§4.1-4.2: Procedures

devo

28 Sep 2005
CMPT14x
Dr. Sean Ho
Trinity Western University

Reminders:

1) journals in folder

Review of (3.4-3.8)

- Loops: WHILE, REPEAT
- Sentinel variables
- Loop counters
- Using mathematical closed forms instead of loops
- An example of problem-solving (postpone til Thu)
  - Stub program
  - Using ReadResult() to test the previous Read operation

What's on for today (4.1-4.2)

- Helper functions: ABS, CAP, INC, DEC
- Qualified IMPORT
- Procedures:
  - No parameters
  - Read-only parameters
  - Writeable parameters
  - Both kinds of parameters
  - Formal vs. actual parameters
  - Scope

Miscellaneous helper functions

- ABS: ABS (-5) = 5
- CAP: CAP (“a”) = “A”
- INC/DEC:
  - INC (counter); => counter := counter + 1;
  - INC (counter, 2); => counter := counter + 2;
- BOOLEAN type:

```
VAR
    M2IsCool : BOOLEAN;
BEGIN
    M2IsCool := TRUE;
```

Qualified IMPORT

- Import individual functions from a library:

```
FROM STextIO IMPORT
    WriteString, WriteLn;
```

- Or import an entire library:

```
IMPORT STextIO;
BEGIN
    STextIO.WriteString ("Hello World!");
    STextIO.WriteLn;
```

- Qualified identifier: STextIO.WriteString
- Unqualified identifier: WriteString

Procedures

- Fourth program structure/flow abstraction is composition
- This is implemented in Modula-2 using
  - Procedures
  - Function procedures
  - Recursion
- A procedure is a chunk of code doing a sub-task
  - Written once, can be used many times
- We've already been using procedures:
  - WriteString, ReadCard, SkipLine, etc.

Procedure input and output

- Procedures can do the same thing every time:
  - WriteLn;
- Or rely upon input data:
  - WriteString (“Hello World”);
- Or produce/alter a variable:
  - ReadReal (width);
- Or both input and output data:
  - INC (counter, 2);
- The variables in parentheses are called parameters (or arguments) to the procedure

Example: no parameters

- Procedure to print program usage info:

```
MODULE SphereVolume;
IMPORT STextIO, SRealIO;

PROCEDURE PrintUsage;
BEGIN
    STextIO.WriteString ("This program calculates the volume ");
    STextIO.WriteString ("of a sphere, given its radius.");
    STextIO.WriteLn;
END PrintUsage;

BEGIN
    PrintUsage;
```

Scope

- Procedures inherit declarations from enclosing procedures/modules:
  - Didn't need to re-IMPORT STextIO
  - Declarations:
    - IMPORT
    - VAR, CONST
    - Other procedures

Example: read-only parameters

```
MODULE SphereVolume;
IMPORT STextIO, SRealIO;

PROCEDURE PrintSphereVol (volume : REAL);
BEGIN
    STextIO.WriteString ("Your sphere has a volume of");
    SRealIO.WriteFixed (volume, 2, 0);
    STextIO.WriteLn;
END PrintSphereVol;

BEGIN
    PrintSphereVol (5.0);
END SphereVolume.
```

formal parameter

actual parameter

Example: writeable parameters

```
MODULE SphereVolume;
IMPORT STextIO, SRealIO;

PROCEDURE GetRadius (VAR userInput : REAL);
BEGIN
    STextIO.WriteString ("What is the radius of your sphere? ");
    SRealIO.ReadReal (userInput);
END GetRadius;

VAR
    radius : REAL;
BEGIN
    GetRadius (radius);
```

Can access both userInput, radius

Can access only radius

Example: both types of params

```
MODULE SphereVolume;
IMPORT STextIO, SRealIO;

PROCEDURE CalcVol (radius: REAL; VAR volume : REAL);
CONST
    Pi = 3.14159265358979323846;
BEGIN
    volume := (4.0 / 3.0) * Pi * radius * radius * radius;
END CalcVol;

VAR vol : REAL;
BEGIN
    CalcVol (5.0, vol);
```

Can access radius, volume, Pi, vol

Can access only vol

More on scope

```
VAR global1, global2, global3: REAL;
PROCEDURE Proc1 (VAR param1, param2 : REAL);
BEGIN
    param2 := param1 * global3;
END Proc1;

BEGIN
    global1 := 2.0;
    global3 := 3.0;
    Proc1 (global1, global2);
    global3 := 5.0;
    Proc1 (global1, global2);
```

Global variable: poor design!

global2 = 6.0

global2 = 10.0

Review of today (4.1-4.2)

- Helper functions: ABS, CAP, INC, DEC
- Qualified IMPORT
- Procedures:
  - No parameters
  - Read-only parameters
  - Writeable parameters
  - Both kinds of parameters
  - Formal vs. actual parameters
  - Scope

TODO items

- Homework: §3.14 #17, 36 (hand in on Fri)
- Lab3 next week: §4.11 #(33 / 34 / 41)
- Reading: through §4.7 for Fri