

```
private static void MethodDecl() {
    Struct type = Type();
    check(ident);
    String mName = t.string;
    Obj meth = Tab.insert(Obj.Meth, mName, type);
    meth.val = Code.pc;
    Tab.openScope();
    check(lpar);
    if (sym == ident) nPars = FormPars();
    check(rpar);
    while (sym == ident) VarDecl();
    ...
    if (mName.equals("main")) {
        Code.mainpc = Code.pc;
        if (meth.type != Tab.noType)
            semError("main method must be void");
        if (nPars != 0)
            semError("main method must not have parameters");
    }
    Code.put(Code.enter);
    Code.put(nPars);
    Code.put(Tab.topScope.nVars);
    ...
    if (meth.type == Tab.noType) {
        Code.put(Code.exit);
        Code.put(Code.return_);
    } else {
        Code.put(Code.trap);
        Code.put(1);
    }
    Tab.closeScope();
}
```

```
private static void Statement(...) {
    Item x;
    switch (sym) {
        case Token.ident: // assignment or call
            x = Designator();
            switch (sym) {
                ...
                case Token.lpar:
                    ActParList(x);
                    Code.put(Code.call);
                    Code.put2(x.val - (Code.pc-1));
            }
        ...
    }
}

private static Item Factor() {
    Item x;
    if (sym == ident) {
        x = Designator();
        if (sym == lpar) {
            ActParList(x);
            if (x.obj == Tab.ordObj || x.obj == Tab.chrObj)
                ; // conversion done automatically
            else if (x.obj == Tab.lenObj)
                Code.put(Code.arraylength);
            else {
                Code.put(Code.call);
                Code.put2(x.val - (Code.pc-1));
            }
        }
        x.kind = Item.Stack;
    }
    ...
}
```