

Teoría de la Computación 2019/2020

Práctica 5. Análisis semántico con JFlex y Cup.

Dado el analizador sintáctico que se implementó en la práctica anterior, se deberán añadir las siguientes funcionalidades:

- Recuperación de errores sintácticos. El programa no se detendrá cuando tiene lugar un error sintáctico. Informará del mismo de la manera más precisa posible y continuará con el análisis.
- Detección de los siguientes errores semánticos:
 - Nombres de departamentos y cargos repetidos.
 - Registros de trabajadores cuyo departamento o cargo no corresponda a un elemento ya declarado en la lista correspondiente.
- Mostrar el total de trabajadores de la empresa en función del número de personas que hay en cada departamento.

Como mejora de la práctica, se podrá comprobar que el número total de empleados en cada departamento es el correcto en función del listado de trabajadores declarado. Además, se podrá comprobar que el cargo que ostenta un trabajador es válido para ese departamento. La relación admitida de departamentos y cargos en las empresas es:

Departamento	Cargos
Programación	Desarrollador, Analista, Director de proyectos
Marketing	Trafficker digital, Creativo, Director de proyectos
Negocio	Analista, Director de proyectos

Esta funcionalidad es *extra* y se evalúa aparte, según se indica en el siguiente apartado.

Entregable

La práctica se realizará de manera individual, y se entregará mediante el espacio de la asignatura en Moodle. Para ello se subirá un archivo comprimido con formato:

APELLIDO1_APELLIDO2_NOMBRE_P5.zip

El archivo deberá incluir en la raíz:

- Proyecto con el código generado, tanto los archivos .flex y .cup como los archivos Java creados por el alumno y los generados por las herramientas.
- Memoria cuyo nombre seguirá el formato: APELLIDOS_NOMBRE_MEMORIA.pdf. Deberá contener la información entregada en la práctica anterior, corrigiendo los posibles problemas presentados y añadiendo: si se ha hecho alguna modificación significativa al código de la práctica anterior (p.ej., cambios en el .flex, tipos de los símbolos), qué reglas semánticas ha habido que añadir y para qué sirve cada una de ellas, etc. Cometer en dicha memoria más de **3 faltas de ortografía o tener más de 3 párrafos señalados como ilegibles**, supondrá una calificación de **suspenso** en la memoria.

La fecha límite para la entrega de la práctica será el miércoles 8 de diciembre para el grupo A y el miércoles 15 de diciembre para el grupo B. Las entregas estarán habilitadas hasta a las 21:00. **Los archivos que no cumplan con el formato de entrega o sean entregados con posterioridad, supondrán una calificación de no presentado en la práctica.**

Si se lleva a cabo la mejora propuesta, se podrán sumar hasta dos puntos adicionales a los hasta 10 puntos sobre los que se evalúa la práctica. Esta práctica tendrá un peso del 50% sobre la nota total de prácticas de la asignatura.

Sugerencias

Los terminales que guardan los identificadores de trabajadores pueden declararse como de tipo *String*. Eso implica, entre otras cosas, que en el archivo *flex* se debe pasar la cadena que ha dado lugar al token al objeto de clase *Symbol*.

Se debe declarar una estructura de datos (p.ej., de tipo *ArrayList*) para los trabajadores, implementando los métodos correspondientes que trabajen con ella. Todo esto se puede poner en la parte correspondiente al *parser code* del archivo *cup* y llamar luego a esos métodos en las reglas del archivo *cup* como *parser.insertaTrabajador(trabajador)* o de forma similar. Puede ser conveniente crear una clase *Trabajador* con atributos para luego crear objetos de dicha clase y que sean los que estén en la estructura de datos. No será válida una mera lista de *String*, ya que deben guardarse también los valores correspondientes.