Inheritance graph

In this lab we want to record the class inheritance relationship, or when one class extends another class.

```
class Bar() extends Foo() { }

class Baz() extends Foo() { }
```

Naive approach

Use a map where key is `Bar` , `Baz` and value is `Foo`

```
var table: Map[Symbol, Symbol] = Map()

override def visit_class_decl (cd: Cclass_decl, name: Symbol, parent: Symbol, features: Features, filename: Symbol) {

    // Add a key value pair to the map
    table += (name -> parent)
}
```

To get classes that extend class `Foo` , i.e. `Bar` & `Baz` we need to invert the map

```
val inverted: Map[String, List[String]] = table
    .groupBy(_._2)                         // GroupBy value
    .map(x => x._1 -> x._2.values.toList)   // _1 is the old value, _2 is the is keys
    .toMap                                  // Create a map
```

How to use tables

- To get class that `Bar` extends: `table.getOrElse('Bar, 'Any)`

- To get classes that extend `Foo` : `inverted.getOrElse('Foo, List())`