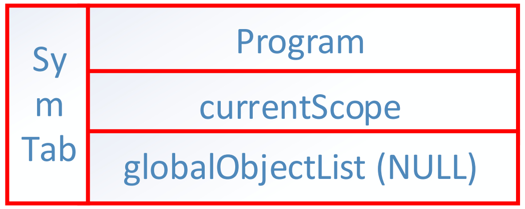
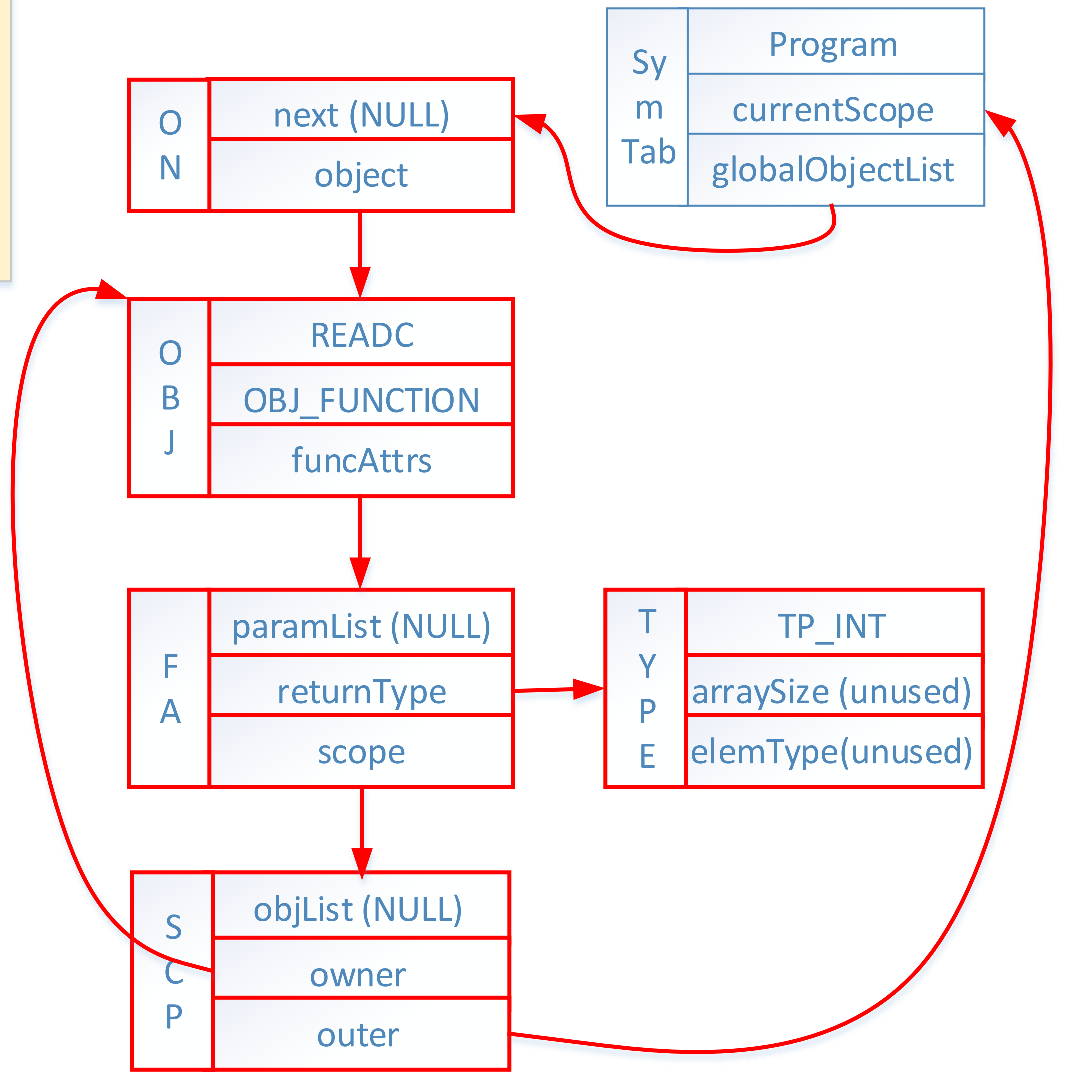
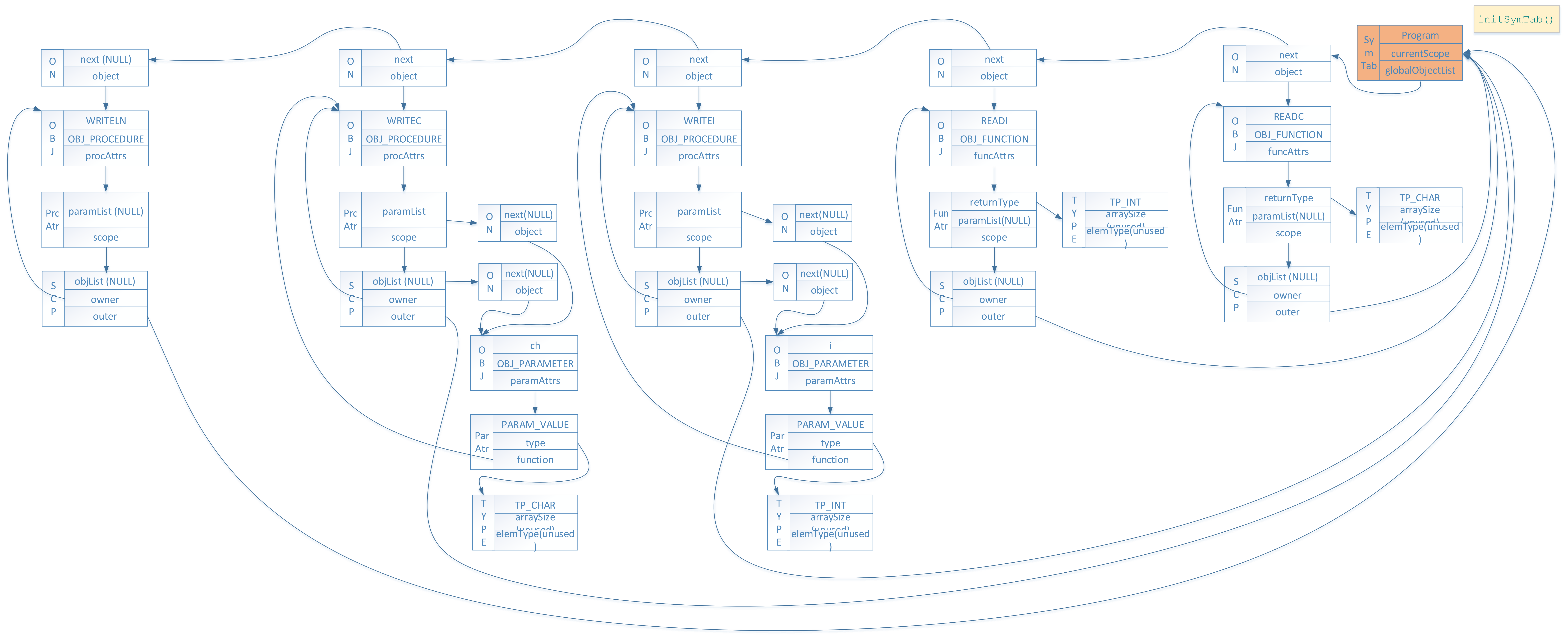
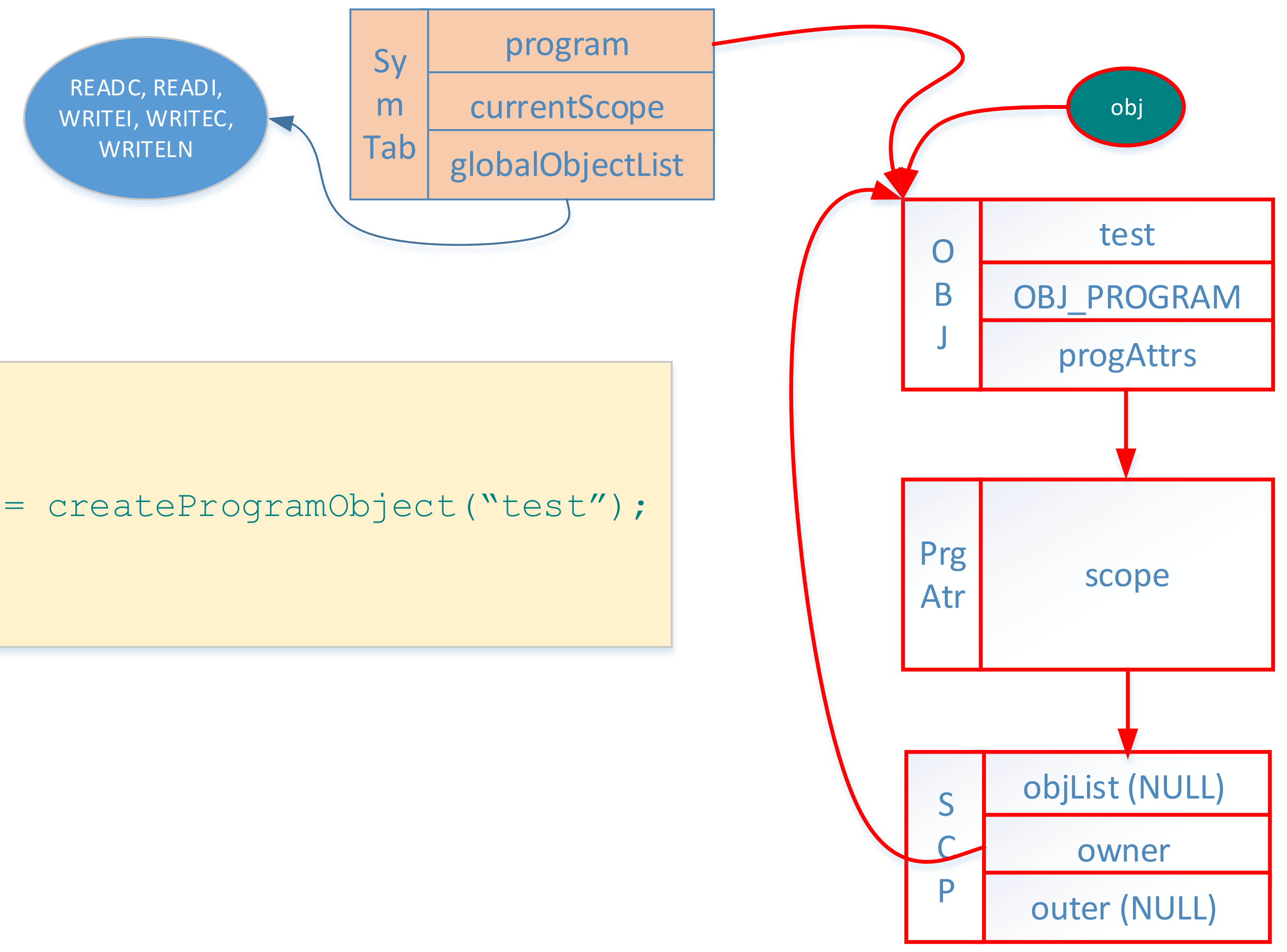
```
initSymTab(){
...
  symtab = (SymTab*) malloc(sizeof(SymTab));
  symtab->globalObjectList = NULL;
...
}
```



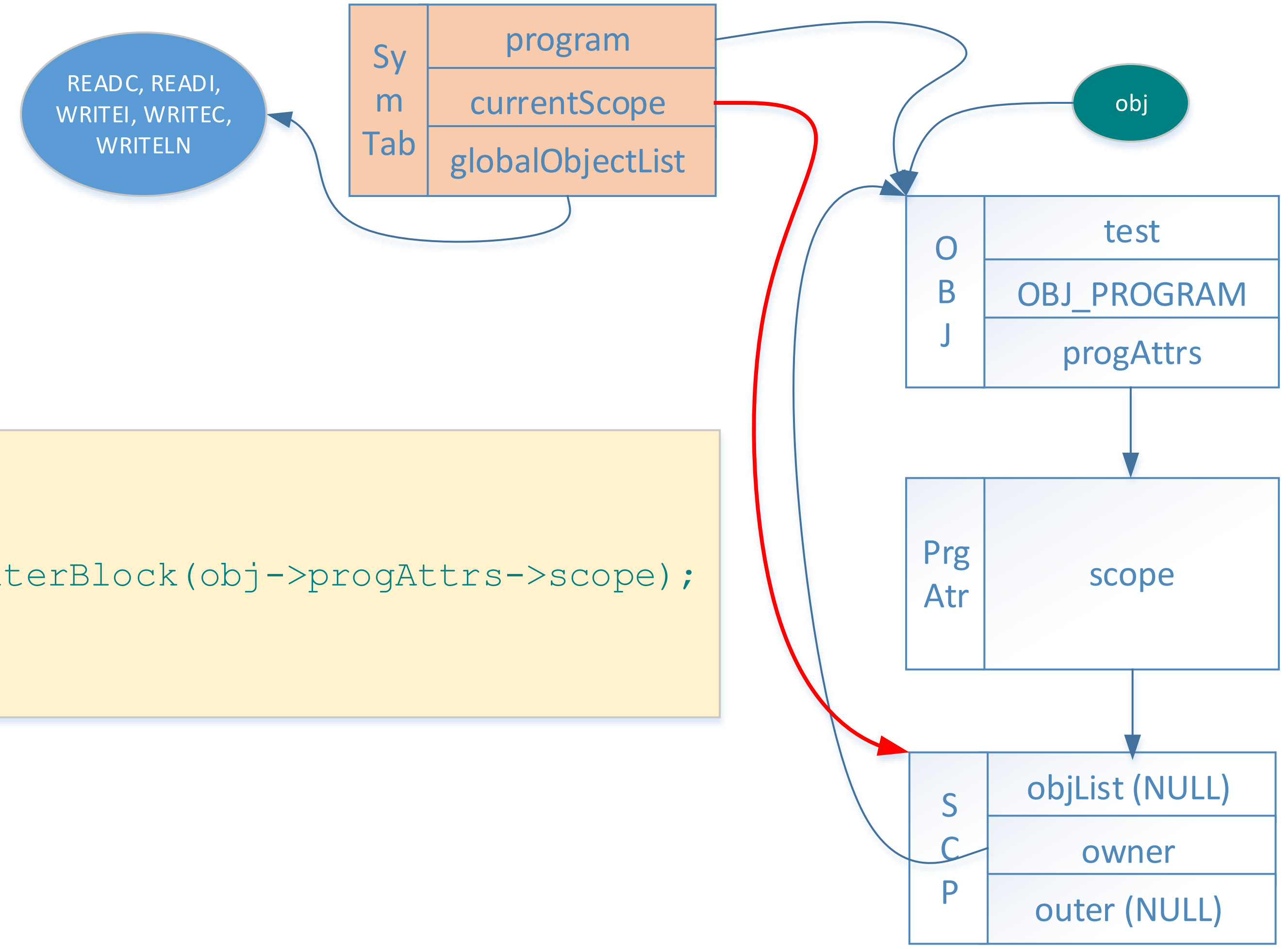
```
initSymTab(){
...
  obj = createFunctionObject("READC");
  obj->funcAttrs->returnType = makeCharType();
  addObject(&(symtab->globalObjectList), obj);
...
}
```



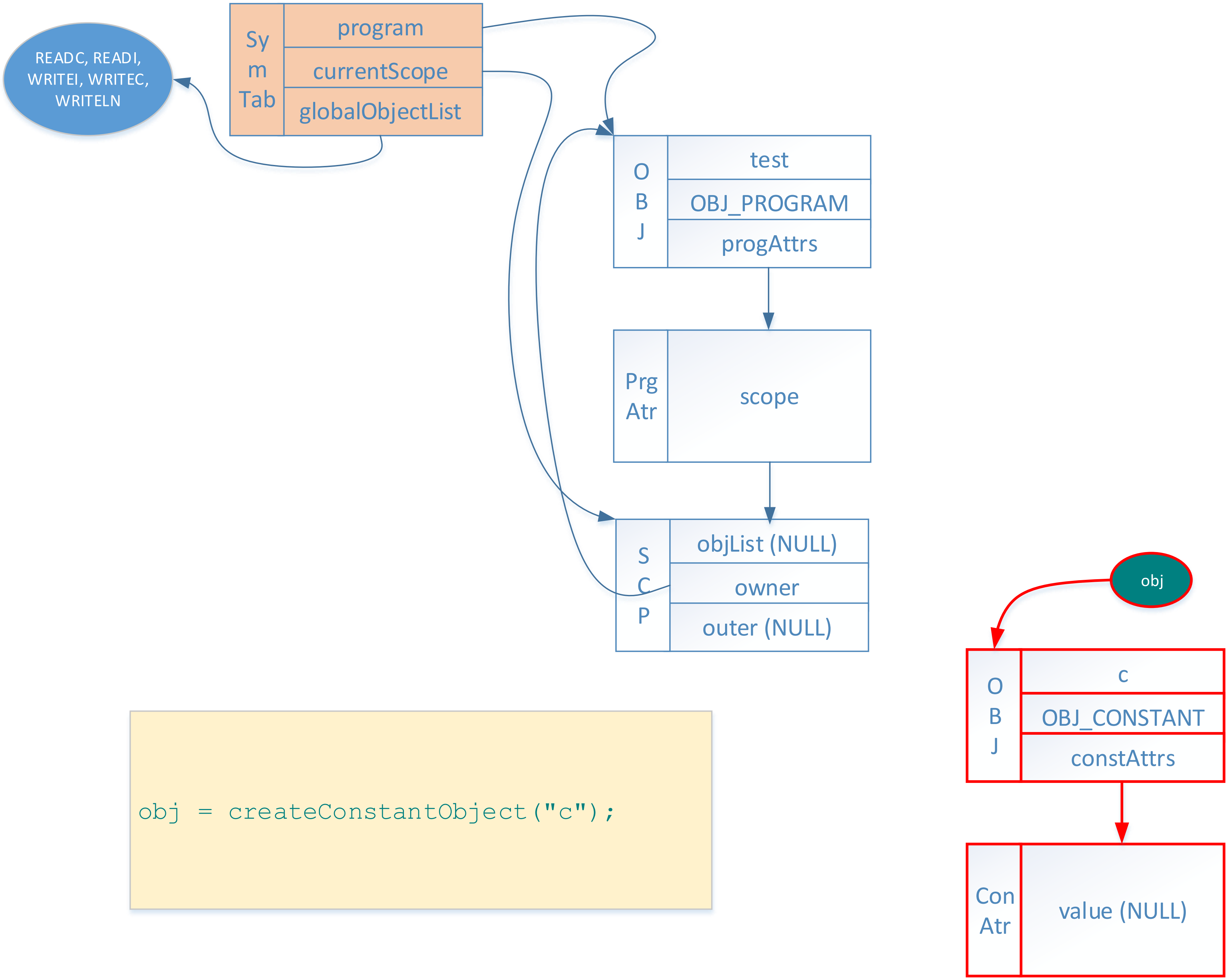

initSymTab()
Sy m Tab
Program
currentScope
globalObjectList
O N
next (NULL)
object
O B J
WRITELN
OBJ_PROCEDURE
procAttrs
Prc Atr
paramList (NULL)
scope
S C P
objList (NULL)
owner
outer
O N
next
object
O B J
WRITEC
OBJ_PROCEDURE
procAttrs
Prc Atr
paramList
scope
O N
next(NULL)
object
S C P
objList (NULL)
owner
outer
O N
next(NULL)
object
O B J
ch
OBJ_PARAMETER
paramAttrs
Par Atr
PARAM_VALUE
type
function
T Y P E
TP_CHAR
arraySize (unused)
elemType(unused)
O N
next
object
O B J
WRITEI
OBJ_PROCEDURE
procAttrs
Prc Atr
paramList
scope
O N
next(NULL)
object
S C P
objList (NULL)
owner
outer
O N
next(NULL)
object
O B J
i
OBJ_PARAMETER
paramAttrs
Par Atr
PARAM_VALUE
type
function
T Y P E
TP_INT
arraySize (unused)
elemType(unused)
O N
next
object
O B J
READI
OBJ_FUNCTION
funcAttrs
Fun Atr
returnType
paramList(NULL)
scope
T Y P E
TP_INT
arraySize (unused)
elemType(unused)
S C P
objList (NULL)
owner
outer
O N
next
object
O B J
READC
OBJ_FUNCTION
funcAttrs
Fun Atr
returnType
paramList(NULL)
scope
T Y P E
TP_CHAR
arraySize (unused)
elemType(unused)
S C P
objList (NULL)
owner
outer



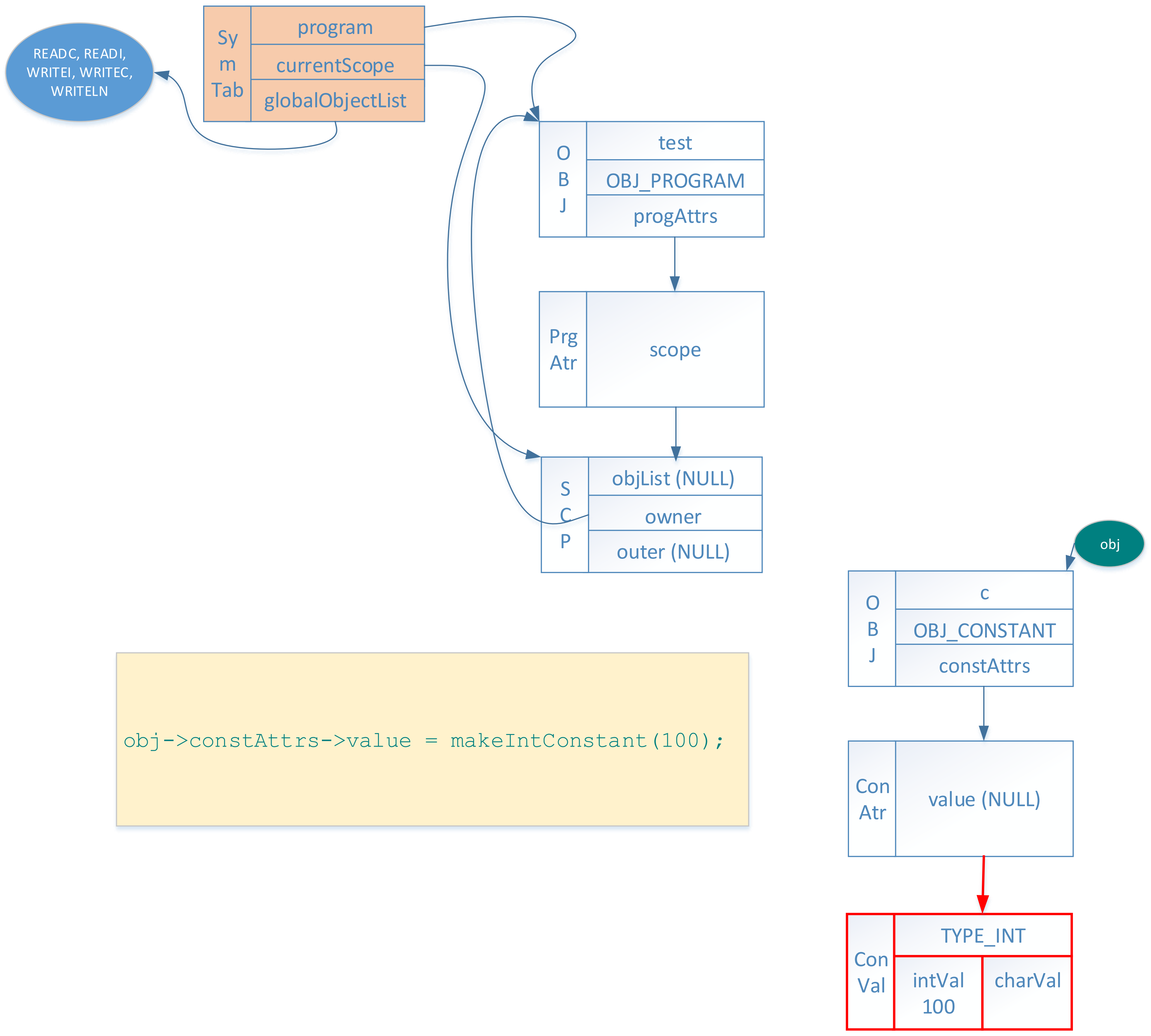
```
obj = createProgramObject("test");
```



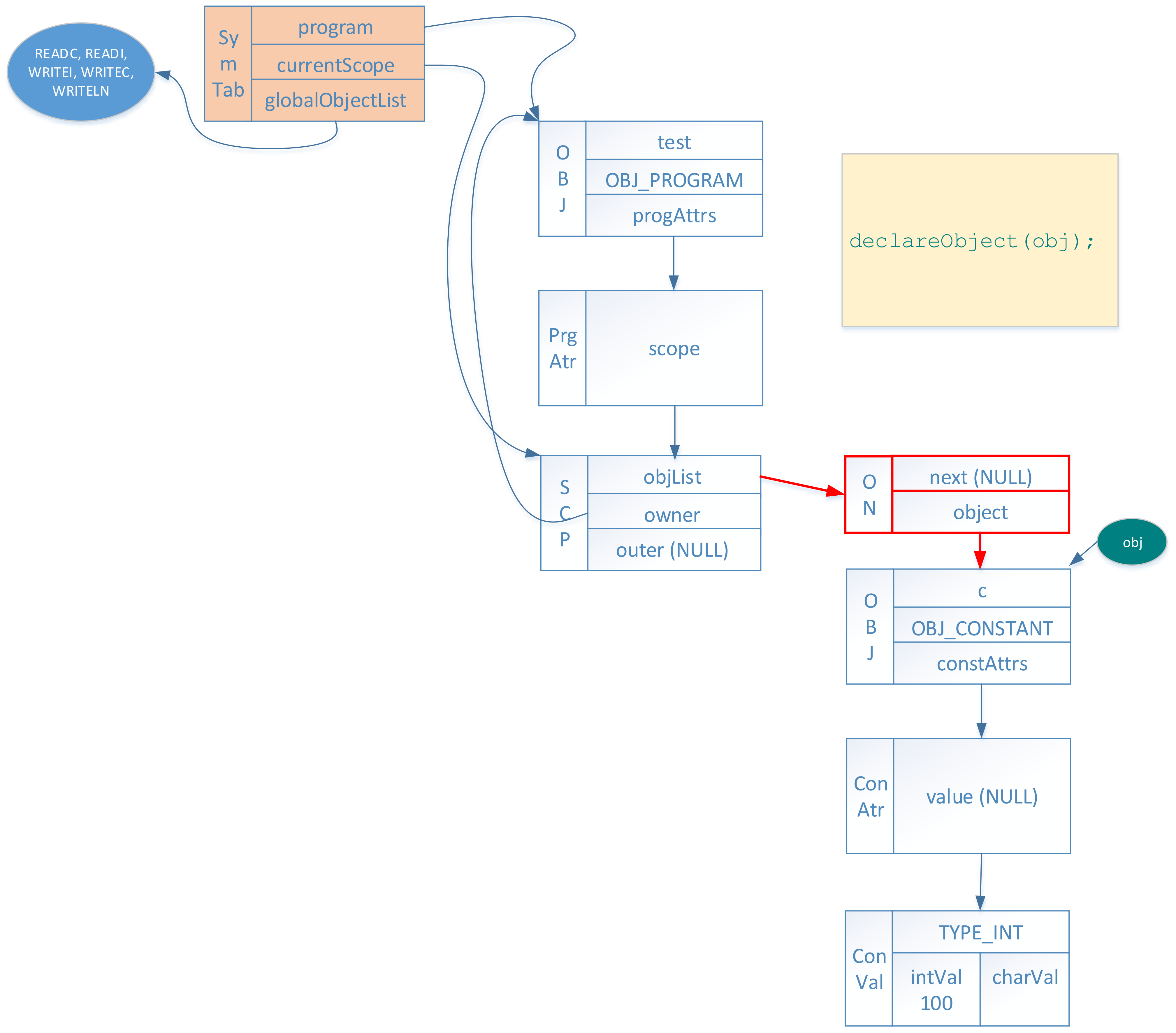
```
enterBlock(obj->progAttrs->scope);
```



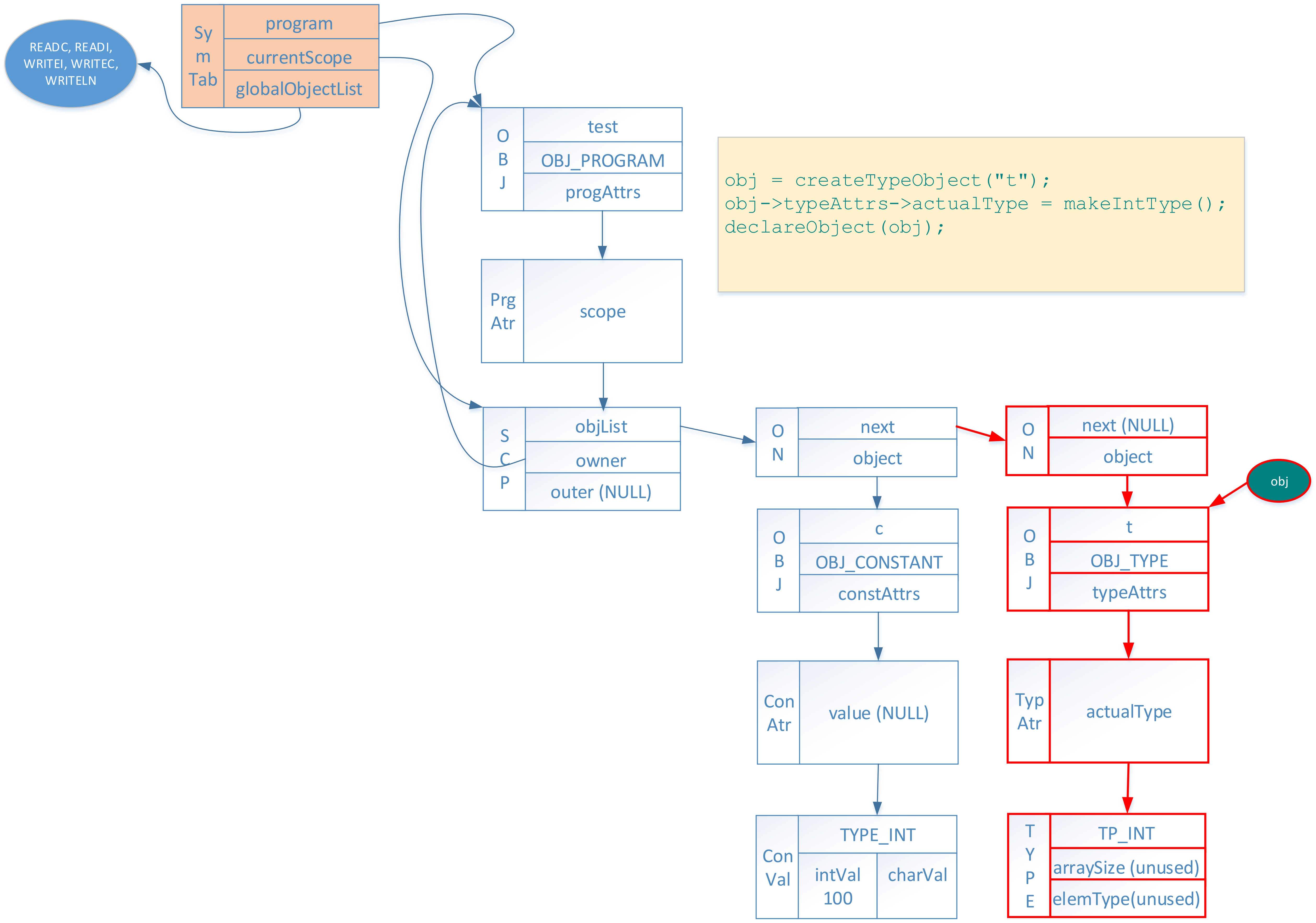
```
obj = createConstantObject("c");
```



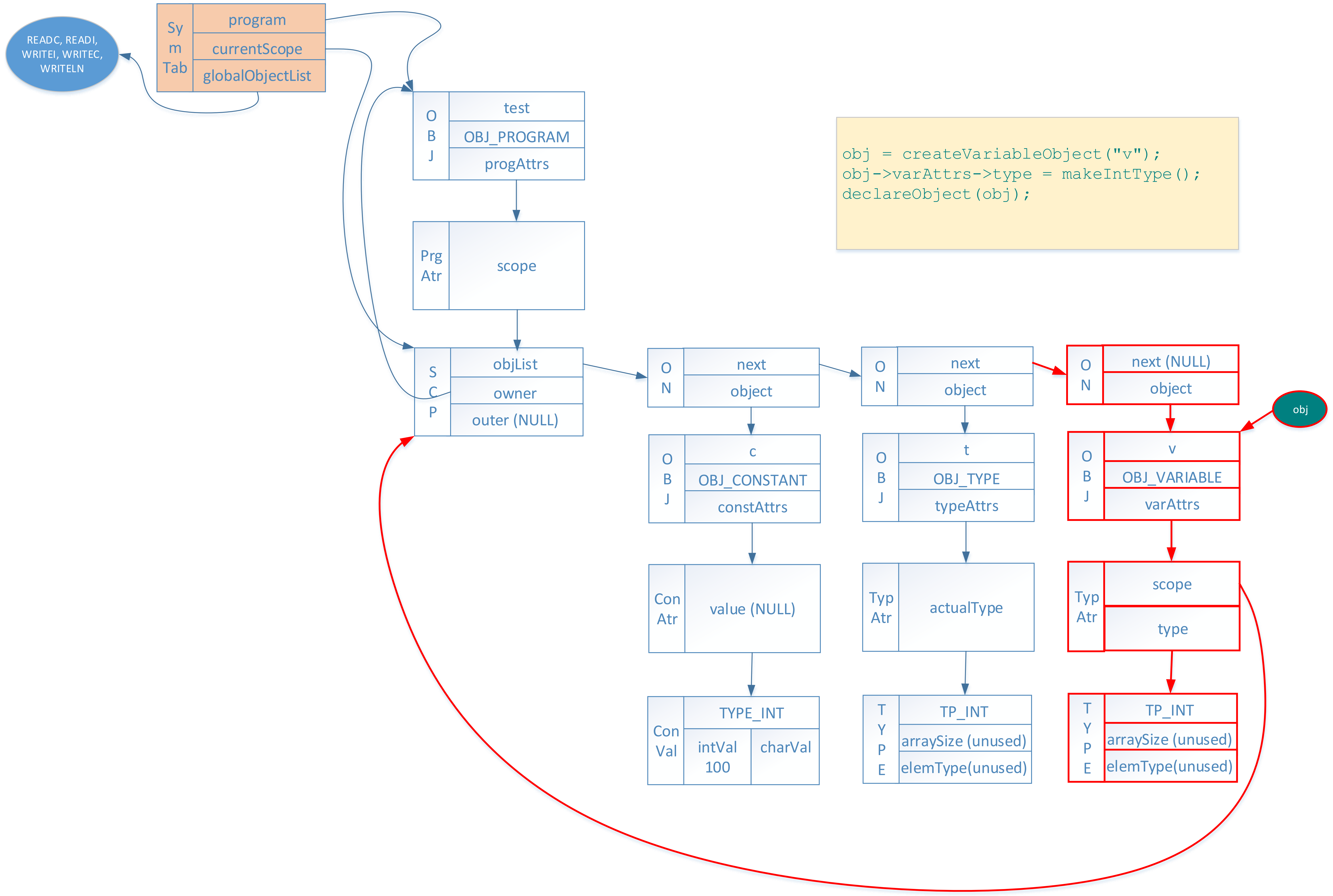
```
obj->constAttrs->value = makeIntConstant(100);
```



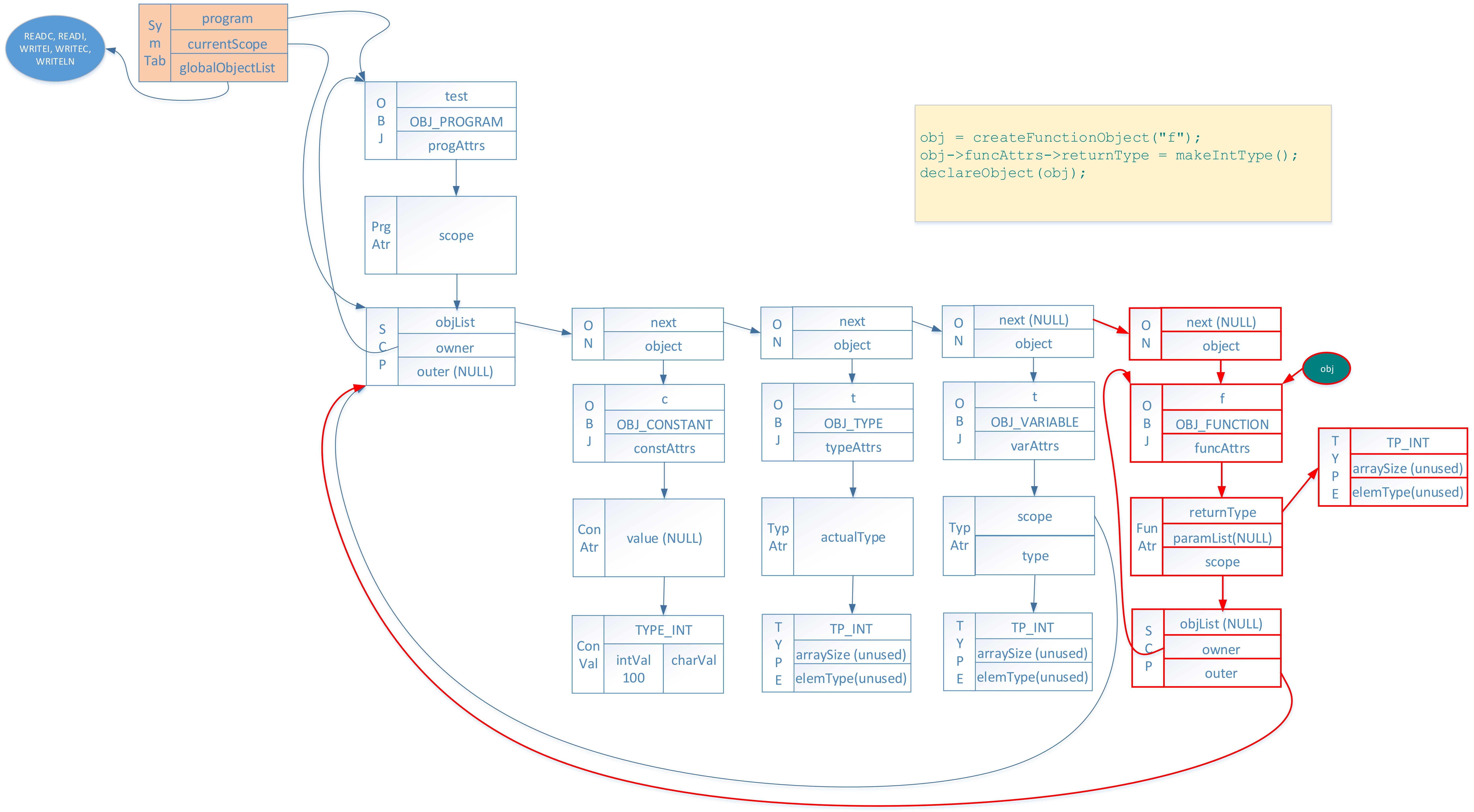
```
declareObject(obj);
```



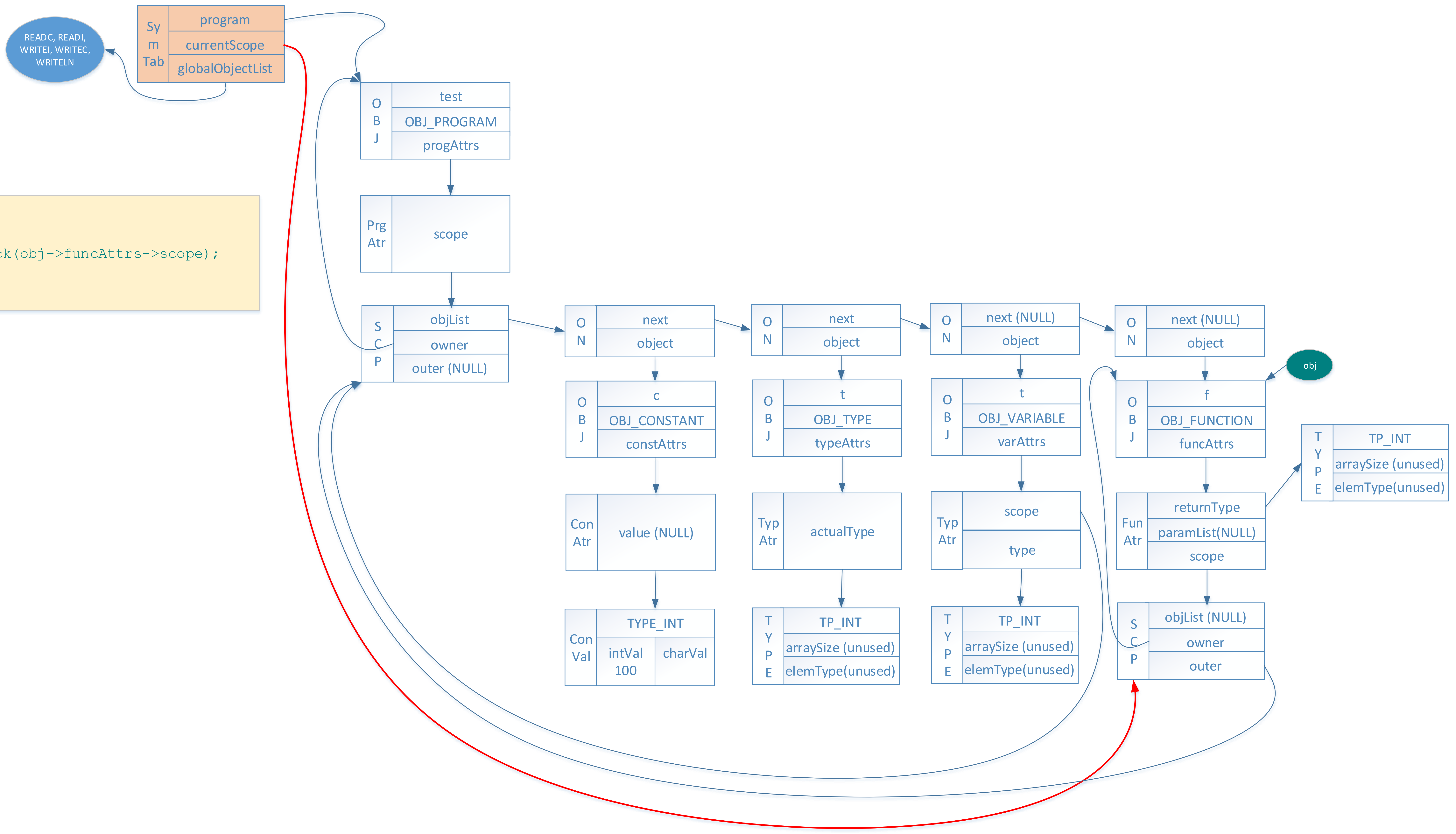
```
obj = createTypeObject("t");
obj->typeAttrs->actualType = makeIntType();
declareObject(obj);
```



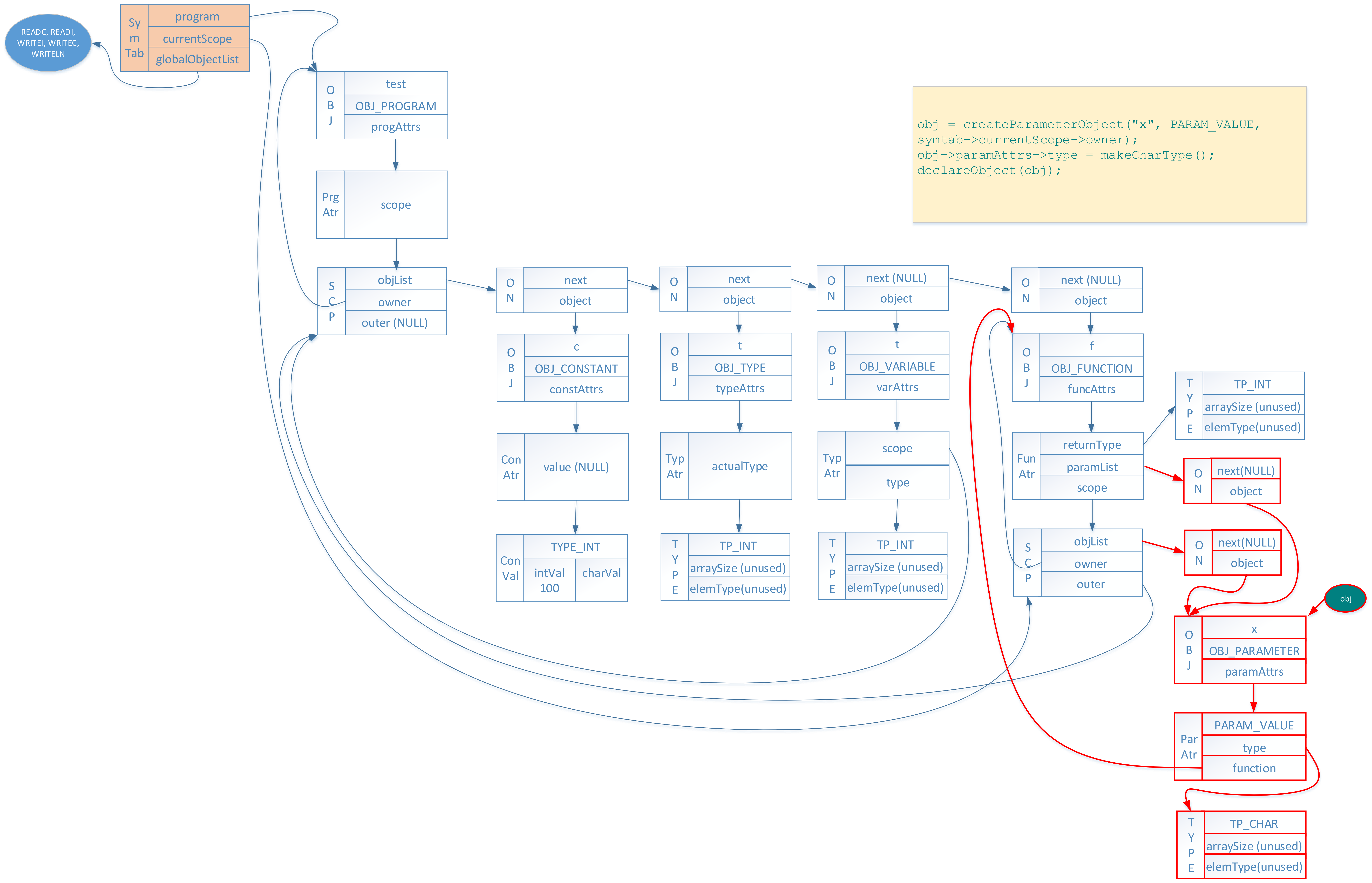
```
obj = createVariableObject("v");
obj->varAttrs->type = makeIntType();
declareObject(obj);
```



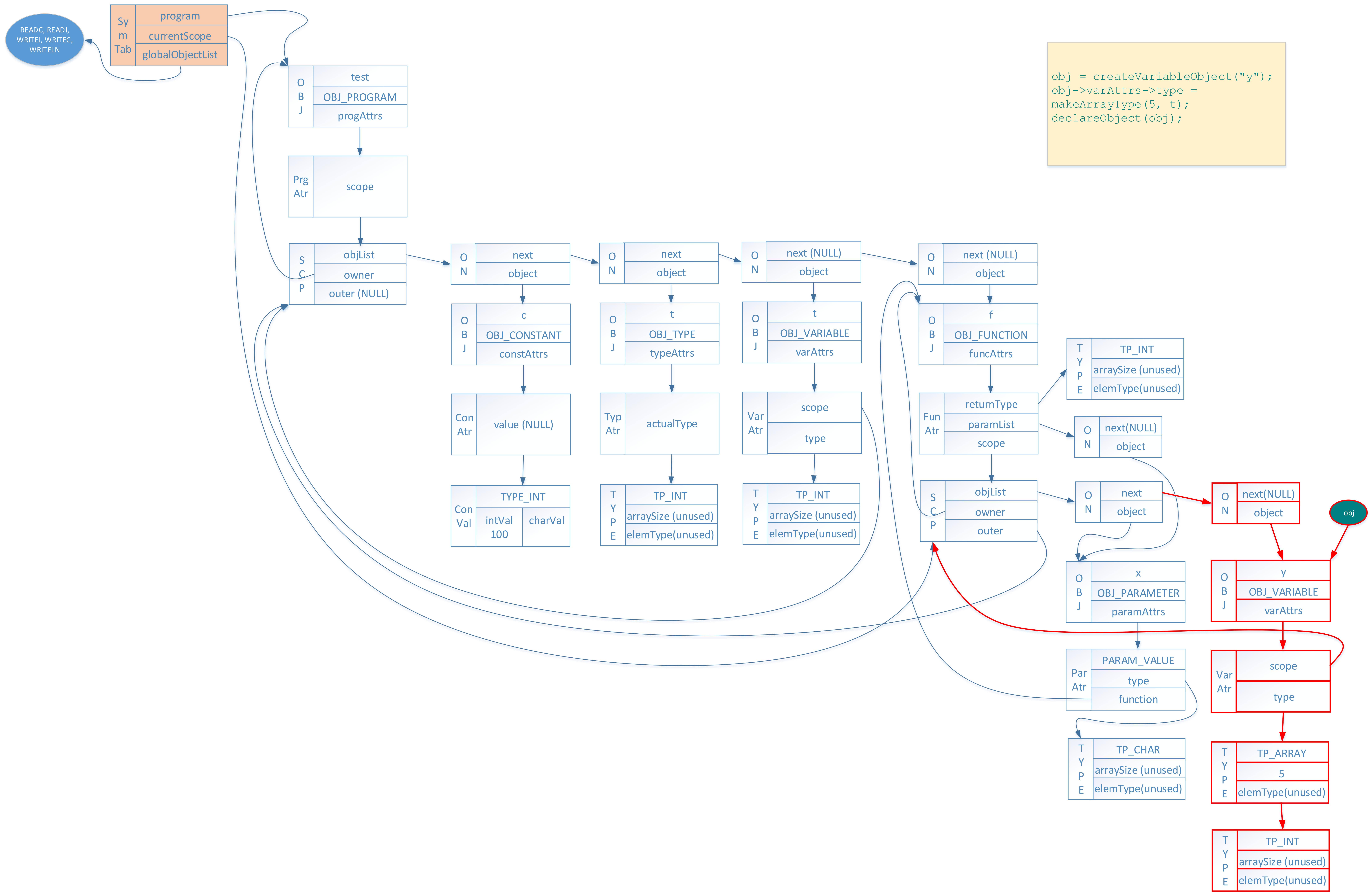
```
obj = createFunctionObject("f");
obj->funcAttrs->returnType = makeIntType();
declareObject(obj);
```



```
enterBlock(obj->funcAttrs->scope);
```

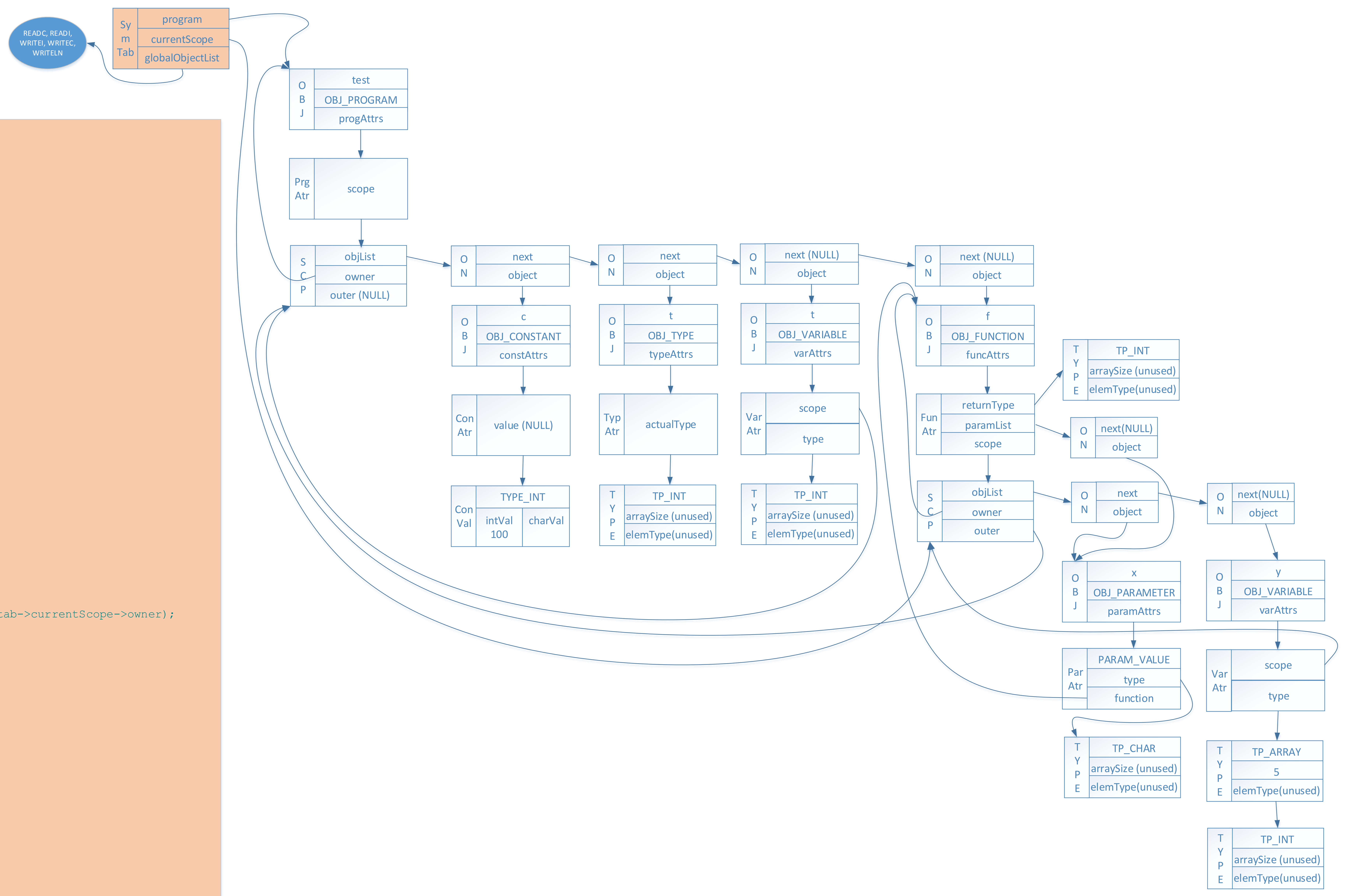


```
obj = createParameterObject("x", PARAM_VALUE,
symtab->currentScope->owner);
obj->paramAttrs->type = makeCharType();
declareObject(obj);
```



```
obj = createVariableObject("y");
obj->varAttrs->type =
makeArrayType(5, t);
declareObject(obj);
```

```c
int main() {
  Object* obj;

  initSymTab();

  obj = createProgramObject("test");
  enterBlock(obj->progAttrs->scope);

  obj = createConstantObject("c");
  obj->constAttrs->value = makeIntConstant(100);
  declareObject(obj);

  obj = createTypeObject("t");
  obj->typeAttrs->actualType = makeIntType();
  declareObject(obj);

  obj = createVariableObject("v");
  obj->varAttrs->type = makeIntType();
  declareObject(obj);

  obj = createFunctionObject("f");
  obj->funcAttrs->returnType = makeIntType();
  declareObject(obj);

    enterBlock(obj->funcAttrs->scope);

    obj = createParameterObject("x", PARAM_VALUE, symtab->currentScope->owner);
    obj->paramAttrs->type = makeCharType();
    declareObject(obj);

    obj = createVariableObject("y");
    obj->varAttrs->type = makeArrayType(5, t);
    declareObject(obj);

    exitBlock();

  exitBlock();
  printObject(symtab->program,0);
  cleanSymTab();

  return 0;
}
```


READC, READI, WRITEI, WRITEC, WRITELN
Sym Tab
program
currentScope
globalObjectList
OBJ
test
OBJ_PROGRAM
progAttrs
Prg Atr
scope
SCP
objList
owner
outer (NULL)
ON
next
object
c
OBJ_CONSTANT
constAttrs
Con Atr
value (NULL)
Con Val
TYPE_INT
intVal 100
charVal
t
OBJ_TYPE
typeAttrs
Typ Atr
actualType
TYPE
TP_INT
arraySize (unused)
elemType(unused)
next (NULL)
OBJ_VARIABLE
varAttrs
Var Atr
type
f
OBJ_FUNCTION
funcAttrs
Fun Atr
returnType
paramList
x
OBJ_PARAMETER
paramAttrs
Par Atr
PARAM_VALUE
function
TP_CHAR
y
TP_ARRAY
5