

## 0.1 Ambiente

In questo progetto l'ambiente viene definito come un insieme di scope, gestiti dalla classe **Environment**, che contengono una HashMap di stringhe e STentry corrispondenti alla entry di una variabile nella Symbol table. Quest'ultima è gestita dalla classe **STentry**. Environment e STentry contengono i metodi per gestire sia l'analisi semantica che l'analisi degli effetti.

### 0.1.1 STentry

Una entry è definita dal **nesting level** in cui compare, il **tipo** che possiede e l'**offset** al quale si trova a partire dal frame pointer. Questo è tutto quello che viene utilizzato di una STentry per l'analisi semantica. Questa classe viene utilizzata dall'Environment per costruire una entry di una variabile all'occorrenza ed aggiungerla al proprio scope.

### 0.1.2 Environment

Come detto sopra Environment si occupa di manipolare la Symbol Table, rappresentata come un parametro della classe, e possiede inoltre informazioni come il livello di annidamento attuale e l'offset di partenza (differisce nel caso sia una funzione o un blocco semplice). Ogni volta che uno scope viene creato il nesting level aumenta e una nuova, pulita, HashMap viene aggiunta in coda alla lista di Symbol Table accendendovi in maniera LIFO (Last In First Out). Ogni elemento di questo array rappresenta quindi una Symbol Table per quel nesting level dove una Symbol Table a sua volta rappresenta una mappa **String** → **STentry** dove la stringa è l'identificativo della variabile in questione. I principali compiti svolti da questa classe sono quelli di **Push & Pop** di uno scope e la ricerca di una variabile tramite il metodo **lookUp(id)**. Questi metodi vengono utilizzati nella sezione successiva per raccogliere eventuali errori semantici e farli presente all'utente alla fine della compilazione.