§6.1–6.4: Standard Libraries: I/O and Math

- *devo*

14 Oct 2005
CMPT14x
Dr. Sean Ho
Trinity Western University

Reminders:

- ***journals** in folder*
- ***hw** due today*
- ***quiz** ch5 today*

Review of (5.4–5.8)

- FOR loops
  - Loop control variable
    - Needs initialization?
    - Value after the loop?
  - FOR vs. WHILE: pros/cons?
- Arrays as procedure parameters
  - Type compatibility for value/variable params
  - Open arrays
    - HIGH
- Multidimensional arrays

Quiz ch5 (4 questions, 20 marks, 10 minutes)

```
TYPE
    DayName = (Sun, Mon, Tue, Wed, Thu, Fri, Sat);
    WeekdayName = [Mon .. Fri];
VAR today : WeekdayName;
BEGIN today := Mon;
```

- "OK" or "not ok" (legal or illegal):

```
INC (today);                    DEC (today);
today := today + 1;             IF (today < Thu) ...
```

- Declare a string type named String20, of length 20.
- Declare an array named TeamLead that stores one string of type String20 for each day of the week.
- Write a for loop that capitalizes the first letter of each string in TeamLead (making changes to the TeamLead array).

Quiz ch5 answers: #1

```
TYPE
    DayName = (Sun, Mon, Tue, Wed, Thu, Fri, Sat);
    WeekdayName = [Mon .. Fri];
VAR
    today : WeekdayName;
BEGIN
    today := Mon;
```

- "OK" or "not ok" (legal or illegal):

INC (today);	ok!	DEC (today);	not ok!
today := today + 1;	not ok!	IF (today < Thu) ...	ok!

Quiz ch5 answers: #2-4

- Declare a string type named String20, of length 20.
  - TYPE

    String20 = ARRAY [1 .. 20] OF CHAR;
- Declare an array named TeamLead that stores one string of type String20 for each day of the week.
  - VAR

    TeamLead : ARRAY WeekdayName OF String20;
- Write a for loop that capitalizes the first letter of each string in TeamLead (making changes to the TeamLead array).
  - FOR today := Mon TO Fri

    DO

    TeamLead [today][1] := CAP (TeamLead [today][1]);

    END;

What's on for today (6.1–6.4)

- ISO Modula-2 Standard Library modules: I/O
- Some new functions in STextIO:
  - ReadRestLine, ReadToken
- I/O channels
- Redirection
  - TWU library RedirStdIO
- Standard Library module: RealMath

M2 Standard Library Modules

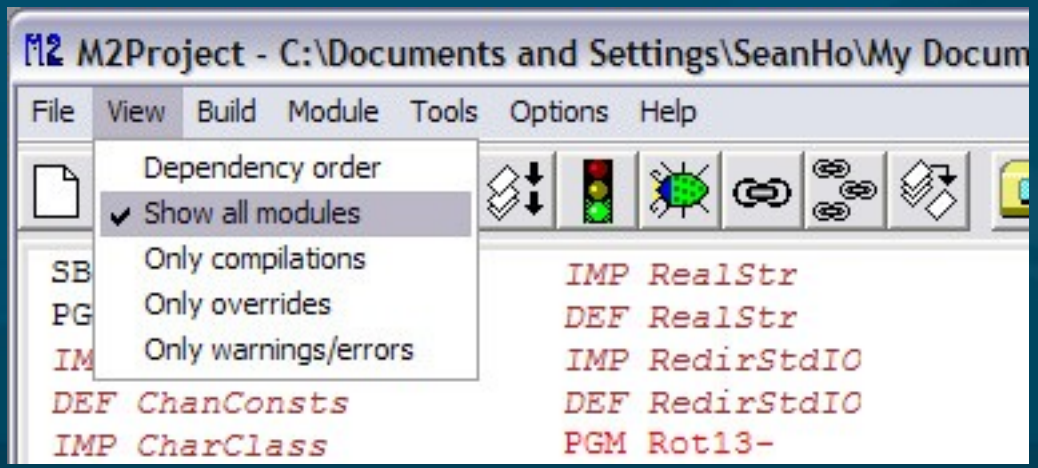
- Modules are really containers that delimit the visibility of entities (data structures, procedures)
- Every ISO Modula-2 installation comes with the Standard Library modules
- Today's focus is on the standard I/O modules:
  - SRealIO, SWholeIO, STextIO
  - RedirStdIO
- Monday we'll talk about standard math modules

ISO vs. classical Modula-2

- Stonybrook implements ISO standard Modula-2
- Some older compilers (e.g., possibly Mac) implement "classical Modula-2"
- There are some differences in naming of standard library modules: InOut vs. STextIO etc.
- For class purposes, just stick to ISO Modula-2 and ignore sections in the book on classical Modula-2

Intro to definition modules

- You can see what's provided in the standard library modules by selecting "View->Show all modules" in Stonybrook
- Double-click on a "DEF" module to see the definition modules; these tell you what procedures are provided in a module, and what parameters they expect (more details on Mon)



SWholeIO, SRealIO

- SWholeIO:
  - ReadInt (VAR int: INTEGER);
  - WriteInt (int: INTEGER; width: CARDINAL);
  - ReadCard (VAR card: CARDINAL);
  - WriteCard (card: CARDINAL; width: CARDINAL);
- SRealIO:
  - ReadReal (VAR real: REAL);
  - WriteReal (real: REAL; width: CARDINAL);
  - WriteFixed, WriteFloat, WriteEng ...

STextIO

- ReadChar (VAR ch : CHAR);
- ReadRestLine (VAR s: ARRAY OF CHAR);
- ReadString (VAR s: ARRAY OF CHAR);
- ReadToken (VAR s: ARRAY OF CHAR);
- SkipLine;
- WriteChar (ch: CHAR);
- WriteLn;
- WriteString (s: ARRAY OF CHAR);

A few new procedures in STextIO

- ReadRestLine (VAR s: ARRAY OF CHAR);
  - Reads input up to next newline (endOfLine state)
  - Stores as much as it can in s, throws rest away
- cf. ReadString:
  - Reads as much as it can, up to next newline
  - If it runs out of space first, then it leaves rest of line unread

ReadRestLine vs. ReadString

- Assume the user types:
  - Hi there!
  - OK, bye
- Assume s1, s2 : ARRAY [0..4] OF CHAR
- This yields s1 = "Hi th", and s2 = "OK, b":

```
ReadRestLine (s1);
ReadString (s2);
```

- This yields s1 = "Hi th", and s2 = "ere!":

```
ReadString (s1);
ReadString (s2);
```

ReadToken

- ReadToken (VAR s: ARRAY OF CHAR);
  - Reads input up to newline or space
  - Ignores leading spaces
  - Helpful for reading one word at a time

SkipLine revisited

- SkipLine
  - Reads input until next newline (endOfLine state)
  - Throws away all those characters
  - Clears endOfLine state

I/O channels

- Abstractly, a stream of input comes over a channel from a source
  - e.g., source can be keyboard, file, program, ...
- A stream is output over a channel to a sink
  - e.g., sink can be screen, file, program, etc.
- I/O channels (file descriptors, file handles) can be opened in one of three modes:
  - Read, write, and read/write
- Default source is keyboard, default sink is screen
- But we can redirect channels to other source/sink

Redirection

- I/O redirection allows us to read and write files as easily as we read and write to the screen
  - Handy for recording output in labs
  - Or scripting interactive programs
- After doing the redirection, all calls to WriteString etc. go to the newly redirected channel instead of the standard console interface
- ISO standard Modula-2 doesn't include standard procedures to redirect, but we have a special TWU-specific module, RedirStdIO:

RedirStdIO

- OpenOutput and OpenInput
  - Begin redirection of output or input from a file
  - Pops up a window asking for filename
- OpenOutputFile and OpenInputFile
  - Same as above, but filename is given as param
- CloseOutput and CloseInput
  - Return to normal console interaction
  - Always remember to close files after use!
- OpenResult()
  - Gets result of last open operation

ISO Standard Math library: RealMath

- We've already used most of the math functions:
  - sqrt, exp, ln
  - power (base, exponent : REAL): REAL
    - base must be positive (> 0.0)
- Constants:
  - pi = 3.14159265358979323846264338327...
  - exp1 = 2.71828182845904523536028747...
- Trigonometric functions:
  - sin, cos, tan, arcsin, arccos, arctan
- round (x : REAL) : INTEGER; (cf INT, TRUNC)

Review of today (6.1–6.4)

- ISO Modula-2 Standard Library modules: I/O
- Some new functions in STextIO:
  - ReadRestLine, ReadToken
- I/O channels
- Redirection
  - TWU library RedirStdIO
- Math Standard Library module: RealMath

TODO items

- Lab5 due next week:
  - §6.11 #(25 / 33) (choose one)
- Quiz ch6 next Wed
- Quiz ch7 next Fri (one week from today)
- CMPT140 Final two weeks from today
- Reading: through §6.8 for Mon