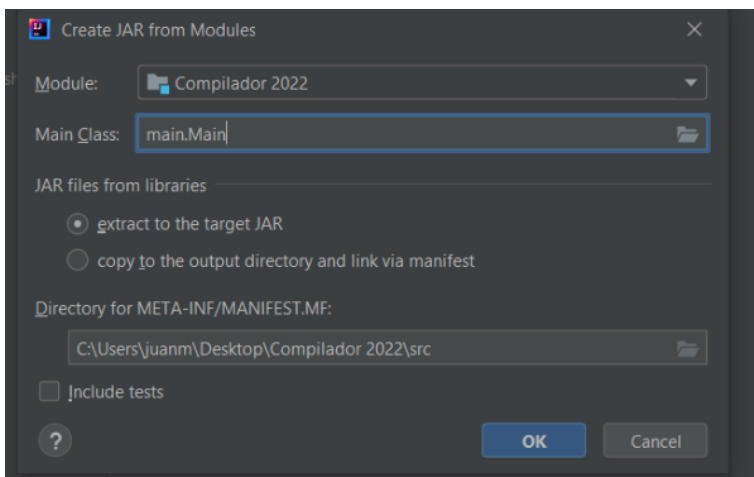
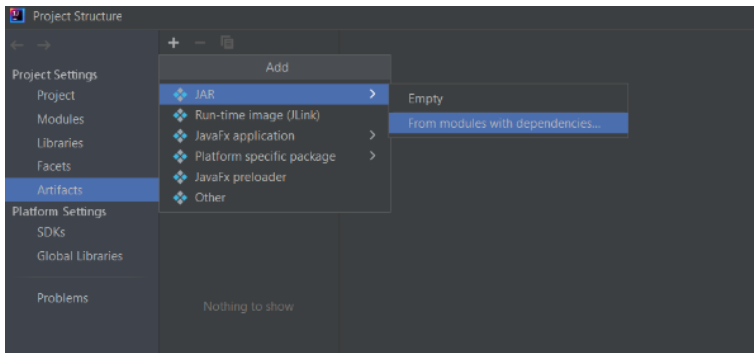


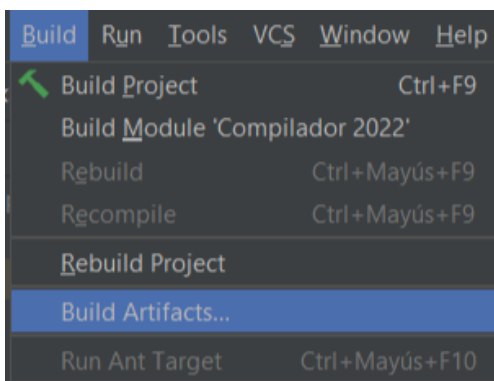
Informe Analizador Semántico parte 1

Como compilar el código fuente

Para poder compilar el código fuente en IntelliJ primero es necesario crear un “artifact”, esto se hace desde el menú de Project Structure en el apartado Artifacts.



Luego, en la barra de herramientas ir a la parte de “build artifacts...”



Luego de esto se creará la carpeta Artifact dentro de la carpeta out. Que contendrá una carpeta con el código compilado en un .jar

Para ejecutar este jar debe abrir cmd y dirigirse a la dirección donde se encuentre el .Jar y luego ejecutar el siguiente comando `java -jar [nombre .jar] [nombre del archivo a pasar por parámetro]`

Tipos de errores semánticos

Los tipos de errores semánticos que es posible detectar son los especificados en las reglas de semánticas subidas por la cátedra.

Decisiones de diseño

Para esta etapa, agregue un nuevo paquete `TablaDeSimbolos` en donde están todas las clases necesarias para realizar esta etapa:

- **Atributo:** esta clase lo que hace es verificar que el tipo del atributo que sea un tipo valido
- **Clase:** esta clase es una clase abstracta en donde se encarga de agregar métodos, atributos y setear un constructor si es que la clase no tenía uno
- **ClaseConcreta:** esta clase verifica que no haya herencia circular, guarda los métodos, atributos, chequea si está bien declarada y consolidada y verifica si se implementaron todos los métodos
- **Constructor:** esta clase verifica que este bien declarada
- **Interface:** esta clase verifica, además de que este bien declarada y consolidada que se chequee la extensión circular, agrega los métodos verificando que no se encuentre ya en la lista de los mismos, también verifica que si extiende de una interfaz o una clase que estas existan.
- **Método:** en esta clase verifica que este bien declarada, que este consolidada, guarda los parámetros correctamente, comprueba que la lista de parámetros sea igual, también verifica los métodos de la clase heredada no sean iguales a los del padre y chequea que los parámetros sean del mismo tipo y tengan el mismo tipo de retorno en caso de que sobre escriban el método
- **Parámetro:** esta clase verifica la existencia de los mismos y los añade a la lista de parámetros
- **TablaDeSimbolos:** en esta clase se crean las clases predefinidas: `Object`, `System` y `String` con sus respectivos métodos, tiene un método `limpiar()` en donde reinicia las clases, métodos e interfaces, verifica que exista una clase y una interfaz, verifica que este bien declarado y consolidado e inserta una clase y una interfaz

En el caso de que una clase no contenga el método `main()` el compilador retornara el eof del archivo

Logros que intenté alcanzar:

- Imbatibilidad