Exercise 4: Object and Structure Nodes

Why is there both an object node and a structure node for named types such as `int` or `Person`? What information is stored in the object node and what information is stored in the structure node?

Solution

Object nodes store information about *declared names*, for example the object kind (constant, type, variable or method) and other information associated with this name (e.g., for constants their value, for variables their address and their declaration level).

Structure nodes, on the other hand, store information about the *structure of a type*, for example the element type for arrays and the fields for classes. For a class

```
class Person {
   char[] name;
   int age;
}
```

the object node contains the type name (`"Person"`) as well as a pointer to the corresponding structure node. The structure node stores the kind and structure of the type, for example that it is a class, that it has two fields, and finally the list of fields.

The object node is needed to be able to look up the type name in the symbol table. The structure node is needed to store the type's structure. As there are also unnamed types (e.g. `int[]`), we need structure nodes and cannot simply store the type information in the object node of the type name