Paradigmas de Programación (EIF-400) Spec: `Kokoslan` y FP-OOP

DR. CARLOS LORÍA-SÁENZ

LIBERADO: 27 DE SETIEMBRE 2017

EIF400 /UNA

Introducción

- Especificar los alcances y entregables el proyecto programado sobre Parsing, Interpretación y Compilación
- Definir lo esperado en distintos aspectos del proyecto
- Establecer mecanismos de revisión y evaluación

Resumen de productos

- P1: Un Parser del lenguaje `Kokoslan`
- P2: Un intéprete de `Kokoslan`
- P3: Casos de prueba para el lenguaje
- P4: Un compilador de `Kokoslan` que genere `Prolog` (potencialmente opcional)
- P1, P2, y P3 son obligatorios
- P4 se hará sólo si el tiempo lo permite.

Objetivos Generales

- Poner en práctica técnicas de programación funcional-OOP aplicada en un caso de uso procesamiento de un lenguaje
- Experimentar con herramientas, paradigmas y tendencias acordes con el curso
- Ejercitar parsing, interpretación y compilación
- Fomentar la investigación temáticas relacionadas al tema de paradigmas en el área de lenguajes y traducción
- Ayudar al estudiante a ampliar su visión sobre temas básicos de ingeniería de software

Problema

- `Kokoslan` es un lenguaje que implementa el cálculo lambda de forma particular
- Los programas son en (archivos de) texto (sufijo `.kl`).
- Permite definiciones, `lambdas`, aritmética entera y booleana y listas (tipos de datos correspondientes). Opcionalmente permite strings.
- Tiene un `print(ln)` que es capaz de imprimir cualquier dato) en consola
- Permite pattern-matching en las lambdas
- Permite evaluación `eager` (por defecto) pero también `lazy` (opcional)
- El problema de estudio es escribir un parser y un intéprete para `Kokoslan`, como mínimo
- Eventualmente escribir un compilador que genere `Prolog` para que corra en esa plataforma

Consideraciones iniciales

- La tarea requiere nociones básicas de parsing e interpretación en las herramientas que se piden
- Se explicará en clase/consulta lo necesario para arrancar asumiendo trabajo en equipo
- Se estima que con unas 4 horas de explicación se puede trabajar en el proyecto

Requerimientos de organización

- Solo puede hacerse en los grupos previamente inscritos desde el primer proyecto
- No se acepta de otra forma
- No se aceptan nuevos grupos SIN EXCEPCIÓN.

Requerimientos diseño e implementación

- Eliminar lo más que se pueda todo patrón imperativo y estado mutable usando FP
- Usar OOP combinada con FP.
- Usar la arquitectura basada en ASTs, visitadores proveída por ANTLR4

Tipo de aplicación y Herramientas obligatorias

- `Kokoslan` es una aplicación sobre `Java8` desktop de consola.
- El intérprete es un REPL. Permite ejecutar un `.kl`.
- Está escrita en `Kotlin` mayormente y `Java` en las partes que sea necesario solamente
- Es una aplicación de consola (no es requerido un `GUI`, esto puede ser un extra). Se usan `.bat` (`scripts`) para interactuar con la aplicación
- El parser está en `ANTLR4` usando visitadores (estos se generan automáticamente en `Java7` por limitaciones de `ANTLR4`)
- El parser, intérprete y eventual compilador están en `Kotlin`

Casos de Prueba Estudiante

- Se deben escribir casos de prueba (archivos de texto `.kl`) que representan features implementados
- Deben ser al menos 10 casos en tres grupos:
  - Sprint Básicos(3 casos): Calculadora: definiciones de valores, cálculos aritméticos, booleanos y de listas
  - Sprint Definiciones funcionales primer orden (3 casos) : factorial, máximo de una lista, reversar una lista
  - Sprint Definiciones funcionales orden superior (4 casos): compose, map, filter, reduce

Casos de Prueba Profesor

- Para la revisión el profesor podrá poner sus propios casos de prueba, variantes de los del estudiante
- La nota de la evaluación dependerá de estos casos también

Pruebas y errores

- Sólo nos concentramos en probar cosas que están bien escritas.
- No ejercitamos manejo de errores del parser
- `Kokoslan` no es simbólico como el cálculo lambda, es decir si al evaluar encuentra símbolos no definidos levanta una excepción y no evalúa el caso más.

Evaluación y Valor

- Demo en clase: 60% (unos 15 minutos por grupo)
- Mejor traerlo en su propia laptop por si acaso haya problemas de setup en el lab. Si no logran la demo por razones ajenas al profesor reciben cero. La revisión se basa principalmente en cumplimiento casos de prueba
- Revisión de código 40%. Se revisará que se emplean técnicas FP-OOP, arquitectura y herramientas según lo solicitado
- Este proyecto vale al menos un 50% del porcentaje dedicado a proyectos (revisable a favor del estudiante, eventualmente)
- Se puede otorgar hasta un 30% extra sobre la nota si se añaden más funcionalidades o herramientas (previamente discutidas con el profesor). Sólo aplica si se tiene lo mínimo obligatorio

Construcción en consola

- Se espera que la aplicación pueda ser construida y corrida usando una herramienta apropiada desde una consola.
- Ud. entrega scripts (bats) que la construyen eficazmente desde consola
- La construcción y corrida no deben depender de un IDE.
- Opcional y muy preferiblemente usar un repositorio de manejo de fuentes versionados (SCM) como Git/Github o comparable

Entregables

- El proyecto en un `.rar` (o equivalente) llamado `kokoslan_AAA`, donde `AAA` son los autores. Se enviará por correo al profesor. El `Subject` siendo "`kokoslan_AAA`" (`AAA` como antes) y en el correo de nuevo vienen los autores y grupos
- Dentro del `rar` viene un directorio (carpeta) `kokoslan` (con el proyecto) y un `README.txt` con los nombres de los autores, su horario.
- Incumplimiento de estos requisitos anula la revisión que hará el profesor (pierden su 60%). Sin derecho a reclamo.
- El conjunto de casos de prueba que sirvan para verificar la funcionalidad implementada

Fecha y hora de entrega

- El día exacto de entrega se da a conocer oportunamente en clase y por los medios usuales en el curso: esperable en semana 18 (7-11 noviembre)
- Ese día de entrega (mismo de la demo) se envía el entregable por el medio requerido en la hora asignada. Esta dependa de la constitución del grupo: normalmente la hora de matrícula de la mayoría de miembros
- Entregas posteriores e injustificadas podrán recibir pérdida de puntos de hasta 5 puntos por minuto de anticipo o atraso hasta llegar a cero.
- Es obligación entregar casos de prueba de estudiante como parte del proyecto. Sino no se acepta el proyecto.