Modula-2 Muddles

Scott Hazelhurst
Department of Computer Science
University of the Witwatersrand
Wits
2050 South Africa
e-mail: 122scott.witsvma@f4.n494.z5.fidonet.org

Dijkstra [1989] has pointed out that appealing to students' intuition is educationally unsound, criticizing those who "consciously try to link new material to what is supposed to be the student's familiar world". This maxim has important consequences for the programming language used in teaching programming. Previous work at my university has shown that Pascal is an excellent tool for ensuring that there is no link between a computer program and anything the student may be familiar with [Machanick *et al.* 1985]. Modula-2 offers much more than Pascal. Particularly, the use of variable names can be made as obscure as possible, severing all links with intuition. The two programs listed below (which compile on the Walterloo Modula-2 compiler) show that now we no longer have to rely on just the variables $i, j, k...$ which may over time gain some meaning in the student's mind. Instead, the case sensitive nature of Modula-2 implies that we can use variable such as *xxxx*, *xxxX*, *xxXx* and so on. In this way we can easily achieve our educational objectives.

References

Dijkstra, E. 1989. On the Cruelty of Really Teaching Computing Science. *SIGCSE Bulletin* 21(1), February, ppxxv–xxxix

Machanick, P., Levieux, C.P., and Dadswell, P. 1987. Pascal Perversions. *SIGPLAN Notices* 22(2), February, p. 6.

```
MODULE Example1;
 FROM InOut IMPORT WriteLn, WriteString;
 FROM RealInOut IMPORT ReadReal, WriteReal;

CONST NumberOfData = 5;
TYPE  NumberOfDataRange = CARDINAL[1..NumberOfData];
      DataType          = REAL;
      LargeDataType     = REAL;

PROCEDURE FindAverage(sum:LargeDataType;
                      number:NumberOfDataRange):DataType;
  BEGIN
    RETURN sum / (FLOAT(CARDINAL(number)));
  END FindAverage;

 PROCEDURE FindSum(VAR sum : LargeDataType;
                   VAR biggest, smallest:DataType);
   VAR n : NumberOfDataRange;
       number : DataType;
   BEGIN
      FOR n := 1 TO NumberOfData DO
        WriteString("Enter next datum: "); WriteLn;
        ReadReal(number);
        IF number > biggest THEN
          biggest := number;
        END;
        IF number < smallest THEN
          smallest := number;
        END;
        sum := sum + number;
      END;
    END FindSum;


  VAR  sum       : LargeDataType;
       big,small : DataType;
       average   : DataType;

 BEGIN
    sum  := 0.0;
    big  :=  -FLOAT(MAX(CARDINAL));
    small:= FLOAT(MAX(CARDINAL));
    FindSum(sum,big,small);
    average := FindAverage(sum,NumberOfData);
    WriteString('Answer 1 is '); WriteReal(average,5); WriteLn;
    WriteString('Answer 2 is '); WriteReal(big,5); WriteLn;
    WriteString('Answer 3 is '); WriteReal(small,5); WriteLn;
  END Example1.
```

```
MODULE Example2;
 FROM InOut IMPORT WriteLn, WriteString;
 FROM RealInOut IMPORT ReadReal, WriteReal;

CONST MOdULE = 5;
TYPE  MoDULE = CARDINAL[1..MOdULE];
      MODUlE          = REAL;
      MODuLE     = REAL;

PROCEDURE MoDuLE(MoDUlE: MODuLE;
                      mODuLE:MoDULE):MODUlE;
  BEGIN
    RETURN MoDUlE / (FLOAT(CARDINAL(mODuLE)));
  END MoDuLE;

 PROCEDURE MoDUlE(VAR MoDulE : MODuLE;
                   VAR mOdUlE, mOduLE:MODUlE);
   VAR mODuLE : MoDULE;
       mODUlE : MODUlE;
   BEGIN
      FOR mODuLE := 1 TO MOdULE DO
        WriteString("Enter next datum: "); WriteLn;
        ReadReal(mODUlE);
        IF mODUlE > mOdUlE THEN
          mOdUlE := mODUlE;
        END;
        IF mODUlE < mOduLE THEN
          mOduLE := mODUlE;
        END;
        MoDulE := MoDulE + mODUlE;
      END;
    END MoDUlE;


  VAR  mOdUlE       : MODuLE;
       MOduLE,MODulE : MODUlE;
       mOdulE    : MODUlE;

 BEGIN
    mOdUlE  := 0.0;
    MOduLE  :=  -FLOAT(MAX(CARDINAL));
    MODulE:= FLOAT(MAX(CARDINAL));
    MoDUlE(mOdUlE,MOduLE,MODulE);
    mOdulE := MoDuLE(mOdUlE,MOdULE);
    WriteString('Answer 1 is '); WriteReal(mOdulE,5); WriteLn;
    WriteString('Answer 2 is '); WriteReal(MOduLE,5); WriteLn;
    WriteString('Answer 3 is '); WriteReal(MODulE,5); WriteLn;
  END Example2.
```