

```

//*****
// File: SymbolFunction.java
// Author: Procesadores de Lenguajes-University of Zaragoza
// Date: julio 2021
// Coms: Atributos públicos para evitar el uso de getters y setters
//*****

package lib.symbolTable;
import java.util.*;
import java.util.AbstractMap.SimpleEntry;

public class SymbolFunction extends Symbol implements Cloneable {

//cada parámetro es un par <id,símbolo> donde "símbolo" es una copia del valor
//introducido en la tabla de símbolos. Esta duplicación de la información es necesaria
//ya que cuando se cierra un bloque, tanto los parámetros como las variables locales
//se eliminan de la tabla, pero la propia función/proc sigue en la tabla, haciendo
//necesario mantener la información de los parámetros.
public ArrayList<Symbol> parList;

public Types returnType; //tipo de la función

public SymbolFunction(String _name, ArrayList<Symbol> _parList,
Types _returnType) {
super(_name, Types.FUNCTION, ParameterClass.NONE);
parList = _parList;
returnType = _returnType;
}

public String toString() {
return "(" + name + "," + type + "," + parList + "," + returnType + "," + parClass + "," + nivel + ")";
}

public SymbolFunction clone () {
SymbolFunction newSymbolFunction = (SymbolFunction) super.clone();
newSymbolFunction.parList = new ArrayList<Symbol> (parList);
return newSymbolFunction;
}
}

```