Universidad del Istmo de Guatemala
Facultad de Ingenieria
Ing. en Sistemas
Informatica II
Prof. Ernesto Rodriguez - erodriguez@unis.edu.gt

Laboratorio #8

Fecha de entrega: 28 de Marzo, 2019 - 11:59pm

Instrucciones: Resolver cada uno de los ejercicios siguiendo sus respectivas instrucciones. El trabajo debe ser entregado a traves de Github, en su repositorio del curso, colocado en una carpeta llamada "Laboratorio #8". Al menos que la pregunta indique diferente, todas las respuestas a preguntas escritas deben presentarse en un documento formato pdf, el cual haya sido generado mediante Latex. Este laboratorio debe ser elaborado en parejas.

Tarea #1 (20%)

Definir una funcion llamada "`bool` *parseInt*(`const std ::string&` *valor*, `int&` *resultado*)" la cual debe aceptar un **string** y lo intenta convertir a un numero entero y almacena ese entero en la referencia "resultado" que se provee como parametro. A continuación se muestra un ejemplo del comportamiento de dicha función.

Listing 1: Tarea 1

```
int main(){
    std::string str1("12345");
    std::string str2("12345_mal");

    int resultado = 0;
    if(parse(str1, resultado)){
        // Imprime "Resultado 12345"
        printf("Resultado1 %i", num1);
    }

    if(!parse(str2, resultado)){
        // Imprimie "No se pudo parsear str2"
        printf("No se pudo parsear str2");
    }
}
```

Tarea #2 (20%)

Definir una clase virtual llamada "Semantica" con los siguientes metodos virtuales:

- `bool` *parse*(`const std ::string` *valor*, `int` *resultado*) `const`

- `int` *opSuma*(`const int` *arg*1, `const int` *arg*2) `const`
- `int` *opProducto*(`const int` *arg*1, `const int` *arg*2) `const`

Tarea #3 (20%)

Defina las clases "Arith" y "ZArith" las cuales deben implementar la clase virtual "Semantica".

La clase "Arith" debe utilizar la función "parse" para implementar el metodo "parse". El metodo "opSuma" simplemente debe sumar sus parametros y el metodo "opMult" debe multiplicar sus parametros.

La clase "ZArith" debe tener un constructor que toma un parametero numerico llamado "base". Esta clase debe comportarse exactamente igual que "Arith" pero sus metodos deben aplicar la operación "modulo (%)" a todos los resultados de sus metodos siendo "base" el segundo parametro a la operación "modulo".

Tarea #4 (40%)

Definir la función "*bool evaluar*(`const Semantica`&, `const std` :: `string`& *expression*, `int`& *resultado*)". Esta función acepta como parametro la semantica que se utilizara para evaluar la expresión que se provee en el parametro expressión. El resultado, si la evaluación es exitosa, debe ser colocado en el parametro "resultado" y debe retornar **true** si la evaluación fue exitosa o **false** de lo contrario. El codigo a continuación ejemplifica como debe comportarse dicha función:

Listing 2: Tarea #4

```
int main(){
    std::string ex1("2*3+5");
    std::string ex2("2*mal+3");
    int resultado;
    Arith sem1;
    ZArith sem2(5);

    if(evaluar(sem1, ex1, resultado)){
        // Imprime: "El resultado es '11'"
        printf("El resultado es '%i'", resultado);
    }

    if(evaluar(sem2, ex1, resultado)){
        // Impirme: "El resultado es '1'"
        printf("El resultado es '%i'", resultado);
    }

    if(!evaluar(sem1, ex2, resultado)){
        // Imprime: "No se pudo evaluar ex2"
        printf("No se pudo evaluar ex2".);
    }
}
```