

```

//*****
// File: Symbol.java
// Author: Procesadores de Lenguajes-University of Zaragoza
// Date: julio 2021
// Coms: Atributos públicos para evitar el uso de getters y setters
//       Dado que los símbolos se almacenarán en un diccionario, nunca
//       se accederá a ellos sin conocer su nombre, por lo que el atributo
//       nombre es innecesario
//*****

package lib.symbolTable;

abstract public class Symbol implements Cloneable {
    static public enum ParameterClass {
        VAL, REF, NONE
    }

    static public enum Types {
        INT, BOOL, CHAR, ARRAY, FUNCTION, PROCEDURE, STRING, UNDEFINED
    }

    public String name;
    // será NONE para no parámetros
    public ParameterClass parClass;
    // dirección en memoria. Para func/proc, dirección de la primera instrucción
    public long dir;
    // mi tipo
    public Types type;
    // es constante?
    public boolean constant;
    public int nivel; // nivel dentro de la TS
    public int numVar;

    public Symbol(String _name, Types _type) {
        this(_name, -1, _type, ParameterClass.NONE, false);
    }

    public Symbol(String _name, Types _type, ParameterClass _parClass) {
        this(_name, -1, _type, _parClass, false);
    }

    public Symbol(String _name, long _dir, Types _type, ParameterClass _parClass, boolean _constant) {
        name = _name;
        dir = _dir;
        type = _type;
        parClass = _parClass;
        constant = _constant;
    }

    public Symbol clone() {
        try {
            return (Symbol) super.clone();
        } catch (CloneNotSupportedException e) {
            return null;
        }
    }

    abstract public String toString();
}

```