

Practica 3

Analizador Sintáctico LR(1)

Implementación usando Objetos

Objetivo:

En esta práctica utilizaras una pila de objetos en lugar de enteros, de esta forma al momento que imprimas la pila aparecerán los símbolos de forma similar a cuando realizas el análisis manualmente.

Requerimientos:

Para esta práctica necesitaras crear una clase ElementoPila y modificaras la clase pila para que acepte objetos de este tipo en lugar de enteros.

Necesitaras crear 3 clases más, las cuales heredan de ElementoPila, las clases son:

- Terminal
- No terminal
- Estado

Nota: Realiza una copia del proyecto de la clase anterior, para que tengas un respaldo.

Ejemplo

En este ejemplo vamos a crear una clase Alumno, y dos clases que heredan de él: Bachillerato y Licenciatura. Después modificaremos la pila para que acepte objetos alumno, esta clase sirve solo como base para las otras clases, en realidad no crearemos ningún objeto "Alumno", solo Bachillerato o Licenciatura.

En el archivo pila.h, agrega el siguiente código (antes de la clase pila) :

```
class Alumno{
protected:
    string codigo;
public:
    virtual void muestra(){}
};

class Bachillerato: public Alumno{
protected:
    string preparatoria;
public:
    Bachillerato(string codigo, string preparatoria){
        this->codigo= codigo;
        this->preparatoria= preparatoria;
    }
    void muestra(){
        cout << "Alumno Bachillerato" << endl;
        cout << "Codigo: " << codigo << endl;
        cout << "Preparatoria: " << preparatoria << endl << endl;
    }
};

class Licenciatura: public Alumno{
protected:
    string carrera;
    int creditos;
public:
    Licenciatura(string codigo, string carrera, int creditos){
        this->codigo= codigo;
        this->carrera= carrera;
        this->creditos= creditos;
    }
    void muestra(){
        cout << "Alumno Licenciatura" << endl;
        cout << "Codigo: " << codigo << endl;
        cout << "Carrera: " << carrera << endl;
        cout << "Creditos: " << creditos << endl << endl;
    }
}
```

```
};
```

En el archivo pila.cpp, modifica las funciones para que en lugar de utilizar un int utilicen un objeto **Alumno***.

```
void Pila::push( Alumno *x){
    lista.push_front(x);
}

Alumno* Pila::pop(){
    Alumno* x= *lista.begin();
    lista.erase ( lista.begin() );

    return x;
}

Alumno* Pila::top(){
    return *lista.begin();
}

void Pila::muestra(){
    list <Alumno*>::reverse_iterator it;
    Alumno *x;
    cout << "Pila: ";

    for (it= lista.rbegin(); it != lista.rend(); it++){
        x= *it;
        x->muestra();
        //cout << (*it) << " ";
    }

    cout << endl;
}
```

En el archivo principal.cpp agrega una función de ejemplo, la cual llamas desde la función main.

```
void ejemplo(){
    Pila pila;
    Alumno *alumno;

    alumno= new Licenciatura("345678", "Computacion", 200);
    pila.push(alumno);

    pila.push(new Bachillerato("456789", "Preparatoria 12"));
    pila.push(new Licenciatura("456789", "Informatica", 50));
    pila.muestra();

    cout << "*****" << endl;

    pila.pop();
    pila.muestra();
}
```

Ejecuta tu programa.

Una vez que funcione tu programa, agrega las clases ElementoPila, Terminal, No Terminal y Estado, como se te indico al principio de la práctica. **Modifica los push, y pop, de todos los ejercicios de la práctica anterior para que ahora utilicen ElementoPila***