|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Asignatura | Estructura de Datos | | | | |
| Código | 1155304 número de Créditos:3 | | | | |
| Área de formación: | *Ciencias Básicas* | Ciencias Básicas Aplicada | Profesional específica (X) | | Socio Humanística |
| Tipo de asignatura: | Obligatoria(X) | |  | | |
| Nombre del Profesor | MARCO A. ADARME JAIMES | | | Código:04041 | |

Semestre: I ( ) II (x) año : 2020

|  |  |
| --- | --- |
| Información de la Actividad | |
| Nombre | Manejo de estructuras de datos estáticas en un caso problémico. |
| Competencias a desarrollar | |
| Ser capaces de desarrollar un programa con estructuras de datos pilas, colas y manejo básico de una GUI | |
| Tipo de Actividad: Teórica ( ) Práctica ( X ) | |
| Fecha de entrega (día/mm/aa): 11 de dic 2020, hora:12m | |
| Descripción | |
| **CALCULADORA DE EXPRESIONES ARITMÉTICAS**  Una expresión aritmética está formada por operandos y operadores. La expresión x\*y – (a+b)  consta de los operadores \*, -, + y de los operandos x, y, a, b. Los operandos pueden ser valores constantes, variables o, incluso, otra expresión. Los operadores son los símbolos conocidos de las operaciones matemáticas. La evaluación de una expresión aritmética da lugar a un valor numérico, se realiza sustituyendo los operandos que son variables por valores concretos y ejecutando las operaciones aritméticas representadas por los operadores (Joyanes Aguilar & Zahonero Martínez, 2011). Si los operandos de la expresión anterior toman los valores x = 5, y = 2, a = 3, b = 4, el resultado de la evaluación es:  5\*2 – (3+4) = 5\*2 – 7 = 10 – 7 = 3  Su evaluación se realiza tomando en cuenta los niveles de prioridad en orden de mayor a menor(Cortés Nazar, 2019):  Paréntesis : ( )  Potencia : ^  Multiplicación/división : \*, /  Suma/Resta : +, -  Una expresión puede estar definida por 2 notaciones adicionales: Prefija y postfija. La Expresión o Notación PreFija nos indica que el operador va antes de los operandos y la postfija lo indica se refiere a que el operador ocupa la posición después de los operandos. Ambos son evaluados según los niveles de prioridad.  **Notación prefija: El orden es operador, primer operando, segundo operando**  **Notación postfija: El orden es primer operando, segundo operando, operador**  Se debe crear un programa que permita:   * Registrar una expresión infija * Evaluar la expresión (dar su resultado) * Imprimir su notación prefija * Imprimir su notación postfija   Recomendaciones de implementación:   1. El proceso de evaluación se puede realizar en la notación que bien considere. 2. Sólo se pueden usar : ListaCD, Pilas y Colas del paquete SEED 3. Realice todas las validaciones para el correcto funcionamiento del programa. 4. Tome en cuenta el siguiente diagrama base para su implementación:      1. La operación de potencia(^) se resuelve usando Math.pow(..). 2. El sistema sólo se toma en cuenta los signos de agrupamiento de paréntesis. 3. Para el sistema un número con signo hacer parte de un token “unario”, es decir:    * la expresión : -3+5 contiene los tokens: -3, +, 5    * la expresión : -3+-5 contiene los tokens: -3, +, -5    * la expresión : --3+-5 GENERAR ERROR (--).   **Protocolo de Entrega:**   1. Trabajo máximo en equipos de trabajo de dos(2) estudiantes. 2. Registrar proyecto en gitlab de forma privada, se debe colocar los dos estudiantes en el proyecto y adicional colocar el usuario “madarme” con rol “maintener”. 3. Realizar la documentación usando javadoc de las clase del Negocio 4. El nombre del archivo debe ser: “CODIGO1-CODIGO2.zip”, donde “CODIGO1” y “CODIGO2” corresponden a sus código ufps que lo identifican como estudiantes del programa. Ejemplo: 1150587-1150569.zip 5. No se reciben trabajos por fuera de la fecha de entrega. 6. Debe ser entregado en classroom en el formulario respectivo   *CUALQUIER FRAUDE MOTIVA A LA ANULACIÓN PARA LOS PLAGIADOS COMO PARA LOS PLAGIARIOS*  **RECUERDE SI TIENE PREGUNTAS AL EMAIL O SOLICITAR ASESORÍAS**  **Ejemplos:**   |  |  |  |  | | --- | --- | --- | --- | |  | **(2+(3\*4))** | **((2+3)\*4)** | **4\*(5-2)/(1\*9^3)** | | **Notación prefija** | + ,2, \*, 3, 4 | \*, +, 2, 3, 4 | /,\*,4,-,5,2,\*,1,^,9,3 | | **Notación infija** | 2+3\*4 | (2+3)\*4 | 4\*(5-2)/(1\*9^3) | | **Notación postfija** | 2, 3, 4, \*, + | 2, 3, +, 4, \* | 4,5,2,-,\*,1,9,3,^,\*,/ |   **Éxitos** | |

**Bibliografía**

Cortés Nazar, E. (2019). *MANUAL DE PRÁCTICAS LENGUAJES Y AUTÓMATAS II*.

Joyanes Aguilar, L., & Zahonero Martínez, I. (2011). *Programación en Java: Algoritmos, programación orientada a objetos e interfaz gráfica de usuario.* McGraw Hill.