

```
//*****  
// File: SymbolProcedure.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 SymbolProcedure 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 boolean principal;
```

```
public SymbolProcedure(String _name, ArrayList<Symbol> _parList) {  
    super(_name, Types.PROCEDURE, ParameterClass.NONE);  
    parList = _parList;  
    principal = false;  
}
```

```
public SymbolProcedure(String _name, ArrayList<Symbol> _parList, boolean _principal) {  
    super(_name, Types.PROCEDURE, ParameterClass.NONE);  
    parList = _parList;  
    principal = _principal;  
}
```

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

```
public SymbolProcedure clone() {  
    SymbolProcedure newSymbolProcedure = (SymbolProcedure) super.clone();  
    newSymbolProcedure.parList = new ArrayList<Symbol>(parList);  
    return newSymbolProcedure;  
}
```

```
}
```