1- Opciones del compilador:

El compilador recibe como entrada un archive .cl (COOL) y devuelve un archivo con código MIPS.

2- Arquitectura del compilador:

El programa funciona usando el patrón visitor para ejecutar cada una de sus fases.
Dispone de 5 módulos diferentes (sin contar el chequeo lexicográfico):

1. Fase de chequeo semántico
2. Fase de construcción de tipos
3. Fase de chequeo de tipos
4. Generación de código de 3 direcciones (CIL)
5. Generación de código MIPS

Se ha utilizado una gramática LALR para el lexer del compilador.

Los diferentes errores se van dando en cada fase. En caso de haber algún error en una fase se muestran los errores de la misma y culmina el proceso de compilación. En caso de detectarse algún error en Runtime no se muestra el tipo de error y finaliza el programa.

3- Problemas técnicos

Los problemas técnicos más interesantes presentados en el proceso de construcción del compilador fueron el hecho de hacer boxing/unboxing para transformar tipos por valor a tipos por referencia a través del tipo Object. Fueron necesario controles a la hora de generación de MIPS para detectar tipos por valor y reservarles espacio en memoria para almacenarlos como tipos por referencia.
En cuanto en el momento del unboxing en la instrucción case hubo que detectar si lo que se tenía en la variable requería de hacerle unboxing para hacer la comparación de tipos en las ramas de la instrucción.

El chequeo de tipos en Runtime se resuelve utilizando una tabla generada en MIPS almacenada en la memoria estática, que guarda en sus primeros 4 bytes la referencia a la posición en la tabla donde se encuentra el padre, en los 4 siguientes el nombre del tipo y posteriormente referencias al punto de entrada de cada método. En caso de Object, la referencia a su padre es 0. Dicho esquema quedaría de la manera siguiente:

Ref. al Padre	Ref. al nombre	Ref. Método 1	(…)	Ref. Método n

4- Líneas Futuras:

- Implementación del SELF_TYPE.
- Añadir a la gramática comentarios de código.
- Añadir self como referencia al tipo actual.
- Optimización de código.

5- Link al repositorio público del código fuente:

https://www.github.com/youkai95/compilador-2019