

## Laboratorio 4

---

Ing. Msc. Víctor Orozco

1 de septiembre de 2022

### 1. DESCRIPCIÓN

El objetivo del presente laboratorio es presentar al estudiante un problema complejo utilizando Cool: The Classroom Object-Oriented Language como lenguaje de programación.

### 2. LABORATORIO

Para iniciar la resolución por favor utilice las siguientes tareas de GitHub Classroom:

<https://classroom.github.com/a/qukA-0z3>

### 3. UNA PILA COOL (85 %)

Un programa con una sola pila para almacenamiento es denominado comúnmente una máquina de pila. Consideremos entonces el siguiente Lenguaje primitivo para programar una máquina de pila:

Comando	Significado
<i>int</i>	enviar el <i>int</i> hacia el stack
+	push de '+' en el stack
s	push de 's' en el stack
e	evaluar el tope de la pila
d	mostrar el contenido de la pila
x	detener el programa

El comando 'd' simplemente imprime el contenido de la pila, un elemento por línea, comenzando con la parte superior de la pila. El comportamiento del comando 'e' depende del contenido de la pila cuando 'e' es emitido:

- Si '+' está en la parte superior de la pila, entonces el '+' se saca de la pila, los siguientes dos enteros también son extraídos y sumados, y el resultado se vuelve a colocar en la pila.
- Si 's' está en la parte superior de la pila, entonces se extrae la 's' y los siguientes dos elementos se intercambian en la pila.
- Si hay un número entero en la parte superior de la pila o la pila está vacía, la pila no se modifica.

Los siguientes ejemplos muestran el efecto del comando 'e' en varias situaciones; la parte superior de la pila está en la columna izquierda:

---

1	+	1	2	5	s...		3	5	s...
2	s	1	+	+	99	.	.	.	+ 1 + 99
3	1	+	3...				1	+	3...

---

En esta línea, usted debe implementar un intérprete para este lenguaje en COOL. La entrada al programa es una serie de comandos, un comando por línea. Su intérprete debe solicitar comandos con >. No es necesario realizar ninguna comprobación de errores, puede suponer que todos los comandos son válidos y que el número y tipo de argumentos están en la pila para su evaluación.

También puede suponer que la entrada los enteros no tienen signo. Su intérprete debería salir sin problemas del programa, es decir simplemente finalizar el algoritmo con un último prompt '>'.

Usted es libre de implementar este programa en cualquier estilo que elija. Sin embargo, en preparación para la construcción un compilador de COOL, la recomendación es implementar una solución basada en POO.

Si lo desea, puede usar las clases definidas en `atoi.cl` en el directorio "examples" para realizar una conversión de entero a cadena, así como cualquier otro ejemplo.

Como referencia una solución para este problema en un lenguaje de programación estructurado y POO suele tener entre 200 y 300 líneas de código.

Para iniciar este ejercicio usted encontrara en su repositorio base un directorio denominado LAB2 en el cual encontrara un archivo 'README.md' con instrucciones adicionales.

### 3.1. EJEMPLO

La siguiente salida es un ejemplo de una sesión de ejecución de cool:

---

```
1 $ spim -file pila.s
2 SPIM Version 5.6 of January 18, 1995
3 Copyright 1990-1994 by James R. Larus (larus@cs.wisc.edu).
4 All Rights Reserved.
5 See the file README a full copyright notice.
6 Loaded: [cool root]/lib/trap.handler
7 >1
8 >+
9 >2
10 >s
11 >d
12 s
13 2
14 +
15 1
16 >e
17 >e
18 >d
19 3
20 >x
21 > COOL program successfully executed
```

---

#### 4. PREGUNTAS (15 %)

Utilizando el archivo 'README.md' responda en formato Markdown las interrogantes ahí incluidas

Fecha límite de entrega: Sabado 11 de junio, 23:59