

Universidad de Costa Rica  
Facultad de Ingeniería  
Escuela de Ciencias de la Computación e Informática  
CI-1201 Programación II - Profesor: Rubén Jiménez Goñi  
Examen Parcial I. Valor: 20%.  
Fecha: 6 de junio del 2015  
Entrega: 9 de junio antes de las 5:00 p.m. por Mediación Virtual.

Nombre: \_\_\_\_\_ Grupo 5

Resuelva el siguiente problema que se le plantea:

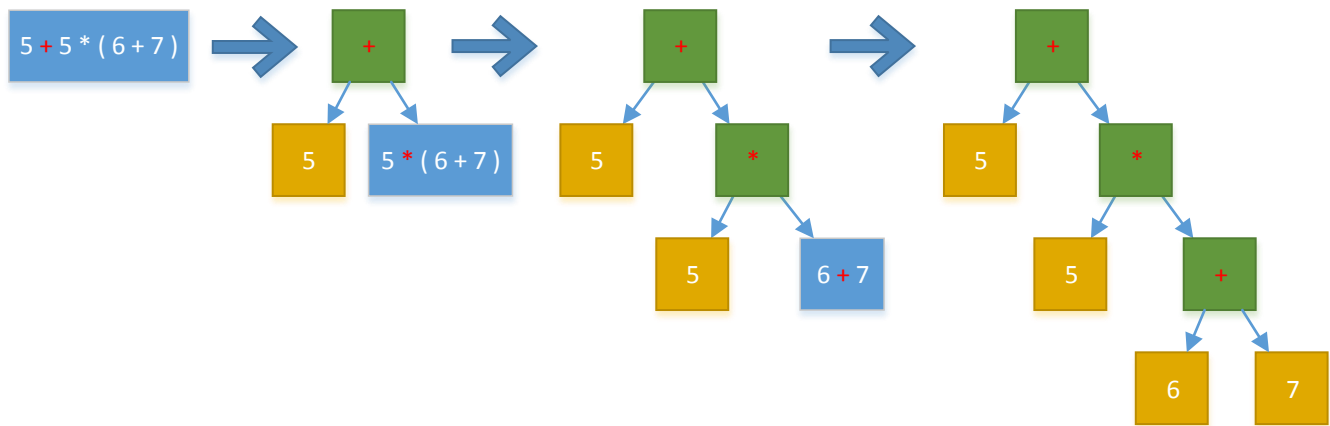
- Imagine que usted tiene un archivo con una serie de operaciones matemáticas y necesita saber los resultados. Los operadores pueden ser: sumas, restas, divisiones y multiplicaciones. Asimismo, algunas operaciones tienen operaciones agrupadas dentro de paréntesis, dentro de los cuales puede haber más agrupaciones hasta cualquier profundidad. Cree una aplicación capaz de abrir dicho archivo, procesar línea por línea cada operación y escribir los resultados en otro archivo. Al procesar el archivo, su aplicación debe respetar las reglas de precedencia de operadores y paréntesis.

La solución de su examen está restringida a las siguientes reglas:

1. No puede utilizar las clases de la *stl* (*Standard Template Library*). Únicamente puede utilizar la *Lista*, la *Pila* y el *Árbol Binario* que hemos realizado en clases como laboratorios y ejemplos. Puede realizar las modificaciones que desee, mientras siga utilizando Herencia y Polimorfismo.
2. Cuando procesa el archivo de entrada, debe crear una *Lista* de operaciones.
3. Cada vez que procesa una operación, agrega el resultado de la misma en una *Lista* y luego la escribe en un archivo.
4. Debe utilizar la misma jerarquía de clases utilizada en el *EjemploCalculadora*.
5. Para resolver cada operación no puede utilizar una *Pila*. Debe utilizar un *Árbol Binario*.
6. Agregue una clase *Operación* que herede de *Elemento*. De forma que su árbol pueda contener ya sea: operadores, operandos u operaciones.
7. Procese el árbol de la siguiente forma:
  - a. Inicialmente su árbol contiene una operación completa.
  - b. Posteriormente cree un método que procese su árbol de la siguiente forma:
    - i. Si un nodo es una operación, busque el operador de menor precedencia y divida la operación en: 2 operaciones (u operandos) y un operador. Ponga el operador en el lugar de la operación y cree 2 hijos que pueden ser operaciones (u operandos).
    - ii. Si el nodo es un operando u operador, no haga nada
    - iii. Continúe haciendo el mismo proceso mientras haya operaciones dentro del árbol.
    - iv. Considere que los paréntesis tienen la precedencia más alta.

Ejemplo:

- Creación del árbol:



- Solución del árbol

