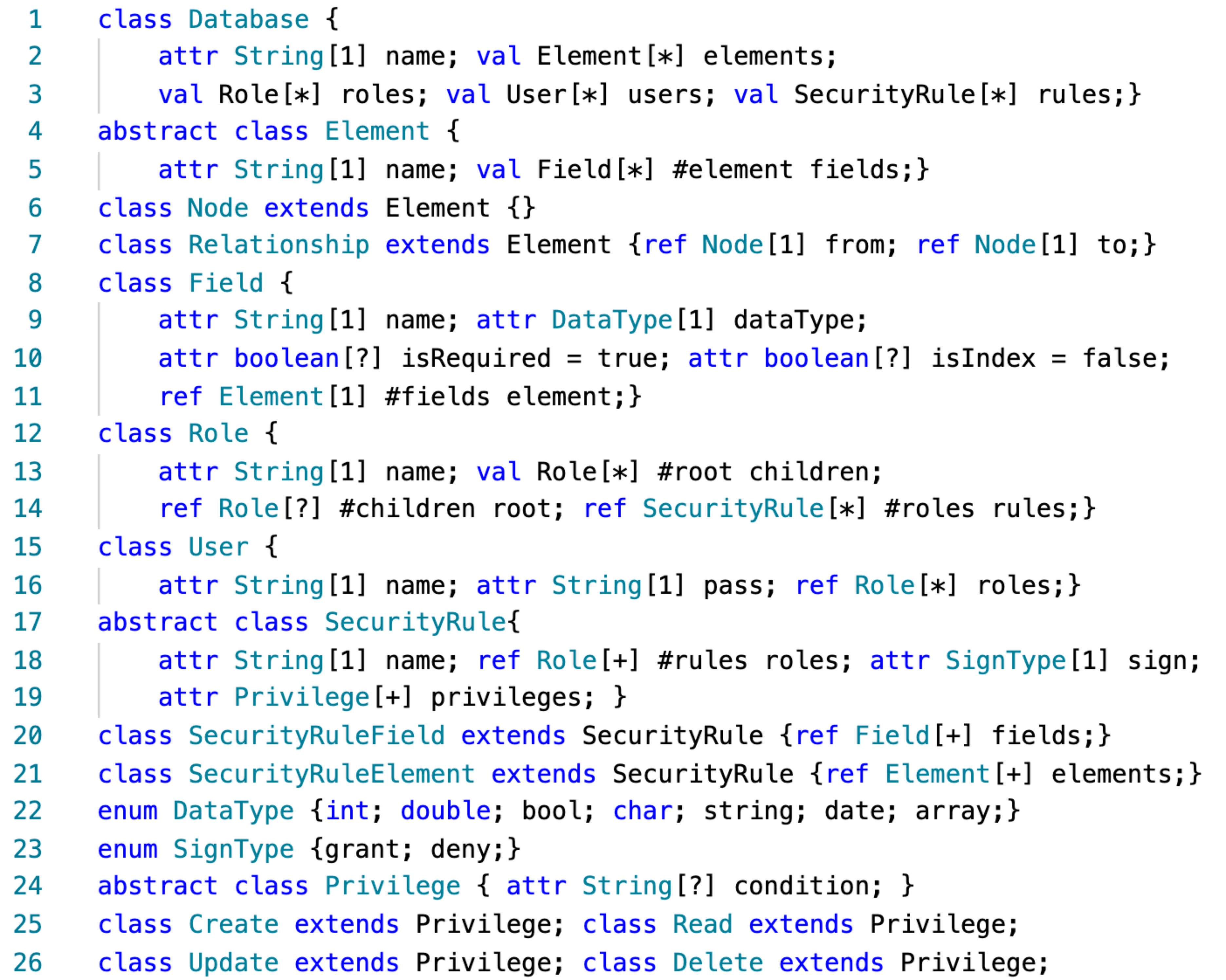

```
class Database {
    attr String[1] name; val Element[*] elements;
    val Role[*] roles; val User[*] users; val SecurityRule[*] rules;}
abstract class Element {
    attr String[1] name; val Field[*] #element fields;}
class Node extends Element {}
class Relationship extends Element {ref Node[1] from; ref Node[1] to;}
class Field {
    attr String[1] name; attr DataType[1] dataType;
    attr boolean[?] isRequired = true; attr boolean[?] isIndex = false;
    ref Element[1] #fields element;}
class Role {
    attr String[1] name; val Role[*] #root children;
    ref Role[?] #children root; ref SecurityRule[*] #roles rules;}
class User {
    attr String[1] name; attr String[1] pass; ref Role[*] roles;}
abstract class SecurityRule{
    attr String[1] name; ref Role[+] #rules roles; attr SignType[1] sign;
    attr Privilege[+] privileges; }
class SecurityRuleField extends SecurityRule {ref Field[+] fields;}
class SecurityRuleElement extends SecurityRule {ref Element[+] elements;}
enum DataType {int; double; bool; char; string; date; array;}
enum SignType {grant; deny;}
abstract class Privilege { attr String[?] condition; }
class Create extends Privilege; class Read extends Privilege;
class Update extends Privilege; class Delete extends Privilege;
```