre de Unidad(es) temática(s)	I1. Estructuras lineales.
Docente que imparte la materia	Aldonso Becerra Sánchez
Fecha de entrega para los alumnos	13-septiembre-2021
Fecha de entrega con extensión y penalización	14-septiembre-2021
Fecha de elaboración	13-septiembre-2021

Objetivo de la Práctica	Comprender el uso de las pilas en un ejemplo común.
Tiempo aproximado de realización	2.5 horas
Introducción	Las pilas son estructuras de datos fáciles de entender, sin embargo, sus aplicaciones muchas veces no son tan sencillas de visualizar en la práctica. Las pilas facilitan muchos procesos, un caso común son el manejo de notaciones infija, prefija y postfija.

Referencias que debe consultar el alumno (si se requieren):

Referencia 1:

1. Cairo, Osvaldo; Guardati, Silvia. Estructura de Datos, Tercera Edición. McGraw-Hill, México, Tercera Edición, 2006.

Referencia 2:

2. Mark Allen Weiss. Estructura de datos en Java. Ed. Addison Wesley.

Referencia 3:

3. Joyanes Aguilar, Luis. Fundamentos de Programación. Algoritmos y Estructuras de Datos. Tercera Edición, 2003. McGraw – Hill.



Universidad Autónoma de Zacatecas

Unidad Académica de Ingeniería Eléctrica

Programa Académico de Ingeniería de Software

Actividades que debe realizar el alumno:

Actividad inicial:

Generar el reporte en formato IDC.

Actividad 1:

Primero genere la **introducción**.

Actividad 2:

Hacer un programa que funcione como una calculadora.

Este programa permitirá funcionar como una calculadora, debe recibir una expresión aritmética en una caja de texto (o el teclado), por ejemplo de la forma:

$(y+2)*\text{resultadoSuma}/2^{(\text{iterador1}+12)}$

Asuma que la entrada es una expresión válida, es decir no deberá realizar validaciones de si está bien formada la expresión (eso lo deberá hacer otro módulo en otro programa, que pudiera usarse independientemente del funcionamiento del programa que se le pide a usted).

El programa deberá recibir de entrada esta cadena con la expresión aritmética. Posteriormente se deberá obtener el resultado de la operación ejecutada (evaluar la expresión).

Consideraciones de diseño:

- a) Considere variables de máximo 30 caracteres de longitud siguiendo las reglas:
 - Una variable comienza con una letra o guion bajo.
 - Las variables son CASE SENSITIVE.
 - Una variable puede contener solo letras, guiones bajo, números y signos de pesos (\$).
 - No debe realizar uso de expresiones regulares de API.
 - Los agrupadores permitidos son el paréntesis (...).
 - Ejemplo de variables válidas: suma2, var_1, resultado, _división, SUMA\$.
 - Ejemplo de variables inválidas: 1suma1, \$suma1, suma 2.
 - El separador de las variables será un operador, paréntesis o espacio (debe asumirse que puede ser uno o varios espacios, los cuales deben ignorarse).
 - El separador de los números o constantes numéricas será un operador, paréntesis o espacio (debe asumirse que puede ser uno o varios espacios, los cuales deben ignorarse).
- b) Revise la expresión aritmética para checar si tiene variables; en caso que tenga, pida los valores que esas variables debe tener para sustituirlas por su contenido.



Universidad Autónoma de Zacatecas

Unidad Académica de Ingeniería Eléctrica

Programa Académico de Ingeniería de Software

c) Considere que los números son enteros y decimales, y de cualquier longitud.
Ejemplos: 456, 56, 4, 44443, 4.5, 9.99.

Ejemplo:

Entrada: $(a-2)*y/2^{(suma-9)}$

donde $a=1$, $y=3$, $suma = 12.0$

Salida: -0.375

NOTA: recuerde que no puede usar librerías de Java que automaticen ningún proceso de manipulación de grupos de datos, en este caso de cadenas.

Al final el programa deberá obtener el resultado de la expresión infija ingresada, para ello deberá convertir la cadena en una expresión “postfija” o “prefija” (usando pilas), para posteriormente realizar la evaluación de dicha expresión convertida (usando pilas).

Haga el programa (actividad 2, la cual es el **Desarrollo** del programa, junto con la captura de pantalla del programa funcionando).

Actividad 3:

Pruebe el funcionamiento del programa de la actividad 2 con todo y sus capturas de pantalla.

Actividad 4:

Realice la sección de **Código agregado** (diagrama de clases UML).

Actividad 5:

Realice la sección de **Pre-evaluación** (use los lineamientos establecidos).

Actividad 6:

Finalmente haga las **Conclusiones**.

Actividad 5:

Enviar en <http://ingsoftware.reduaz.mx/moodle>

Archivo anexo que se requiere para esta tarea (opcional):

Dudas o comentarios: a7donso@gmail.com